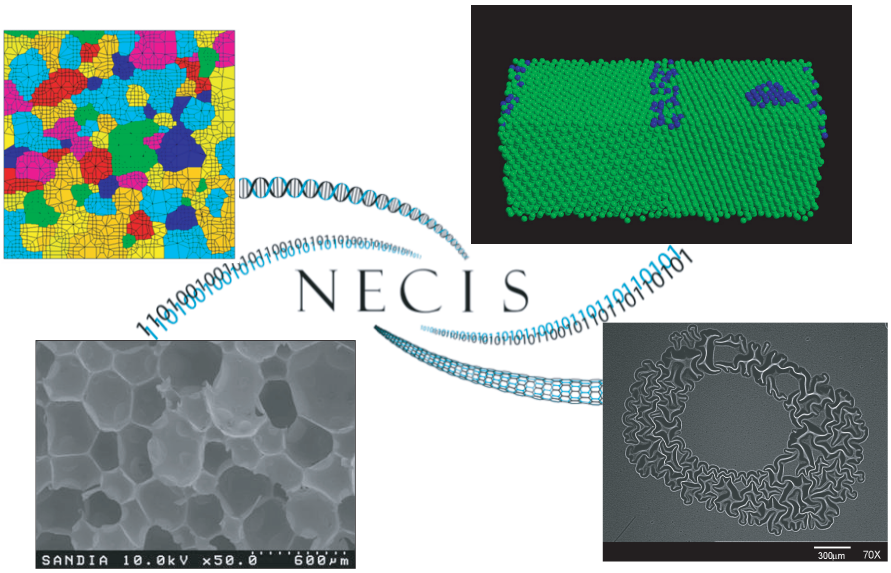
CSRI SUMMER PROCEEDINGS 2006

The NECIS Focus in Nanotechnology

Editors:
S. Scott Collis, Jean Lee, Jonathan Zimmerman
Sandia National Laboratories

October 18, 2006



SAND 2006-6564P

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed-Martin Company, for the United States Department of Energy
under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/ordering.htm

Preface

NECIS is The **N**anoscience, **E**ngineering, and **C**omputation **I**nstitute at **S**andia that was established as a project within the 2006 Sandia Engineering Excellence late start Laboratory Directed Research and Development (LDRD) program. NECIS conducted 48 coordinated research collaborations, from May through September of 2006, focused on the integration of nanoscale physical and biological science with computational science. In collaboration with key university partners, NECIS was designed to inspire and expedite breakthroughs in nanotechnology that lay the foundation for discovery institutes in the spirit of the President's "American Competitiveness Initiative." In this capacity, NECIS has addressed these challenges by enabling:

1. Collaborations between nanoengineering and computational science;
2. Students to conduct research, on-site, with Sandians to train a new generation of researchers skilled in experimental, modeling, and simulation aspects of nanotechnology;
3. Domain experts from academia to work on-site at Sandia to perform cutting-edge research that will impact new curricula for nanotechnology;
4. Workshops and short courses to facilitate innovative research that combines experimental and computational approaches to nanoengineering.

This proceedings is a collection of research articles documenting NECIS collaborative research in the technical focus areas of: Nanoscale Defects in Materials, Nano and Micro Fluidics, the Nano-to-Micro Interface, the Nano/Bio Synergy, and Enabling Computational Nanoscience. Articles are organized into these technical areas with an overview for each contributed by the focus-area technical leads. As part of the research and research training goals of NECIS, it was the intent that these articles serve as precursors-to or first-drafts-of articles that could be submitted to peer reviewed archival journals. As such, each article has been reviewed by a Sandia staff member knowledgeable in that technical area with feedback provided to the authors. Several articles have or are in the process of being submitted to peer-reviewed conferences or journals and we anticipate that additional submissions will be forthcoming. NECIS' overall accomplishments including faculty visits, seminars, short-courses, and the NECIS Workshop will be documented in a separate internal Sandia report.

We would like to thank all NECIS participants who have contributed to the outstanding technical accomplishments of this project as documented by the high quality articles in this proceedings. The success of NECIS hinged on the hard work of 48 enthusiastic student collaborators, 54 dedicated Sandia technical staff, and 14 visiting faculty participants. We think that readers will be duly impressed when they realize that the research described herein occurred primarily over a three month period of intensive collaboration.

NECIS benefited from the administrative help of Deanna Ceballos, Bernadette Watts, Eilene Cross, Mel Loran, Dee Cadena, Jennifer Bamberger, Colleen Loretto, Kelly Doty, and Martha Campiotti. The success of NECIS is, in large part, due to their dedication and care and it is much appreciated. Finally, we would like to thank David Womble for his advice, guidance and overall project management.

S. Scott Collis
Jean Lee, and
Jonathan Zimmerman

October 30, 2006

Table of Contents

Preface
S.S. Collis, J. Lee, and J. Zimmerman iii
Nanoscale Defects in Materials
J. Lee 1
Ab Initio Simulations of Compound Semiconductor Alloys
J.E. Bickel, N.A. Modine, and J. Mirecki-Millunchick 3
Current-voltage characterization of molecular species on Au nanospheres
J. Funamura and D. Robinson 10
Cohesive Zone Model for Void Nucleation
S. Goff and J. Dike 14
Thermal Conductivity of Carbon Nanotubes
A. Henry and S. Plimpton 21
Entaglement in Polymer Brushes
R.S. Hoy, S.J. Plimpton, and G.S. Grest 27
Adsorption of Nucleotide Monophosphates on Carbon Nanotubes
T. Iaconis and D. Robinson 35
Functional Metal-Organic Frameworks
V. Liu, C. Bauer, M. Allendorf and R. Simmons 39
Cleaning Techniques and Mechanical Properties of Diatom Frustules
T. Lynch and B. Simmons 44
A Generalized Continuum Model for Finite Deformation Crystal Plasticity
J. Mayeur and D. Bammann 51
Deposition of Nanoparticles on Textured Surfaces
H. Moore and B. Simmons 54
Atomic Simulations of Stress Evolution
C-W. Pao, E.B. Webb III, et. al 58
Site-Specific Atom-Probe Tomograpy Analysis of CU Alloys
J. Riesterer and E. Marquis 65
Modeling Polycrystalline Mechanics
J. Sanders, T. Voth, and J. Robbins 76
Mechanical Properties of Metal-Organic Framework-5
M. Shindel, M. Allendorf and R. Stumpf 85
Nanofiller and Chemical Strengthening of Polyurethane Foams
B. Weissman and A. Vance 89
Nano/Micro-Fluidics
S.S. Collis 97
Distributions of Atoms in Fluids using LOCA
K.I. Dickson, C.T. Kelley, et al. 99
Stokes flow with Polymerization
J. Fettig and S.S. Collis 106
Coupled Problems for Microfluidics
C. Harder and P. Bochev 114
Adhesion of Nanoscale Gold Films on Silicon Substrates
M.S. Kennedy and N.R. Moody 128
Immersed Finite Element Method
A. M. Kopacz, T. D. Nguyen, and G. J. Wagner 140
Diffusive Transport of Fluorescein in a Nanofluidic Device
S. Rhieu, D. Huber, and S. Pennathur 152

Minimum-Dispersion Microfluidic Devices
A.R. Terrel and K.R. Long 158
Multiscale Methods for Nonlinear Gas Chromatography
G. von Winckel, L.A. Romero, and E.A. Coutsias 168
The Nano-to-Micro Interface
J. Zimmerman 175
Local Atomic-Scale Elastic Moduli
N. Burgess, J. Zimmerman, and T. Delph 177
Peridynamic Framework for Simulation of Cracks
J. Riendeau and R. Lehoucq 183
Polycrystalline Microstructures for Crystal Plasticity Simulation
J. Stinson, E. Marin, and D. Bammann 187
Numerical Experiments for Local Quasicontinuum Analysis
E.B. VanderZee, M.L. Parks, and P. Knupp 194
The Nano/Bio Synergy
S. Plimpton 203
Quantum Dot Bioconjugates for Cancer Detection
O. Elboudwarej and V. Vandernoot 205
Lock-In Amplifier for the NIH Biodetector
B. Hinze and J. Brennan 209
Design and Fabrication of Robust Gas Sensors
S. Marshall, W. Medlin, and R. Bastasz 216
Visual Simulation of Biological Pathways
M. Mehne and E.E. May 219
Sub-structured Biomolecular Dynamics Simulations
R.M. Mukherjee and P. Crozier 229
Block Preconditioners applied to Induced-charge electro-osmosis
R. Shuttleworth, el at. 241
Enabling Computational Nanoscience
A.G. Salinger 249
Reduced-Order Model Construction for High-Dimensional Systems
O. Bashir, K. Willcox, et al. 251
Java-Based Atomistic Visualization Tools
N. Benavides, C.A. Phillips, and J.-P. Watson 264
Multimodal Reliability Assessment
B.J. Bichon, M.S. Eldred, and L.P. Swiler 273
Massively Multithreaded Shortest Path Algorithms
J. Crobak and J. Berry 284
Optimizing Mesh Optimization
E. Johnson and M. Brewer 288
Reversible Quantum-dot Cellular Automata
S. Frost-Murphy, E. DeBenedictis, and P. Kogge 294
Composite Refinement of Unstructured Conformal Hexahedral Meshes
M.H. Parrish, M.L. Staten and M. Borden 306
Reconfigurable Functional Units
K. Rupnow and K. Underwood 313
Deeply Pipelined QCA Architectures
M. Vance and K. Underwood 321

Nanoscale Defects in Materials

The role of nanoscale defects in a macroscopic materials system is analogous to the role of David in the proverbial David and Goliath story: these tiny defects can dramatically change the properties and performance of a macroscopic materials system. The presence of nanoscale defects in materials is often the reason why there is a mismatch between the ideal, theoretical behavior of a material and its actual, real behavior. Frequently the presence of nanoscale defects in materials is inescapable, and it behooves us to understand and account for them if we seek to tailor materials behavior.

The articles in this section of the Proceedings describe NECIS research conducted by Sandia staff and student collaborators that investigate how nanoscale defects and features in nanoscale materials influence materials properties and performance. For example, Weissman and Vance studied the use of nanomaterial fillers to strengthen TufFoam without causing any chemical incompatibility between the nanomaterial filler and TufFoam. Hoy, Plimpton, and Grest investigated the effects of brush chain length and brush coverage on the adhesive behavior of materials containing nanoscale polymer brush adhesion promoters. The insertion of fluorescent linker molecules into a Zn-O matrix by Liu, Bauer, Allendorf, and Simmons moves forward the potential for these metal-organic frameworks to be used for gas storage or sensing. The creative and original research presented in this section support Sandias programs in nuclear weapons; homeland security; and energy, resources, and non-proliferation. The research described here fills knowledge gaps and helps seed future Sandia research in the important area of nanoscale defects in materials.

Jean Lee

October 30, 2006

AB INITIO SIMULATIONS OF COMPOUND SEMICONDUCTOR ALLOYS

J.E. BICKEL*, N.A. MODINE†, AND J. MIRECKI-MILLUNCHICK‡

Abstract. The atomic surface structure of compound semiconductors plays a large role in the growth of semiconductor films and the final microstructure of the film. During growth of $In_xGa_{1-x}As$ films, a mixed-termination surface consisting of a (4x3) reconstruction with common binary InAs or GaAs reconstructions, such as the α2(2x4), has been observed. We have used Density Functional Theory (DFT) to determine the effects of strain and alloying on the α2(2x4) reconstruction and combined DFT calculations and simulated Scanning Tunneling Microscopy (STM) images to determine the structure of the (4x3) reconstruction.

1. Introduction. The atomic surface structure of compound semiconductors plays an important role in the growth of thin films and devices made of these alloys, influencing the compositional uniformity and the morphology of the grown material. It also influences ordering and surface segregation of individual atoms [1]. It is important to understand the surface structure in order to control the growth of electronic and optoelectronic devices made of III-V semiconductors where planar and compositionally abrupt interfaces are required to obtain optimal device properties. A lot of work has been aimed at understanding binary compound semiconductors such as GaAs [2] InAs [3] and GaSb [4], however, less is known about ternary alloys such as InGaAs.

Experimental work on InGaAs shows that, under most growth conditions, a (4x3) reconstruction dominates the surface with some mixed surfaces containing regions of α2(2x4), β2(2x4), and c(4x4) reconstructions, which are common reconstructions of binary InAs and GaAs. An example of this mixed surface reconstruction can be seen in Figure 1.1. Two models for the atomic structure of the (4x3) reconstruction

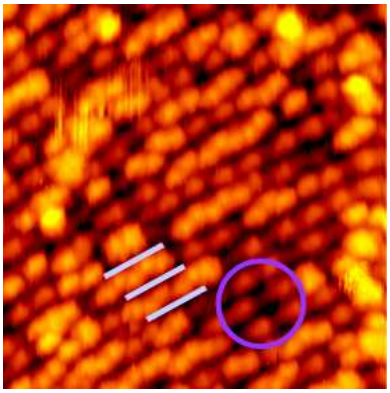


FIG. 1.1. *STM image of $In_{0.27}Ga_{0.73}As$ grown on GaAs(100) showing a mixed reconstruction of α2(2x4) (between lines) and (4x3) (in the circle). Image taken at -2.3V, 100pA.*

have been proposed, one by our group [5] and one by the Jones group [6] which is a variation on a model proposed by Sauvage-Simkin [7]. We have used density functional

*The University of Michigan, jebickel@umich.edu
†Sandia National Laboratories, namodin@sandia.gov
‡The University of Michigan, joannamm@umich.edu

theory (DFT) to calculate the energies of the (2x4) and (4x3) reconstructions and to simulate scanning tunneling microscope (STM) images of both in order to compare experimental and simulated results directly.

2. DFT calculations and STM simulations. Each reconstruction was studied using a slab consisting of 4 bilayers of bulk III-V material terminated on top by the desired reconstruction and on the bottom with pseudo-hydrogen. The charge of the pseudo-hydrogen was chosen to make the bottom surface electrically passive (i.e., able to neither accept electrons from nor donate electrons to the top surface reconstruction). Each slab had a sufficient amount of vacuum separating it from its periodic images, 14$\AA$ for most slabs and 11$\AA$ for 5 layer slabs used for bulk energy calculations. Calculations were performed while keeping the bottom bilayer and the pseudo-hydrogen atoms fixed at positions found during an initial relaxation.

All energy calculations were performed using the Vienna *Ab Initio* Simulation Package, VASP, which is a plane wave pseudopotential code that solves for the electronic structure of solids and surfaces using DFT [8]. Ultrasoft pseudopotentials [9,10] and the local density approximation (LDA) [11] were used. A k-point convergence test was performed and led to the use of 3x4 k-point sampling for (4X3) unit cells, 6x3 k-point sampling for (2x4) unit cells, and 3x3 k-point sampling for (4x4) unit cells (the zigzag-α2(2x4) cell is a (4x4) cell due to its change in periodicity). An energy cut-off of 203.1 eV was used for all calculations, and the calculations were relaxed until the total energy was converged to better than 0.1 meV.

Lattice parameters were determined using volume relaxation of bulk crystals with an energy cut-off of 350eV. This higher energy cut-off avoids errors inherent to volume relaxations in VASP. In general, LDA lattice parameters are smaller than experimental values. A chart of LDA lattice parameters used, and experimental lattice parameters is shown in Table 2.1.

Table 2.1
Lattice parameters determined by LDA and Experiment

	Experimental (Å)	LDA (Å)
GaAs	5.653	5.592
InP	5.869	5.828
mid-lp	N/A	5.900
InAs	6.058	6.018

Simulated STM images were generated after the method described by Tersoff and Hamann [12]. Tersoff and Hamann used the Bardeen formalism [13] to show

$$I \propto \int_{E_f}^{E_f+eV} \rho(\mathbf{r}, E)dE \tag{2.1}$$

where

$$\rho(\mathbf{r}, E) = \Sigma|\Psi_i(\mathbf{r})|^2\delta(E - E_i) \tag{2.2}$$

In these equations, I is the tunneling current, E_f is the Fermi Energy, e is the charge of an electron, V is the bias voltage, and $\rho(\mathbf{r}, E)$ is the local density of states (LDOS) of the surface, which is calculated by summing over the wave functions Ψ_i with energy E_i at the tip position $\mathbf{r}$. VASP can be used to calculate the integral in equation 2.1

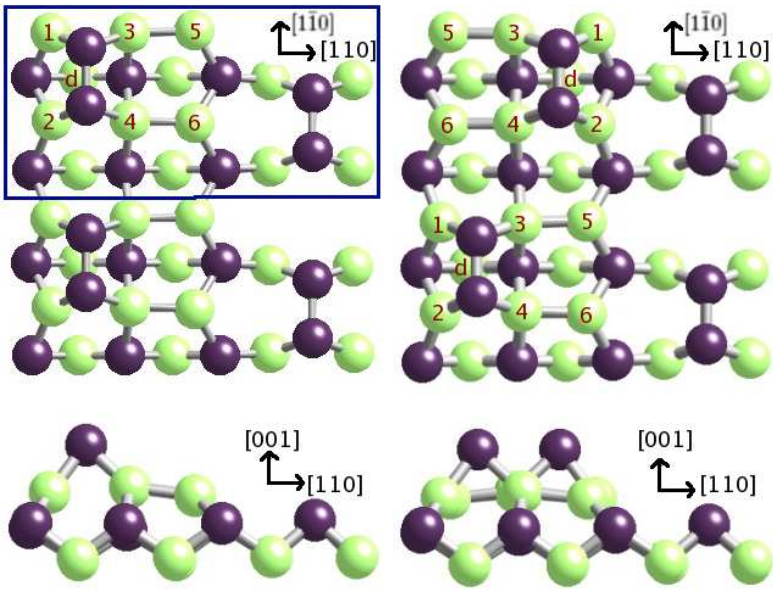
and will output the integrated LDOS for a given bias voltage. This output consists of the current density at each (x,y,z) grid point calculated. Filled state LDOS grids were computed by using a bias voltage of -1V, and empty state grids were computed at a bias voltage of +1V. Constant current images were then generated by computing the height of a a particular isosurface. Since equation 2.1 does not allow the calculation of the actual current, but only a value proportional to it, the LDOS value used to generate the isosurface was varied until the simulated images resembled experimental images for the well understood α2(2x4) structure. The images were generated by cycling through each (x,y) grid point from z=max to z=min until the LDOS at point (x,y,z) was greater than the desired isosurface value. The z-value was then exported and plotted to generate the constant current image. This method of generation is exactly like an experimental STM tip slowly approaching the surface until the desired tunneling current is reached.

3. Zigzag-α2(2x4) reconstruction. It has been seen experimentally that 20 monolayer (ML) films of $In_{0.27}Ga_{0.73}As$ grown at 475°C have a surface reconstruction that is a combination of (4x3) and α2(2x4) with the α2(2x4) appearing in small domains. Interestingly, the bright dots of the surface dimers in the α2(2x4) reconstruction form a regular zigzag pattern rather than the line with occasional stochastic kinks that is generally seen in binary GaAs and InAs. This zigzag pattern is highlighted by the lines in Figure 1.1, where the $In_{0.27}Ga_{0.73}As$ alloy was grown on a GaAs(100) substrate with a misfit strain of 1.9%.

The α2(2x4) reconstruction is a surface reconstruction that consists of an anion, or Group V, dimer which can sit in 2 places above 6 surface cations, or Group III atoms. A regular straight-row-α2(2x4) reconstruction can be seen in Figure 3.1(a). This shows two unit cells next to each other, with the dimer sitting above cations 1-4. The dimer can also sit above cations 3-6. In two neighboring cells, the dimer can either sit in the same position, or swap positions as it does in the zigzag-α2(2x4) shown in Figure 3.1(b).

In order to determine whether strain is responsible for stabilizing the zigzag-α2(2x4) relative to the straight-row-α2(2x4), we performed DFT calculations for GaAs and InAs slabs with the zigzag-α2(2x4) and straight-row-α2(2x4) reconstructions at the GaAs, InP, mid-lp, and InAs lattice parameters given in Table 2.1. By comparing the total energies of these calculations, we found that the zigzag structure is more stable than the straight-row structure under these binary conditions, and becomes relatively more stable with tensile strain and less stable with compressive strain. The energy difference of the straight-row-α2(2x4) and zigzag-α2(2x4) is given in Figure 3.2, where it can be seen that the largest energy difference (for a purely GaAs slab at an InAs lattice parameter) is only 21 meV. Temperature effects should randomize the occupations of structures whose energies differ by less than $k_B * T$, where k_B is the Boltzmann constant and T is the temperature in Kelvin. At room temperature, $k_B * T$ is 25meV, and this results in a stochastic switching of the α2(2x4) dimer positions with no long range order. This explains why an ordered zigzag pattern is not seen in binary GaAs and InAs even though it is energetically favored. Therefore, strain does not explain the zigzag-α2(2x4) seen in the *compressively* strained $In_{0.27}Ga_{0.73}As$ system.

In an effort to understand the effects of alloying on the relative stability of the zigzag and straight-row α2(2x4) reconstructions, different combinations of cations were studied: (1) an InAs slab with a Ga atom beneath the surface dimer at the position marked d in Figure 3.1, (2) a GaAs slab with In atoms in the top six positions



(a) Straight-row-α2(2x4) (b) Zigzag-α2(2x4)

FIG. 3.1. *Surface reconstruction of straight-row and zigzag α2(2x4). The blue box shows a single α2(2x4) unit cell. Purple = Anion (As), Green = Cation (Ga or In)*

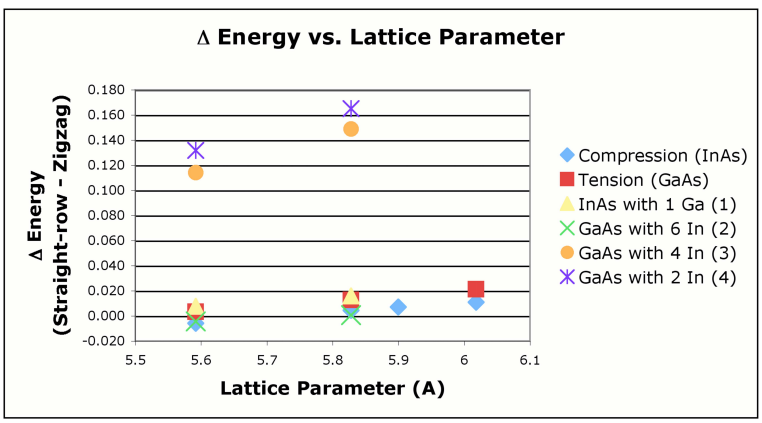
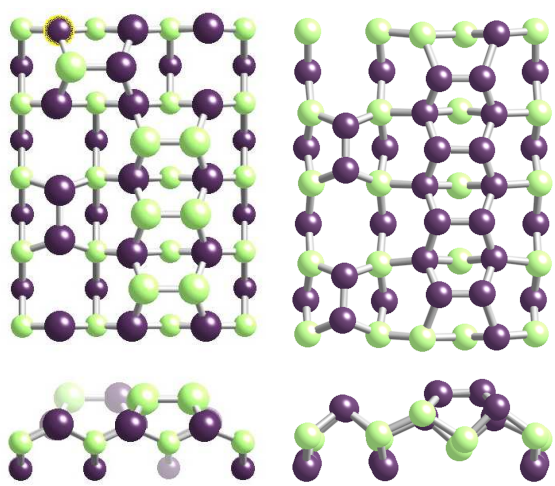


FIG. 3.2. *Stability of zigzag-α2(2x4) vs. straight-row-α2(2x4) as a function of lattice parameter. A positive value means the zigzag formation is more stable.*

marked 1-6 on Figure 3.1, (3) a GaAs slab with 4 In atoms in positions 3-6 of Figure 3.1, and (4) a GaAs slab with 2 In atoms in positions 5 and 6 of Figure 3.1. The difference in energy between the zigzag and straight-row structures is shown in Figure 3.2. In all cases the zigzag structure becomes more stable as the slab is strained in tension. Configuration (1) did not allow the dimer bond to contract sufficiently to induce the dimer to alternate in the zigzag pattern even at room temperature. Configuration (2) actually resulted in the straight-row structure being more stable or equally stable at both the GaAs and InP lattice parameter. Configurations (3)

and (4), however, resulted in a large stabilization of the zigzag-α2(2x4) relative to the straight-row structure. Both variants allow the In atoms, which are large relative to the Ga atoms, to segregate to the surface to relieve strain. Furthermore, the stretched cation-cation bonds between atoms 3 and 5 and atoms 4 and 6 become more stable with the larger In atoms. Indium atoms in positions 5 and 6 also would like to expand in the $[1\bar{1}0]$ direction. This expansion can be accommodated easily in the zigzag structure, where they are next to Ga atoms, but not in the straight-row structure, where they are next to more In atoms. As a result, the zigzag structure becomes much more stable than the straight-row with both configurations (3) and (4) showing a stabilization of the zigzag structure by over 100 meV for both the GaAs and InP lattice parameters. This difference in energies is large enough to be significant even at growth temperatures of 475°C where $k_B * T$ is 64meV.

4. (4x3) Reconstruction. Experimentalists have suggested two different models to account for the (4x3) reconstruction seen in the InGaAs system. The first was proposed by the Mirecki-Millunchick group [5] which can be seen in Figure 4.1(a), and the second is a model by Sauvage-Simkin [7] that was modified by the Jones group [6] so that it obeys the electron counting rule and is shown in Figure 4.1(b). There are a few important differences between these models. The MM model has heterodimers (cation-anion) on the surface as well as cation homodimers, whereas the SSJ model only has anion dimers. Because of the cation terminated surface, the MM model has been criticized as not having enough As, though it should be noted that the surface stoichiometry of the MM(4x3) is the same as for the α2(2x4) and as such is not anion deficient but simply cation-capped. The SSJ model, on the other hand, is very anion rich. Another notable difference between the two models is the movement of the heterodimer over one position in the MM-model, compared to the straight dimer row in the SSJ-model.



(a) MM-Model (b) SSJ-Model

FIG. 4.1. *Two models for the (4x3) InGaAs surface reconstruction. Purple = Anion (As), Green = Cation (Ga or In). Note, some lower layer atoms excluded for image clarity.*

Both the Mirecki-Millunchick and the Jones groups have STM images of the (4x3) reconstruction that show "raised dot features" [6] as seen in Figure 4.2. We used DFT

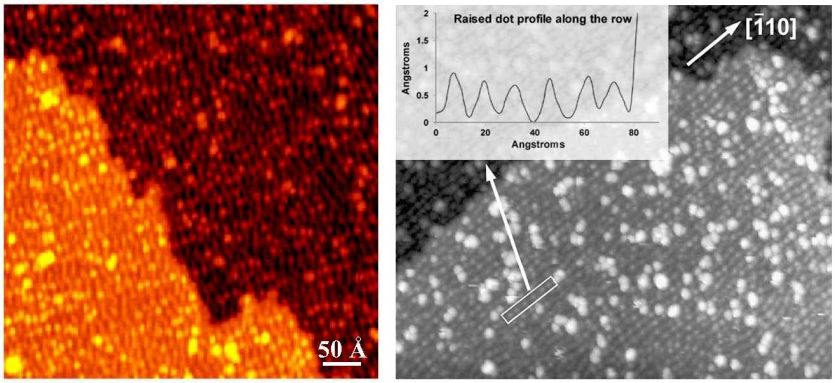
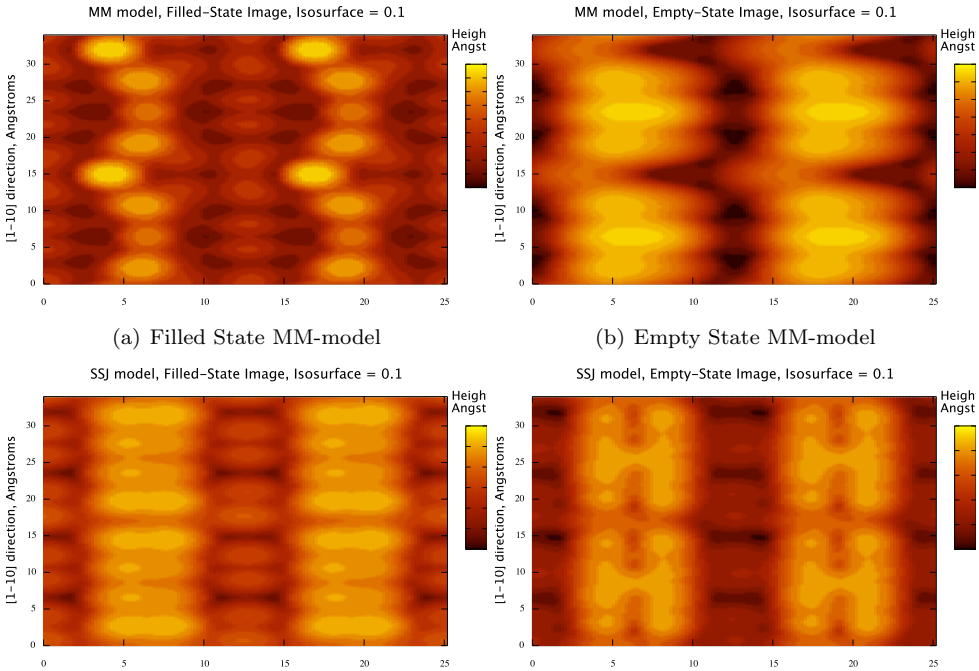


FIG. 4.2. *Filled state STM images from the Mirecki-Millunchick [14] and the Jones group [6] showing the dot-like (4x3) reconstruction.*

to simulate STM images for both the MM and SSJ models so that comparison of the simulated and experimental STM could be used to help determine the correct model. Both empty state and filled state simulated STM images can be seen in Figure 4.3. The atomic structure of these figures matches that of Figure 4.1 repeated twice in both the x and y directions. The anion dimers of the SSJ-model show a row of



(a) Filled State MM-model (b) Empty State MM-model

(c) Filled State SSJ-model (d) Empty State SSJ-model

FIG. 4.3. *Simulated STM images at $\pm 1V$ bias voltage for empty and filled state images respectively.*

broad oval shapes commonly seen in STM of anion dimers; the anion dimer in the $\alpha 2(2x4)$ is usually described as a dumbbell. The heterodimer in the MM-model, on the other hand, appears as a bright dot, with the faded cation-dimers forming a chain between heterodimers. Given the limitations in resolution of experimental STM to resolve lower height features, the MM-model could appear as a bright dot with a faint connecting line similar to that seen in the experimental STM in Figure 4.2. The one drawback of this model is the calculated energy. The MM-model has the same stoichiometry as the $\alpha 2(2x4)$ reconstruction, which allows the direct comparison of DFT-calculated total energies. The MM (4x3) model has a considerably higher surface energy than the $\alpha 2(2x4)$ suggesting that it is not stable. However, it is possible that the effects of strain or alloying may explain this difference. As was seen with the zigzag-$\alpha 2(2x4)$, small changes in alloying or strain may cause a reconstruction to become more or less stable. Some variations to try include adding a surface hetero or homodimer above the cation dimers which may lower the surface energy and may also result in simulated STM images that match even better with experimental STM.

5. Conclusions. DFT has been used to explore the effects of strain and alloying on the stability of surface reconstructions, and we have been able to show that the stability of the zigzag-$\alpha 2(2x4)$ reconstruction is due to the effects of alloying rather than the strain in the system. We have also been able to successfully simulate filled and empty state STM images to address different models of the (4x3) reconstruction proposed for the InGaAs system and show that the MM model is the best fit thus far between simulated and experimental STM. However, energy calculations suggest that additional variations of the MM model should be explored.

REFERENCES

[1] S. Froyen, A. Zunger, Phys. Rev. B 53 (1996) 4570.
[2] V. Bresslerhill, M. Wassermeier, K. Pond, R. Maboudian, G.A.D. Briggs, P.M. Petroff, W.H. Weinberg, J. Vac. Sci. Technol. B 10 (1992) 1881.
[3] W. Barvosa-Carter, R.S. Ross, C. Ratsch, F. Grosse, J.H.G. Owen, J.J. Zinck, Surf. Sci. 499 (2002) L129.
[4] A.S. Bracker, M.J. Yang, B.R. Bennett, J.C. Culbertson, W.J. Moore, J. Cryst. Growth 220 (2000) 1492.
[5] J. Mirecki-Millunchick, A. Riposan, B.J. Dall, C. Pearson, B.G. Orr, Surf. Sci. 550 (2004) 1.
[6] P.A. Bone, J.M. Ripalda, G.R. Bell, T.S. Jones, Surf. Sci. 600 (2006) 973.
[7] M. Sauvage-Simkin, Y. Garreau, R. Pinchaux, A. Cavanna, M.B. Véron, N. Jedrecy, J.P. Landesman, J. Nagle, Appl. Surf. Sci. 104/105 (1998) 16177.
[8] G. Kresse, J. Hafner, Phys. Rev. B 47 (1993) 558. G. Kresse, J. Haafner, Phys. Rev. B 49 (1994) 14251. G. Kresse, J. Furthmüller, Comput. Mat. Sci. 6 (1996) 15. G. Kresse, J. Furthmüller, Phys. Rev. B 54 (1996) 11169.
[9] G. Kresse, J. Hafner, J. Phys.: Condens. Matter 6 (1994) 8245.
[10] D. Vanderbilt, Phys. Rev. B 41 (1990) 7892.
[11] D.M. Ceperley, B.J. Alder, Phys. Rev. Lett. 45 (1980) 566. J.P. Perdew, A. Zunger, Phys. Rev. B 23 (1981) 5048.
[12] J. Tersoff, D.R. Hamann, Phys. Rev. B 31 (1985) 805.
[13] J. Bardeen, Phys. Rev. Lett. 6 (1961) 57.
[14] A. Riposan, "Surface Reconstructions and Morphology of InGaAs Compound Semiconductor Alloys" (PhD dissertation) University of Michigan.

CURRENT-VOLTAGE CHARACTERIZATION OF MOLECULAR SPECIES ON AU NANOSPHERES

J. FUNAMURA* AND D. ROBINSON†

Abstract. We present a new approach for a flexible, low-cost, high-yield device for the current-voltage (IV) characterization of molecular species. Using a close-packed array of gold nanospheres over patterned electrodes, we coat the nanospheres with an organic molecule and measure the IV relationship through the particle array. Adsorption of different molecules on the gold particles results in reliably distinct IV relationships, leading us to believe that we are measuring the resistance of a network of molecules bridging the gold nanospheres. Surface-enhanced Raman spectroscopy, made possible by the scale of our device and the optical properties of the gold nanospheres, is used to verify the presence and identity of an adsorbed molecule.

1. Introduction. Twenty-two years ago, Aviram and Ratner [1] proposed the usage of individual organic molecules as electronic devices. Since then, the field of molecular electronics has grown, with researchers choosing candidate molecules for devices, developing computational models to predict and better understand electron transport, and creating experimental platforms to test the models. Today, the big challenge in molecular electronics is still developing reliable devices for characterizing the electrical properties of candidate molecules for molecular wires and other devices. Previous efforts involve elaborate setups of with small length scales on the order of 30 nm or smaller [2]. A larger length scale would allow the usage of other forms of spectroscopy that may help further understand the fundamentals of molecular-level charge transfer. Here we present an innovative approach to creating a test platform that is low-cost, easy to produce, and allows characterization methods that require this larger length scale.

2. Experimentation. We start with a silicon wafer of evaporated gold interdigitated electrode arrays that enable electrical measurements using a probe station. They have nominal gap widths of 1-2 microns. The wafer is primed with a monolayer of hexamethyldisilizane (HMDS) to promote adhesion of the gold nanospheres. Wafer pieces are then individually dipped in a toluene suspension of 20nm Au nanospheres with 1,2-Dioleoyl-sn-glycero-3-phosphoethanolamine (DOPE) as a surface stabilizer. Retracting the wafer at a rate of 2mm/min consistently produced a relatively uniform single monolayer of close-packed nanospheres.

Current-voltage (IV) measurements were taken using a probe station across adjacent electrodes to determine a baseline IV curve from -4V to 4V. Most results were undetectable (sub-nanoamp) or showed a symmetric sub-microamp current.

After baseline IV characterization, we soak the wafer piece in an ethanol solution containing the candidate molecule for 12 or more hours to allow the DOPE to go into solution and be replaced by the other molecules in the solution. After removal and a gentle ethanol rinse, IV measurements are taken again throughout the same voltage range and compared to the baseline set. The candidate molecules we tested were 1,4-benzendithiol, 1,4-benzenedimethanethiol (xylyl dithiol), benzyl mercaptan, and thiophenol.

3. Results. In our experiments, we found consistent changes in total resistivity after replacing the DOPE with a candidate molecule. Furthermore, we found trends

*University of California at Berkeley
†Sandia National Laboratories, drobins@sandia.gov

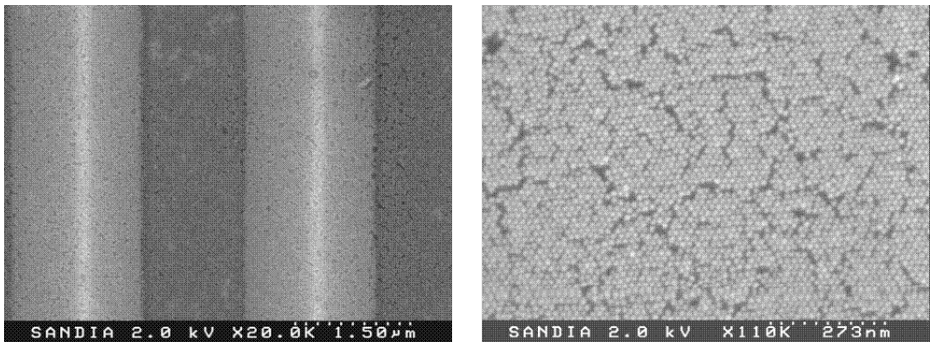


FIG. 2.1. *Field emission scanning electron microscope images of close-packed Au particles deposited between gold-patterned electrodes.*

that distinguished between the resistivity changes of different candidate molecules. On a normal electode array, a total of 16 measurements were taken: a set of four for each nomial gap size. The results were generally consistent within each set of four measurements, with each set generally showing increased resistivity with an increase in gap size.

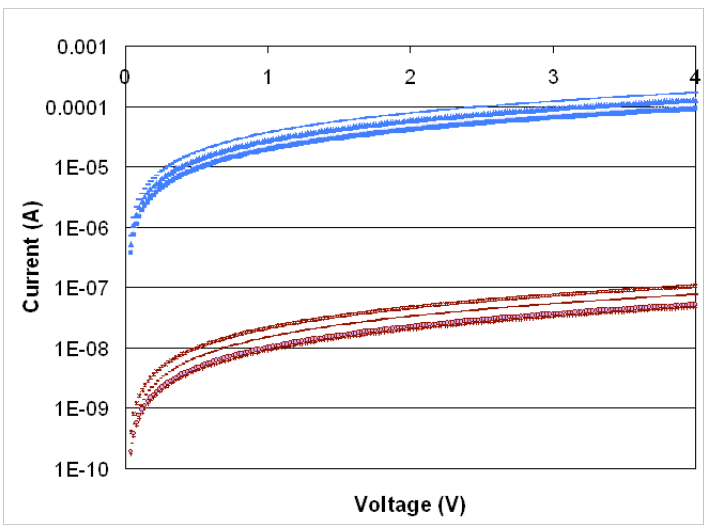


FIG. 3.1. *Characteristic pre/post-soak current-voltage data. The lower grouping is a set of electrode baseline IV data. The upper grouping is the same set after benzenedimethanethiol molecules has replaced the DOPE on the Au surface.*

The overall resistivity measurement averages were calculated from the difference between the final IV measurements of the candidate molecule and the baseline IV measurements of the DOPE-coated particles. The following data shows resistivity changes for the various gap sizes. The control, soaking in an ethanol + DOPE solution, resulted in no measurable difference in IV characteristics.

The nanoscale geometry of the gold nanospheres lends itself well to the use of surface-enhanced Raman spectroscopy (SERS) to determine molecular species adsorbed to the gold. The size and accessibility of the test platform allows the laser

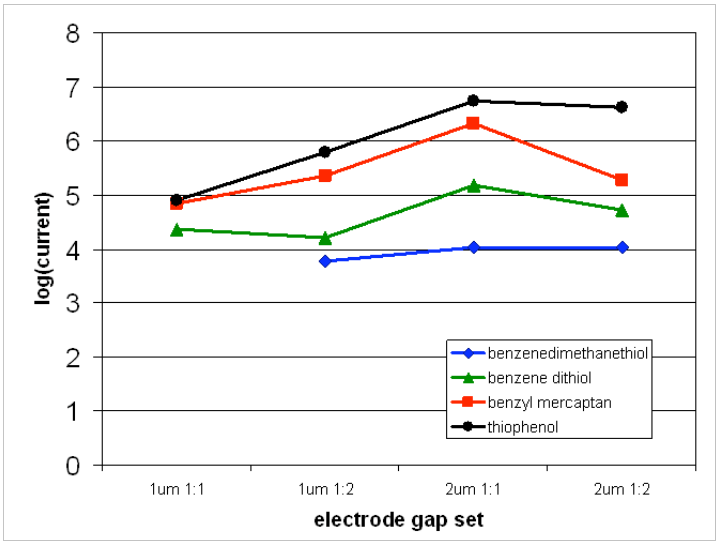


FIG. 3.2. *The current increase observed when comparing the IV measurements of the candidate molecules to the baseline IV observed with DOPE-coated particles. Note that the order of resisitivities of the four molecules is consistent for different gap sizes as expected.*

beam required by SERS to see the molecules. Raman spectroscopy performed on these samples confirms that the DOPE molecules were actually replaced by the intended molecule. Here, we show the presence of benzenedimethanethiol (xylyl dithiol) after soaking.

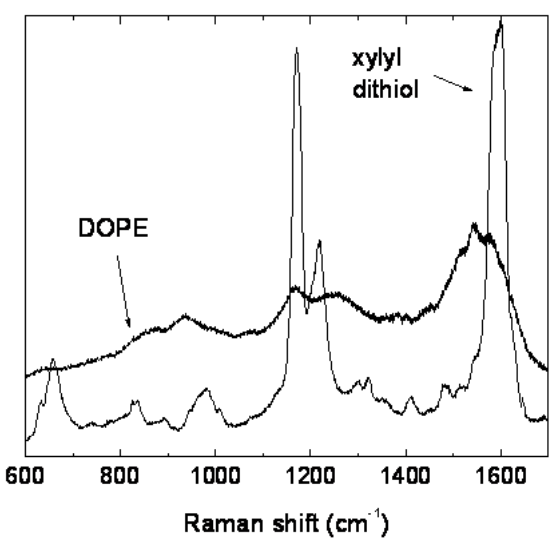


FIG. 3.3. *Raman spectroscopy data showing the presence of DOPE before soaking and the presence of xylyl dithiol after soaking.*

4. Discussion. This system achieves our current goals. It is relatively easy to fabricate, requires no prohibitively expensive equipment, has the geometry and

device size to accommodate different characterization methods, and has a high yield of productive electrode pairs. The results show clear increases in current from switching the ligand to different candidate molecules, and the optical data corroborates the existence of the newly adsorbed molecule. The results also demonstrate that, unlike some other device geometries, characterization is possible at every step of the process.

Future work on this platform might expand to measuring the properties of other interesting aromatic thiols and further characterization of adsorbed molecules using additional methods besides just SERS and IV. In this work, SERS was used primarily as a method for identification and verification of the ligand swap for the IV data. Additional methods such as infrared reflectance spectroscopy and X-ray photoelectron spectroscopy will be employed to further characterize the adsorbed molecules. The device geometry allows the possibility of using multiple characterization techniques simultaneously.

Molecular electronics shows promise as the future of our electronic devices. The more we know about the electrical transport properties of molecules, the more we can utilize them. Thus it is imperative for us to continue to develop devices and techniques with which we may discover and explore this exciting and emerging field.

REFERENCES

[1] A. Aviram and M.A. Ratner, Chem. Phys. Lett., 29 (1974).
[2] M.A. Reed, C. Zhou, C.J. Muller, T.P. Burgin, and J.M. Tour, Science, 279 (1997), pp. 252–254.

INVESTIGATION OF A COHESIVE ZONE MODEL FOR SIMULATING VOID NUCLEATION AND ITS CONNECTION TO NUCLEAR SAFETY ASSESSMENT

S. GOFF* AND J. DIKE†

Abstract. Nuclear safety assessments have been the focus of abnormal mechanical environment modeling efforts for multiple weapons systems over the last few years (i.e., W80-3, W76 and W87). The modeling efforts are aimed at predicting metal failure of common engineering materials such as steel and aluminum in order to determine if exclusion regions within the weapon are breeched during abnormal mechanical events such as a drop during handling or a crash during transport. Simulations of the W80-3 subjected to drops have been performed using the Sandia-developed EMMI (Evolving Microstructure Model of Inelasticity) plasticity and failure model. EMMI accounts for void nucleation and growth in an elastic-plastic material with failure parameters derived from calibration with tensile and shear material characterization tests. It was observed during determination of these failure parameters that it was difficult to obtain a single set of material parameters that maintained good correlation across the entire suite of material characterization test geometries. Also, it is not well understood how well the model captures the portion of damage due to nucleation. Therefore, to better understand the mechanisms associated with ductile failure (void nucleation, growth, and coalescence) and their relative importance, micromechanical models of these phenomena are investigated. For void nucleation specifically, the decohesion of a metal matrix from an elastic particle in varying tensile loading conditions is modeled with the use of a cohesive zone model based on a traction-displacement relation. The responses obtained from the cohesive zone model are then qualitatively compared with those from a simpler critical stress model to help direct future work on this topic.

1. Introduction. Abnormal mechanical environments for weapons systems are characterized by the unexpected application of mechanical energy. Such events include, for example, drops during handling and crashes during transport. Efforts to model these events for various weapons systems over the last few years (i.e., W80-3, W76 and W87) have been a critical part of nuclear safety assessment. The models are aimed at predicting metal failure of common engineering materials such as steel and aluminum and the subsequent consequences of these failures when applied to screw failures, weld failures, and tearing of component walls. Of particular interest is the possible breech of exclusion regions within the weapon. To this end, extensive experimental data has been collected at both the material and subsystem levels, which has then been simulated using the Sandia-developed EMMI (Evolving Microstructure Model of Inelasticity) plasticity and failure model in the explicit dynamics code LS-DYNA.

Based on the BCJ model [3,4], EMMI [8] is a mechanism-based dislocation model. For its use in nuclear safety assessment, the model is incorporated into continuum finite element models in which entire weapon systems are modeled. It has been developed for a thermo-elasto-viscoplastic polycrystalline metal subjected to isotropic damage and allowing for anisotropic plasticity. It is strain rate, temperature, and load path dependant with normalizing physical constants in order to cast it into a dimensionless form. EMMI contains three definitive internal state variables: a scalar representing statistically stored dislocations, a tensor representing geometrically necessary dislocations, and a scalar for damage. My focus has been on the damage variable, which encompasses void nucleation and growth, and on investigating the uncertainties obtained in the damage parameters across the spectrum of material characterization tests.

*Stanford University
†Sandia National Laboratories, jjdike@sandia.gov

The low-level material characterization tests were used for determining the EMMI parameters. The plasticity model was calibrated for each material based on data from smooth tension tests while the failure model was calibrated using notched tension and double-notched shear tests for void growth and void nucleation parameters respectively. At this point, it was observed that it was difficult to obtain a single set of failure parameters for a given material that correlated with all of the characterization test geometries. Due to this, and due to the desire to better understand the individual mechanisms (void nucleation, growth, and coalescence) at work, micromechanical models of these phenomena are studied. Given the additional uncertainty concerning void nucleation derived from experimental-numerical comparisons and determining that a change was needed in the nucleation damage equation itself, my focus has been this specific phase of the ductile failure process. The decohesion of a metal matrix from an elastic particle in varying tensile loading conditions is modeled with the use of a cohesive zone model and a simpler tied interface critical stress model in order to qualitatively compare the two methods and help direct future work on this topic. Also, a new form of the nucleation damage equation is proposed with initial results reported.

2. The Model. Three materials within the scope of materials being investigated were identified as the most troublesome having the largest variance in their damage parameters for the different test geometries. These were the stainless steels 304L and A286 and the aluminum alloy 7075-T7351. For these materials, information regarding their microstructures, dominant failure mechanisms, and interface properties was collected in an attempt to ascertain any causes for this variance. Along with determining the basic microstructure of each material, specific information regarding the inclusions present and their shape, size, and spacing was desired. Identifying the dominant failure mechanisms includes pinpointing perhaps a dominant phase of the ductile failure process or categorizing the void nucleation as either intergranular or intragranular, which could then lead to the necessity of obtaining the properties of the interface between the bulk material and the inclusions. Useful interface information includes values for fracture energy as well as values for normal and shear critical stresses. Basic microstructure findings are summarized in Table 2.1. Only general bounds for the interface properties of intermetallics are currently available and SEM-aided experiments are recommended for further characterization of the particular materials of interest.

Table 2.1
Material microstructure properties

Material	**Crystal lattice**	**Failure mechanism**	**Inclusions**		
			Particles	**Diameter (mm)**	**Volume fraction (%)**
304L [6]	FCC		silica (SiO_2), chromite ($MnOCrO_3$)	0.63	0.031
A286 [5]	FCC		$Ni_3(Al,Ti)$		
7075 [7]	FCC	interganular decohesion	$(Fe,Mn,Cu)Al_6$, Mg, Si, $CuAl_2$	5.7	2.6
			$Al_{12}Mg_2Cr$	0.05	0.03
			$MgZn_2$	0.025	

From these researched values, and choosing to focus on just one material, a representative micromechanical model is constructed for a spherical inclusion in a cube of aluminum 7075. The intent of this model is to be investigated in tension, compression, and torsion, although this study currently only covers varying stress states of tension. Since stress state has been noted to have a large effect on damage progression, the model is subjected to uniaxial, biaxial, and triaxial tensile loading conditions while also varying the input parameters for each of the interface modeling methods (tied interface versus cohesive zone elements) and briefly investigating the possibility of mesh dependency with two separate meshes (see Figure 2.1).

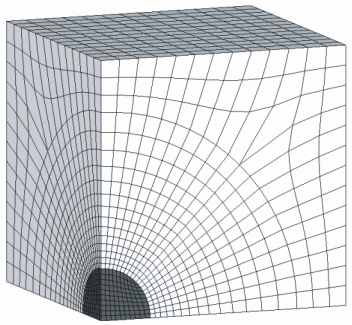


Fig. 2.1. *Mesh*

A simulation matrix of completed results is presented in Table 2.2.

Table 2.2
Simulation matrix

	Tied interface model			**Cohesize zone model**	
	Max/yield stress	**Fracture energy (J/m^2)**	**Mesh**	**Max/yield stress**	**Mesh**
Uniaxial loading	2:2 (normal:shear)	0.5	coarse	2:1 (normal:shear)	coarse
	2:2	1	coarse	2:1	fine
	2:2	1	fine	3:3/2	coarse
	3:3	0.5	coarse	2:2	coarse
	3:3	1	coarse		
Biaxial loading	2:2	0.5	coarse	2:1	coarse
	2:2	1	coarse	2:1	fine
	2:2	1	fine	3:3/2	coarse
	3:3	0.5	coarse	2:2	coarse
	3:3	1	coarse		
Triaxial loading	2:2	0.5	coarse	2:1	coarse
	2:2	0.5	fine	2:1	fine
	2:2	1	coarse	3:3/2	coarse
	3:3	0.5	coarse	3:3/2	fine
	3:3	1	coarse	2:2	coarse

Without experimental data, there is no way to know the accurateness of these chosen values, however this study is an attempt at beginning to bound the constitutive

behavior of void nucleation. Several earlier simulations were performed at varying values with spurious results in order to narrow these parameters down to these reported values.

The tied interface model employs the tied surface-to-surface failure algorithm in LS-DYNA [2] in which one ties the facesets of the two parts at the interface together and then specifies the normal and tangential critical stresses. Failure is then determined by when the condition

$$\left[\frac{\max\left(0, \sigma_{normal}\right)}{FS}\right]^2 + \left[\frac{\sigma_{shear}}{FD}\right]^2 - 1 > 0 \tag{2.1}$$

is met (where FS and FD are the normal and shear failure stresses respectively), the tie breaks, and a void is nucleated. Based on previous work in this area, the range of critical normal stress is taken as two to three times the yield stress of the bulk material and the critical tangential stress is taken as half that of normal. Further discussion of the consequences of this decision is included with the results.

The cohesive zone model used is a traction-displacement relationship developed by Tvergaard and Hutchinson [9] and implemented into Tahoe [1] by Klein in 2001. It was added to LS-DYNA as material 185 in 2005. For this method, a layer of initial zero volume solid elements is created between the inclusion and surrounding matrix merging each respective side with the appropriate part. Similarly, one then specifies the normal and tangential critical stresses and then also inputs a length scale for each of these stresses, which is the distance these elements can grow before failing and nucleating a void. From this information, this algorithm determines the fracture energy of the cohesive elements based on the area under the potential density function shown in Figure 2.2 and given by

$$\phi\left(\lambda\right) = \begin{cases} \sigma_{max}\frac{\lambda}{\Lambda_1} & \lambda < \Lambda_1 \\ \sigma_{max} & \Lambda_1 < \lambda < \Lambda_2 \\ \sigma_{max}\frac{1-\lambda}{1-\Lambda_2} & \Lambda_2 < \lambda < \Lambda_{fail} \end{cases} \tag{2.2}$$

This function is further simplified into a bilinear form by setting Λ_1 and Λ_2 equal

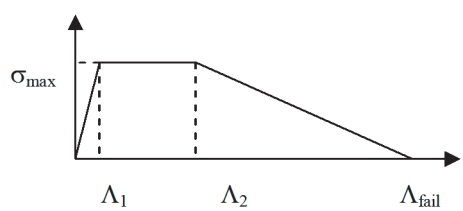


FIG. 2.2. *CZM potential density function*

to each other. The normal and tangential critical stresses in this case are initially set equal. Again, the consequences of this decision as compared with the normal to tangential ratio chosen for the tied interface will be further elaborated on in the results.

Both methods were validated using simple geometries before applying them to this more complicated problem. The tied interface was tested to ensure that it did indeed fail at the appropriate stress combinations while the cohesive zone model was

run with a double cantilever beam problem in order to analytically verify the results. Both methods were deemed as implemented correctly. After moving on to the void nucleation model, the stress progression was studied in order to verify the length of the cohesive zone to ensure proper mesh refinement.

3. Results. The simulations run using a tied interface exhibited a very abrupt failure of all tied contacts at once even in the heavily biased uniaxial loading condition. This is credited to the highly mixed mode response due to the governing equation and the 2:1 ratio of the normal:tangential critical stresses. Quick failure was obtained in all three loading conditions for this ratio, while nucleation was not observed for the uniaxial condition when using a 2:2 ratio. This directly corresponded with the behavior shown by the cohesive zone in this loading condition, which also did not fail. Even with an extremely low fracture energy and extremely high strains, the critical stress of two times yield was too high to reach due to the lack of hardening in this particular material (see Figure 3.1). Unfortunately, changing the ratio for the uniaxial

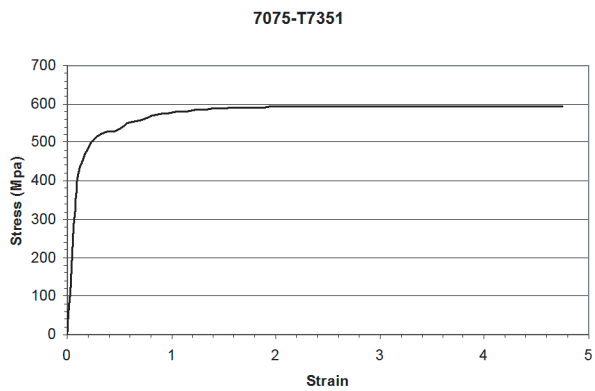


Fig. 3.1. *Stress-strain curve for 7075-T7351*

cohesive zone model to 2:1 does not maintain a correlation between the two methods and more investigation is needed to determine if there is an ideal ratio and how the two methods would compare at that particular ratio.

Additional unresolved problems were discovered with the tied interface algorithm including its necessity to be run on a single processor since it currently lacks parallel processing implementation and its failure to function at the micron scale required for this model. LSTC has plans to fix the former problem in the near future. The latter problem, however, requires further investigation as to the cause and the affect it has had on the results since the tied interface models were run at a different scale than the cohesive zone models. A brief investigation of this affect has yielded a significant difference that will require further study.

The biaxial and triaxial loading conditions were more comparable between the two methods although the cohesive zone model failed later in every case. The cohesive zone model failure was also more systematic demonstrating a “peeling” away behavior as opposed to the “snapping” away behavior of the tied interface. Appropriate behavior was again verified with the potential density function outputs that identically resembled the input.

The existing rate of nucleation equation is directly proportional to the void volume

fraction, ϕ_{total}:

$$\begin{aligned}\phi_{nucleation} =& \nu_0 \left\| D_{in} \right\| \frac{d^{1/2}}{K_{IC} f^{1/3}} \phi_{total} \text{ x} \\ & \left\{ a \left(\frac{4}{27} - \frac{J_3^2}{J_2^3} \right) + b \frac{J_3}{J_2^{3/2}} + c \left\| \frac{I_1}{J_2^{1/2}} \right\| \right\} \exp\left(\frac{-C_{\eta T}}{T} \right)\end{aligned} \tag{3.1}$$

Given a fully dense material (no initial voids), this equation indicates that voids will never nucleate and, subsequently, failure will never occur.

The new proposed rate of nucleation equation is

$$\begin{aligned}\phi_{nucleation} =& \nu_0 \left\| D_{in} \right\| \frac{d^{1/2}}{K_{IC} f^{1/3}} \left[\frac{1}{(1-\phi_{total})^n} \right] \text{ x} \\ & \left\{ a \left(\frac{4}{27} - \frac{J_3^2}{J_2^3} \right) + b \frac{J_3}{J_2^{3/2}} + c \left\| \frac{I_1}{J_2^{1/2}} \right\| \right\} \exp\left(\frac{-C_{\eta T}}{T} \right),\end{aligned} \tag{3.2}$$

which maintains the nonlinear behavior of the original form as per experimental observation, but eliminates the problem of having no void nucleation when the initial void volume fraction is zero. This is, of course, a guess at a possible modification and is still subject to further understanding of the micromechanical model.

Several simple tension and shear simulations were run with both the old and the new implementations to verify that the appropriate correction was made. Next, the new implementation will be run at the subsystem level in order to identify any improvements or deteriorations in the results.

4. Conclusions. A micromechanical model has been developed for modeling the decohesion of a metal matrix from an elastic particle under varying tensile loading conditions using both a tied interface and a cohesive zone. Both were run using similarly defined parameters in order to qualitatively compare the two. It would be beneficial to use a tied interface model for future work since it is easy to define and does not require any additional elements. However, there are two key issues to resolve before this is a possibility: its inability to function at the micron scale and its failure of the entire interface at once, rather than a growing detachment from the particle, and whether or not this relates to the normal:tangential stress ratio. In comparison, the cohesive zone model is much more difficult to define requiring some knowledge of realistic values of the fracture energy in order to properly distribute the energy dissipation given the phenomenological nature of cohesive zone models in general. It also requires extra elements and a certain amount of mesh refinement to resolve the cohesive zone itself. Otherwise, it has been verified to be mesh-independent, have a more intuitive failure response, and work at any scale. Experimental data would be very valuable in determining appropriate values for this cohesive zone model, which could then perhaps be represented via a simpler tied interface model.

Determining the appropriate model to use on this relatively simple problem is critical to improving the nucleation equation and in developing more complicated micromechanical models that would include void nucleation, growth, and coalescence. These would obviously require multiple inclusions in order to observe the interaction that takes place and would ideally include inclusions of varying sizes, shapes, orientations, and spacing. From there, the gained knowledge is to be applied to the subsystem and system level models so that simulations can accurately replace more expensive physical testing for nuclear safety assessments.

REFERENCES

[1] *Tahoe*, 2001. Sandia National Laboratories, http://tahoe.ca.sandia.gov/.

[2] *LS-DYNA*, 2005. Livermore Software Technology Corporation, Version 971.

[3] D.J. Bammann, *Internal variable model of viscoplasticity*, Int. J. Eng. Sci., 22 (1984), pp. 1041–1053.

[4] ———, *Modeling temperature and strain rate dependant large deformations of metals*, Appl. Mech. Rev., 1 (1990), pp. 312–318.

[5] J. A. Brooks and A. W. Thompson, *Microstructure and hydrogen effects on fracture in the alloy a286*, Metall. Trans., 24 (1993).

[6] S. J. Goodwin, F. W. Noble, and B. L. Eyre, *Inclusion nucleated ductile fracture in stainless steel*, Acta Metall., 37 (1989), pp. 1389–1398.

[7] G. T. Hahn and A. R. Rosenfield, *Metallurgical factors affecting fracture toughness of aluminum alloys*, Metall. Trans., 6A (1975), pp. 653–668.

[8] E. B. Marin, D. J. Bammann, R. A. Regueiro, and G. C. Johnson, *On the formulation, parameter identification and numerical integration of the emmi model: Plasticity and isotropic damage*, Tech. Report SAND2006-0200, Sandia National Laboratories, January 2006.

[9] V. Tvergaard and J. W. Hutchinson, *The relation between crack growth resistance and fracture process parameters in elastic-plastic solids*, J. Mechanics and Physics of Solids, 40 (1992), pp. 1377–1397.

THERMAL CONDUCTIVITY OF CARBON NANOTUBES USING MOLECULAR DYNAMICS SIMULATIONS

A. HENRY* AND S. PLIMPTON†

Abstract. Carbon nanotubes (CNTs) are known to exhibit excellent mechanical and transport properties. In particular the thermal conductivity of individual CNTs may be higher than that of any known material. Although there has been considerable study devoted to measurement and prediction of CNT thermal conductivity, many questions remain. In this study we implemented a new interatomic potential in the large-scale atomic/molecular massively parallel simulator (LAMMPS), that will allow for the study of multi-walled nanotubes. We intend to use equilibrium molecular dynamics simulations to determine reliable (converged) thermal conductivity values for a wide variety of CNTs. This work is intended to address the temperature, chirality, diameter and length dependencies that have been debated in the literature.

1. Introduction. As some of the most promising nanostructures, carbon nanotubes (CNTs) have stimulated significant scientific interest because of their attractive material properties. Their unique symmetries give CNTs excellent electrical and thermal conductivities, which generally depend on diameter and chirality of the tube. Because of their high bonding strength and one-dimensional character, CNTs are thought to have the highest thermal conductivity of all known materials. This possibility has generated strong interest within the heat transfer community, where CNTs could be used in synthesizing new materials to enhance heat conduction. This is particularly important for industries, such as microelectronics, where heat dissipation has become a bottleneck to continued miniaturization of processor components.

We are aware of eight [1, 3, 7–9, 12–14] simulation studies and four [4–6, 11] experimental studies in the literature that have addressed the topic of CNT thermal conductivity. Of the four experimental studies, two have measured the thermal conductivity of an individual tubes (2000 - 3000 W/m-K). The variety of experimental techniques and specimens employed in other studies have generated estimates for the thermal conductivity of individual tubes between 200 and 3000 W/m-K. The lack of quantitative agreement amongst the different studies has rendered the thermal conductivity of an individual CNT unresolved. Eight different modeling studies used molecular dynamics (MD) simulations to calculate thermal conductivity, but in all cases the approaches and results show little quantitative agreement. Some results show diverging thermal conductivity with increasing tube length, while others show the opposite. One study indicates that the thermal conductivity also diverges with increasing temperature, while another study shows a peak suggesting that umklapp scattering begins to dominate above room temperature. One study showed strong dependence on tube chirality, while another showed no dependence on chirality. As a result, questions surrounding the geometric dependencies and dominating mechanisms in CNT thermal transport are unresolved, while the lack of quantitative agreement leads to estimates between 1700 and 6600 W/m-K.

One common thread throughout all the modeling studies [1,3,7–9,12–14] has been the study of single walled nanotubes (SWCNTs). Multi-walled nanotubes (MWCNTs) have not been addressed, partly due to computational constraints and the lack of suitable potentials that can describe the interactions between walls. As a result none of the previous simulation studies have been able to compare directly to the experimental

*Massachusetts Institute of Technology, ase@mit.edu
†Sandia National Laboratories, sjplimp@sandia.gov

measurement of an individual tube's thermal conductivity.

Two main approaches have been used in simulating the thermal conductivity of CNTs. The first uses velocity rescaling to induce a kinetic energy or temperature gradient and calculates thermal conductivity in accordance with Fourier's law. It has been argued that this approach, however, introduces artificial phonon scattering where the rescaling conditions are applied and would lead to a reduced thermal conductivity. The second approach uses the formula developed by Green and Kubo, which is based on the system's linear response to an external perturbation:

$$\kappa_{\alpha,\beta} = \frac{V}{k_B T^2} \int_0^\infty \langle Q_\alpha(t) Q_\beta(0) \rangle dt \tag{1.1}$$

$\kappa_{\alpha,\beta}$ is the thermal conductivity tensor, $\langle Q_\alpha(t) Q_\beta(0) \rangle$ is the heat flux autocorrelation function, V is the system volume, k_B is Boltzmann's constant and T is the system temperature. This method has proved accurate for a variety of materials, but has the drawback of requiring long simulation times for convergence. One possible reason for lack of quantitative agreement amongst the simulation studies, is that each study used a different simulation length. In applying the Green-Kubo method, the resulting thermal conductivity can vary by 100% before converging to the value obtained at longer times. One study developed a hybrid method, based on the Green-Kubo formulation, which adds a driving force to further disturb the system, resulting in faster convergence.

2. Actual Content. In this study we intend to produce a wide variety of results for both SWCNTs and MWCNTs using LAMMPS, to provide reliable quantitative calculations and explanation of the mechanisms involved in CNT thermal transport. One of the most common potentials used in the various MD studies was developed by Brenner [2], which has shown good agreement with experiments. Although Brenner's potential is well suited for short range carbon bonds within a single nanotube, the cut off function would not allow for interactions between walls of a MWCNT. Recently, this constraint has been overcome by Stuart [10], who has adapted Brenner's potential to include longer range Van der Waals interactions with the AIREBO potential. Using AIREBO [10] we can simulate SWCNTs as well as MWCNTs to compare thermal conductivities and test length, temperature and chirality dependence. The advantage of adding the potential to LAMMPS is that larger simulations can be done in parallel, allowing us to better test the convergence with CNT length, diameter as well as the number of walls. To further discuss the various implementation issues involved with adding the AIREBO potential to LAMMPS, we first provide an overview of the AIREBO potential to highlight its features.

AIREBO consists of three terms and contains parameters to describe energy E of a hydrocarbon system.

$$E = \frac{1}{2} \sum_{i,j} (REBO + LJ + TORS) \tag{2.1}$$

The term labeled REBO has the same structure as Brenner's revised potential, with exponentials for the repulsive and attractive components.

$$REBO = V^R + b_{ij} V^A \tag{2.2}$$

where

$$V^R = w_{ij}[1 + \frac{Q_{ij}}{r_{ij}}] A_{ij} e^{-\alpha_{ij} r_{ij}} \tag{2.3}$$

$$V^A = -w_{ij} \sum_{n=1}^{3} B_{ij}^{(n)} e^{-\beta ij^{(n)} r_{ij}} \tag{2.4}$$

The strength of the bond between a pair of atoms is dictated by the bond order. The bond order expression contains four terms with many-body dependence.

$$b_{ij} = \frac{1}{2}(p_{ij} + p_{ji}) + \pi^{RC} + \pi^{DH} \tag{2.5}$$

Other than explicit use of the interatomic distances, the terms of the bond order depend on five intermediate variables with many-body dependence.

$$N_{ij} = \sum_k w_{ik} - w_{ij} \tag{2.6}$$

$$N_{ji} = \sum_l w_{jl} - w_{ij} \tag{2.7}$$

$$N_{ij}^{conj} = 1 + (\sum_k \delta_{kC} w_{ik} Sp(N_{ki}))^2 + (\sum_l \delta_{lC} w_{jl} Sp(N_{lj}))^2 \tag{2.8}$$

$$\cos\theta_{jik} = \frac{\tilde{\mathbf{r}}_{\mathbf{ji}} \cdot \tilde{\mathbf{r}}_{\mathbf{ki}}}{|\tilde{\mathbf{r}}_{\mathbf{ji}}||\tilde{\mathbf{r}}_{\mathbf{ki}}|} \tag{2.9}$$

$$\cos\omega_{kijl} = \frac{\tilde{\mathbf{r}}_{\mathbf{ji}} \times \tilde{\mathbf{r}}_{\mathbf{ik}}}{|\tilde{\mathbf{r}}_{\mathbf{ji}} \times \tilde{\mathbf{r}}_{\mathbf{ik}}|} \cdot \frac{\tilde{\mathbf{r}}_{\mathbf{ij}} \times \tilde{\mathbf{r}}_{\mathbf{jl}}}{|\tilde{\mathbf{r}}_{\mathbf{ij}} \times \tilde{\mathbf{r}}_{\mathbf{jl}}|} \tag{2.10}$$

w_{ik} is the cut off function which terminates interactions between carbon atoms further than 2 Å (subscripted indices indicate the interatomic distance involved), which is written in terms of a more general cut-off function S'.

$$w_{ij} = S'(t_{r_{ij}}) \tag{2.11}$$

where

$$t_{r_{ij}} = \frac{r_{ij} - r_{min}}{r_{max} - r_{min}} \tag{2.12}$$

S' is a cut off function written in terms of a generic nondimensional variable t, such that $S' = 1$ for $t <= 0$, $S' = 0$ for $t >= 1$ and transitions smoothly in between for $0 < t < 1$. The quantity t is defined using a minimum and maximum value for the argument of the S' cut off function. The minimum and maximum values for each variables used in S' were treated as parameters when the potential was fit. A second cut-off function $S(t)$ is also used in AIREBO and only differs from $S'(t)$ in the way it transitions from 1 to 0 on the interval $0 < t < 1$.

The quantities p_{ij} and p_{ji} contain three body summations over triplet angles, which are then used to evaluate a fifth order spline g.

$$p_{ij} = (\sum_k w_{ik} g(\cos\theta_{jik}, Nij) e^{\lambda_{jik}} + P_{ij}(N_{ij}^C, N_{ij}^H))^{-1/2} \tag{2.13}$$

P_{ij} is a two dimensional correction spline that modifies p_{ij} and/or p_{ji} for specific hydrocarbon configurations. The corresponding angular function p_{ji} is defined by reversing the indicies and summing over the neighbors of the atom labeled j.

The AIREBO potential was particularly designed as a reactive potential. As a result a few different splines with many-body dependence are used in specific locations to reproduce interactions for a wide variety of hydrocarbon configurations. AIREBO is written in a symmetric form, so that any interaction between two atoms is equivalent to the interaction when the ij indicies are reversed. The second term in the bond order is π^{RC}, which is a tricubic spline that depends on the coordination numbers N_{ij} and N_{ji} and N_{ij}^{conj}. The last term π^{DH} depends uses another tricubic spline T_{ij} to determine whether the four-body summation over dihedral angles should be included. The dihedral term is written as

$$\pi^{DH} = T_{ij} \sum_{k} \sum_{l \neq k} (1 - \cos^2 \omega_{kijl}) w_{ik} w_{jl} \tag{2.14}$$

These four terms define the bond order between any two atoms and can induce forces on atoms two bonds away from the ij bond. This extensive many-body dependence creates efficiency issues when evaluating the potential in parallel.

The major advancement of AIREBO is the inclusion of long-range interactions through the adaptive LJ potential. The LJ potential was chosen over other long-range forms, because of its simplicity and established accuracy in describing a variety of systems. The LJ potential does not include a long range cut off function, because it is designed for intramolecular interactions and is suitable for describing interactions between a wide variety of hydrocarbon molecules. The drawback to the LJ potential however is that it uses a complicated set of short range cut off functions, to shield the atoms involved in bonds described by the REBO term, from experiencing the steep r^{12} repulsion. To screen this repulsion AIREBO uses a combination of three switching functions.

$$S_{tr} = S(r_{ij}) \tag{2.15}$$

$$S_{tb} = S(b_{ij}^{*}) \tag{2.16}$$

$$C_{ij} = 1 - max[w_{ij}, w_{ik} w_{kj} (\forall k), w_{ik} w_{kl} w_{lj} (\forall k, \forall l)] \tag{2.17}$$

The adaptive LJ potential is then written as

$$LJ = S_{tr} S_{tb} C_{ij} V^{LJ} + [1 - S_{tr}] C_{ij} V^{LJ} \tag{2.18}$$

where,

$$V^{LJ} = 4\epsilon_{ij} [(\frac{\sigma_{ij}}{r_{ij}})^{12} - (\frac{\sigma_{ij}}{r_{ij}})^{6}] \tag{2.19}$$

so that the LJ potential is fully "on" for large separations and is modified at shorter distances using the three switching functions. The major computational expense comes from evaluating b_{ij}^{*} for pairs in the switching region. b_{ij}^{*} is the same as the REBO bond order except that the distance r_{ij} is adjusted to the inner cut off of the REBO potential.

The last term of the AIREBO potential is an explicit torsional term, which sums over all combinations of four atoms.

$$TORS = \sum_k \sum_{l \neq k} (\frac{256}{405} \epsilon_{kijl} (\cos^{10} \frac{\omega_{kijl}}{2}) - \frac{1}{10} \epsilon_{kijl}) w_{ik} w_{ij} w_{jl} \tag{2.20}$$

This term uses the same cut off as the REBO potential and is designed to provide rotational barriers for hydrocarbon molecules. For most condensed matter systems, the torsional and LJ contributions are heavily outweighed by the REBO potential, yet they are needed to describe hydrocarbon intermolecular forces and reactions.

To implement the AIREBO potential, we were very fortunate to receive a copy of Stuart's energy and force routines, which were written in Fortran. LAMMPS however is written in C++ and a new/translated routine was needed. The AIREBO potential form can be written explicitly, however the splines require numerical interpolation when atoms are partially coordinated. This presented serious issues when debugging because the spline knots are provided, but the spline coefficients have to be generated from many interpolation points.

To implement AIREBO in parallel, information about atoms that are owned on neighboring processors must be exchanged. The extensive many-body dependence through the conjugated coordination number implies that atoms up to 8 Å can experience forces as part of the LJ term due to many-body effects. This requires that the positions of atoms within a buffer region of either 8 Å or a user specified LJ long-range cut off (depending upon which is larger), be communicated within the LAMMPS framework. This requires that positions and forces for atoms within the buffering region be updated after every time step.

Our current status is that we have a preliminary implementation of the AIREBO potential in LAMMPS and are testing various configurations to check energies and forces. The remaining steps are to include the calculation of the heat flux vector (needed to evaluate the Green-Kubo expression), generation of CNT geometries as input files and to find ways to reduce the computational expense. Existing code written for TOWHEE can be used to generate the atom coordinates for any size and chirality of CNT. However a few scripts will be needed to reorganize the coordinate outputs as LAMMPS input files.

The many-body dependence in the LJ switching function Stb makes the LJ term of the AIREBO potential the most computationally expensive. One approach to increasing the speed would be to trade-off computation for increased memory usage. Explicit evaluation of the various splines contained in the bond order function is very expensive. This could be alleviated by storing spline values for particular pairs of atoms between time steps. Saving the information until the next time step would require additional book keeping procedures, but could cut the computational cost. After the heat flux vector is incorporated into the force routines, a simple set of analysis routines can be written to determine the heat flux autocorrelation function. The heat flux autocorrelation function can then be integrated to determine the thermal conductivity. Following these steps we can begin testing a variety of systems to understand thermal transport in CNTs.

3. Conclusions. Several additional steps are required to begin simulating MD trajectories for CNTs, however the largest challenge of implementing the potential has been overcome. The remaining challenges are related to analysis routines, input file generation and increasing efficiency. We expect the remaining tasks to be straight forward and hope to begin testing various types of CNTs within the coming weeks.

REFERENCES

[1] S. Berber, Y. Kwon, and D. Tomanek, *Unusually high thermal conductivity of carbon nanotubes*, Phys. Rev. Lett., 84 (2000).

[2] D. Brenner, *Reactive bond order potential for hydrocarbons*, Phys. Rev. B., 46 (1992), p. 1948.

[3] J. Che, T. Cagin, and W. Goddard III, *Thermal conductivity of carbon nanotubes*, Nanotechnology, 65 (2000).

[4] M. Fujii, X. Zhang, H. Xie, H. Ago, K. Takahashi, T. Ikuta, H. Abe, and T. Shimizu, *Measuring the thermal conductivity of a single carbon nanotube*, Phys. Rev. Lett., 95 (2005), p. 065502.

[5] J. Hone, M. Llaguno, N. Nemes, A. Johnson, J. Fischer, D. Walters, M. Casavant, J. Schmidt, and R. Smalley, *Electrical and thermal transport properties of magnetically aligned single wall carbon nanotube films*, Appl. Phys. Lett., 77 (2000), p. 666.

[6] P. Kim, L. Shi, A. Majumdar, and P. McEuen, *Thermal transport measurements of individual multiwalled nanotubes*, Phys. Rev. Lett., 87 (2001), p. 215502.

[7] S. Maruyama, *A molecular dynamics simulation of heat conduction in a carbon nanotube*, Physica B, 323 (2002), p. 193.

[8] J. Moreland, J. Freund, and G. Chen, *The disparate thermal conductivity of carbon nanotubes and diamond nanowires studied by atomistic simulation*, in Symposium on Micro/Nanoscale Energy Conversion and Transport, MECT, 2002.

[9] M. Osman and D. Srivastava, *Temperature dependence of the thermal conductivity of single-wall carbon nanotubes*, Nanotechnology, 12 (2001), p. 21.

[10] S. Stuart, A. Tutein, and J. Harrison, *A reactive potential for hydrocarbons with itermolecular interactions*, J. Chem. Phys., 112 (2000), p. 6472.

[11] D. Yang, Q. Zhang, G. Chen, S. Yoon, J. Ahn, G. Wang, and J. Li Q. Zhou, Q. Wang, *Thermal conductivity of multiwalled nanotubes*, Phys. Rev. B., 66 (2002), p. 165440.

[12] Z. Yao, J. Wang, B. Li, and G. Liu, *Thermal conduction of carbon nanotubes using molecular dynamics*, Phys. Rev. B, 71 (2005), p. 085417.

[13] G. Zhang and B. Li, *Thermal conductivity of nanotubes revisited: Effects of chirality, isotope, impurity, tube length, and temperature*, J. Chem. Phys., 123 (2005), p. 114714.

[14] W. Zhang, Z. Zhu, F. Weng, T. Wang, L. Sun, and Z. Wang, *Chirality dependence of the thermal conductivity of carbon nanotubes*, Nanotechnology, 15 (2004), p. 936.

COUPLING MOLECULAR DYNAMICS WITH FINITE ELEMENTS AND THE ENTAGLEMENT IN POLYMER BRUSHES

ROBERT S. HOY*, STEVEN J. PLIMPTON†, AND GARY S. GREST‡

Abstract. Our research foxed on two separate projects. The "major" project, mentored by Steve Plimpton, was to implement a scheme devised by Mark Robbins for coupling molecular dynamics (MD) and finite element (FE) calculations. The MD code used was the Sandia LAMMPS code. The question of how to implement the finite element part of the project was initially open, and coming to a decision took up much of the summer. We finally decided to write a simple FE code of our own. Currently, we have implemented almost all of the coupling algorithm for the MD "end" of the code, but the FE end is far from complete. The "minor" project, mentored by Gary Grest, was to investigate entanglement in polymer brushes and melt-brush systems. This investigation was carried out using LAMMPS MD simulations on Sandia's institutional clusters. Currently we have finished the first round of simulations, have performed preliminary data analysis, and are planning additional simulations. These simulations are "exploratory" in that the results will be used to identify the physical questions we wish to answer. This project will be continued after I return to Johns Hopkins and will form part of my dissertation work.

1. Molecular Dynamics — Finite Element Coupling. Many problems in the sciences and engineering involve coupling of phenomena with a wide range of characteristic length and time scales. It is often difficult or impossible to treat these problems with a single simulation technique. For example, molecular dynamics is an excellent tool for studying the behavior of systems at the atomic level. However, the largest volumes which can be simulated with atomistic MD on modern-day supercomputers are of order $.01\mu m^3$, and such large volumes can only be simulated over a time of order 1 nanosecond. This maximum volume is less than the volume of even medium-sized nanostructures, and the time scale is far less than that of many physical processes of interest.

The finite element method is the standard tool for simulating larger systems and longer time scales. It is capable of treating essentially infinitely large systems and long time scales on the "high end". However, FE approaches usually break down on the "low end" when the length scales of interest (the scale over which the fields vary "significantly") approach atomic dimensions, or in other words the scale at which matter can no longer be regarded as a continuum.

In many cases, problems involve phenomena on atomic length scales but also on length scales too large to be treated with MD. An example is shown in Figure 1.1; the contact of a flat surface with a rough surface [4]. As the flat surface (represented by the green row of atoms) is lowered onto the rough surface, the region near the top of the rough surface deforms plastically. Atomic bonds are broken and rearrangement of neighbors occurs. This phenomenon is very difficult to capture with FE and so a MD solution is desirable. At the same time, however, long-range elastic fields extend hundreds of atomic diameters below the rough surface. The problem shown is 2-dimensional, so it is possible to treat these long-range fields atomistically, but it is also wasteful, and would be very expensive in 3 dimensions. The FE method, on the other hand, can treat the elastic fields both accurately and efficiently.

In this and other cases it is therefore advantageous to couple MD and FE simulations. However, there are many difficulties in doing so, and so MD-FE coupling is

*Johns Hopkins University, robhoy@pha.jhu.edu
†Sandia National Laboratories, sjplimp@sandia.gov
‡Sandia National Laboratories, gsgrest@sandia.gov

an active field of research. A MD-FE coupling scheme must satisfy the following:

1. The MD must provide a boundary condition for the FE.
2. The FE must provide a boundary condition for the MD.

Satisfaction of these is challenging due to the fundamentally different nature of the methods. MD is a tool for time integration of Newton's equations of motion for mobile particles, while FE is a tool for the solution of partial differential equations on a (in the simplest form of the method) fixed grid. Boundary conditions are often quite complicated in FE, but are usually simple in MD.

One promising MD-FE coupling scheme, known as the "hybrid method", was developed by others in the group of my academic adviser, Mark Robbins [4]. The coupling scheme is illustrated in Fig. 1.1. The boundary condition for the FE solution (at the red nodes) is obtained from the displacements of MD atoms. Displacements of atoms within the pink circle are time- and space-averaged to set the displacement of the enclosed red node. Then the positions of the other nodes are solved for, and this provides a boundary condition for the (purple) MD atoms. Displacements of the purple atoms are fixed affinely by the time-extrapolated displacements of the yellow nodes. The procedure is described in detail in Ref. [4].

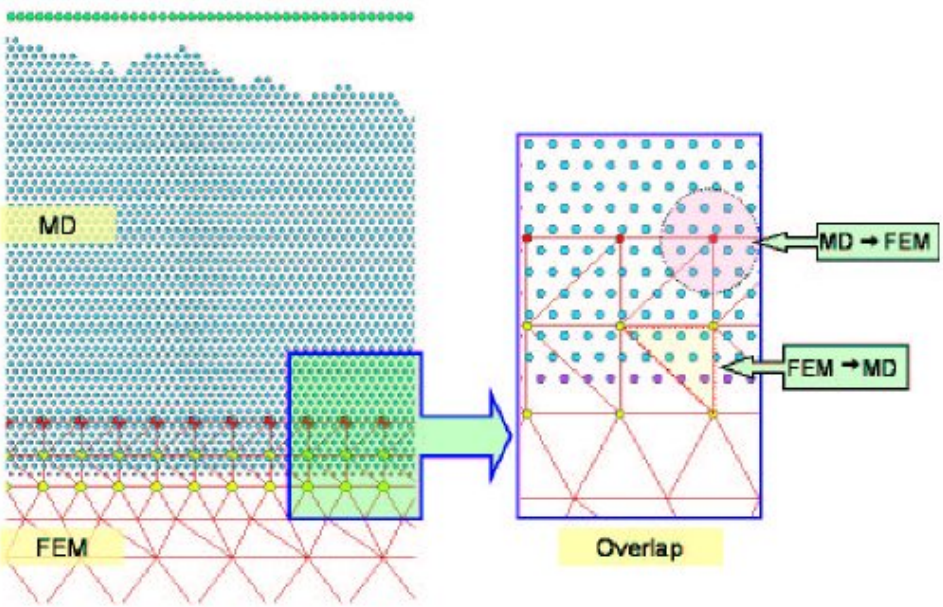


Fig. 1.1. *The molecular dynamics - finite element coupling scheme developed in Ref. [4]. Standard MD atoms are blue, while boundary MD atoms are purple. Boundary FE nodes are red, while affine-coupling FE nodes are yellow. The overlap region lies between a line just above the red nodes (coincident with the top of the pink circle) and the bottom row of yellow nodes.*

The hybrid method has been implemented [4] in two dimensions using a simple FORTRAN public domain FE code known as Flagshyp (for "Finite element Large Strain Hyperelasticity"), and the older, FORTRAN version of the LAMMPS MD code. My goal for this summer was to implement the hybrid method in three dimensions using the newer, C++ version of LAMMPS.

Implementing the coupling scheme with an artificial FE "solution" proved to be relatively easy once I had gained sufficient knowledge of the LAMMPS code. However,

implementing a real FE solver has proven to be quite difficult. Our initial plan was to interface LAMMPS with Sandia's TAHOE FE code. TAHOE is a parallel code with a wide variety of constitutive models and FE solvers. Consulting with TAHOE developers, we discovered that creating a fully functional LAMMPS-TAHOE interface was too ambitious an effort for the scope of this summer project. Instead, we decided to implement a simple subset of FE functionality directly in LAMMPS: a C^0 piecewise-linear Lagrangian formulation for triangle- and tet-elements with explicit Newmark time integration.

We also implemented a constitutive model present in Flagshyp and used in the FE regions of systems as depicted in Figures 1.1 and 1.2, which is the orthotropic nonlinear (quadratic) elasticity (ONE) model. Choice of this model is dictated by the following: we do not require a plastic model, because the regions where plastic deformation occurs are best treated with MD. This is also a time savings for coupled MD/FE formulations, since the linear elastic model typically used in FE deformation codes would require the atomistic region to be too large.

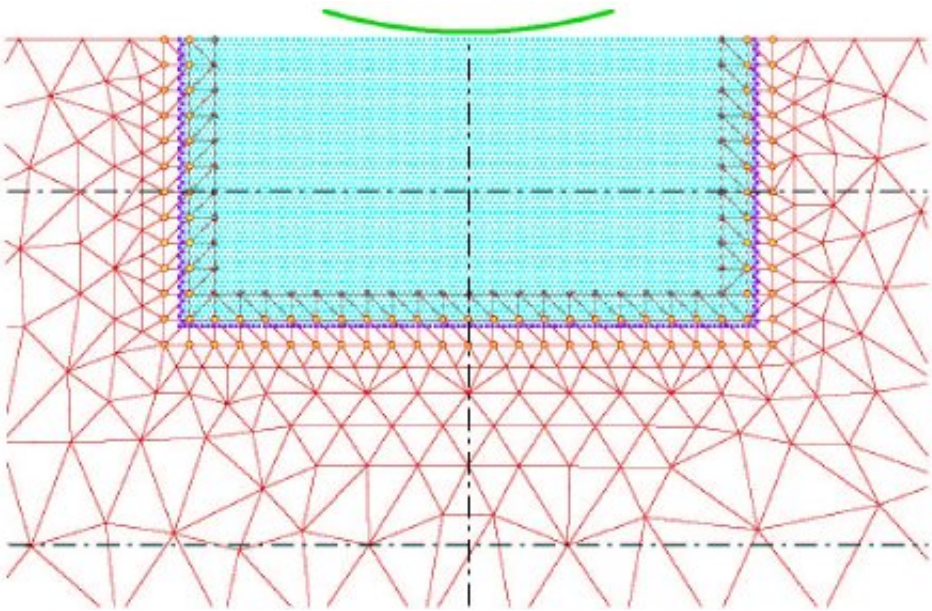


FIG. 1.2. *Schematic of another problem treated in Ref. [4]. A cylindrical punch (green) was brought into contact with a flat atomistic surface (blue). Stress fields were measured along various vertical and horizontal profiles, e.g. the dash-dotted black lines, and compared to predictions from continuum mechanics. The area of the atomistic region was approximately* $100\sigma \times 50\sigma$*, while the area of the region including FE nodes was* $400\sigma \times 200\sigma$*. The overlap region is indicated by purple atoms and by red and yellow nodes. Periodic boundary conditions were applied in the horizontal direction. For this problem the computational cost of the hybrid method was less than* 10% *of the cost of a fully atomistic solution.*

Our current status is that we hope to have a 2D FE solver working by the end of the summer. We still need to add functions for evaluating the strains on the elements, converting these to stresses using the ONE constitutive model, converting the stresses into forces on the nodes, and performing Newmark integration to obtain nodal displacements. Remaining tasks on the "LAMMPS side" of the project include coding in the weights of the yellow nodes on the displacements of the purple atoms

in 3 dimensions, and building in the capability to write restart files.

2. Entanglement in Polymer Brushes. A polymer brush is formed by polymer chains grafted at one end to a substrate. These brushes are often used as adhesion promoters. Adhesive bonds are often formed using a layer of entangled polymer confined between the adhered surfaces. When a brush is attached to one or both of these surfaces prior to the introduction of the adhesive, adhesion is enhanced. This is due to "entanglement" of the brush with the adhesive. Entanglements are topological constraints on chain motion arising from the connectivity and random-walk-like structure of polymer chains.

Adhesive failure in a brush-enhanced system typically occurs through one or more of three mechanisms; brush chain pullout, brush chain scission, and crazing. The latter two are depicted in Figures 2.2–2.4. Which method of failure occurs is a function of the degree to which brush and adhesive are entangled. This in turn is a function of the length of the brush chains and the "coverage" density Σ with which they are attached to the adherend surface.

Figure 2.1 schematically depicts how brush chain length and coverage affect the failure mode and work of adhesion. If the brush chains are not sufficiently long to be entangled with the adhesive, they simply pull out under load. Longer chains are broken (via "scission" of backbone covalent bonds) if the brush coverage is low, or, at moderate coverages, when adhesion strength is greatest and the brush and adhesive are most entangled, form "crazes." At still higher coverages, polymer melts and brushes phase-separate for entropic reasons; entanglement and adhesion strength decline.

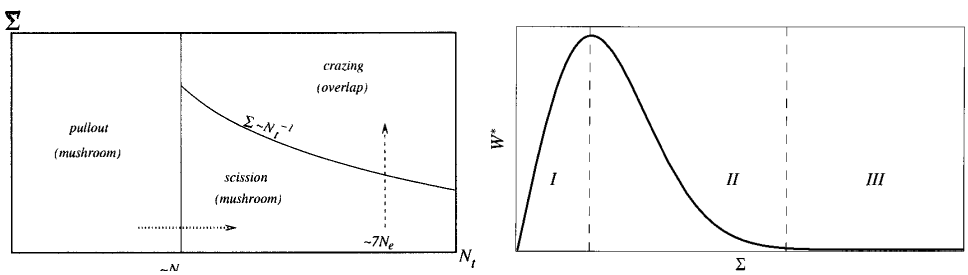


FIG. 2.1. *(Left) Schematic "phase" diagram for breaking of an adhesive brush-melt system. The horizontal axis is chain length N. The transition from pullout to scission or crazing takes place at $N = N_e$. The vertical axis indicates brush coverage (increasing tether density going upwards). Figure is taken from Ref. [6]. (Right) Schematic of work of adhesion W^* for a brush-melt system. (I) mushroom regime (II) partially overlapped regime (III) phase-separated regime. Σ indicates brush coverage. Figure is taken from Ref. [5].*

While this is qualitatively known, little is known quantitatively about the details of entanglement in brushes and brush-adhesive systems. For this reason Gary Grest and I have been studying entanglement in model systems this summer using a new simulation technique [1]. We employ a coarse-grained bead-spring polymer model [3] that incorporates key physical features of linear homopolymers such as covalent backbone bonds, excluded-volume and adhesive interactions, and the topological restriction that chains may not cross. All monomers interact via the truncated Lennard-Jones potential:

$$U_{LJ}(r) = 4u_0\left(\left((\tfrac{\sigma}{\sigma})^{12} - (\tfrac{\sigma}{r_c})^{12}\right) - \left((\tfrac{\sigma}{r})^{6} - (\tfrac{\sigma}{r_c})^{6}\right)\right), \tag{2.1}$$

where r_c is the potential cutoff radius, $U_{LJ}(r) = 0$ for $r > r_c$, u_0 is the binding energy, and σ is the effective atomic diameter.

Covalent bonds between adjacent monomers on a chain are modeled using the finitely extensible nonlinear elastic (FENE) potential

$$U_{FENE}(r) \quad = \quad -\frac{kR_0^2}{2}\ln(1-(\mathrm{r}/\mathrm{R}_0)^2), \tag{2.2}$$

with the canonical parameter choices [3] $R_0 = 1.5a$ and $k = 30k_BT/\sigma^2$. The equilibrium bond length $l_0 \simeq 0.96a$. N_t monomers are bound together to make linear brush chains, while N_m monomers similarly form the melt chains.

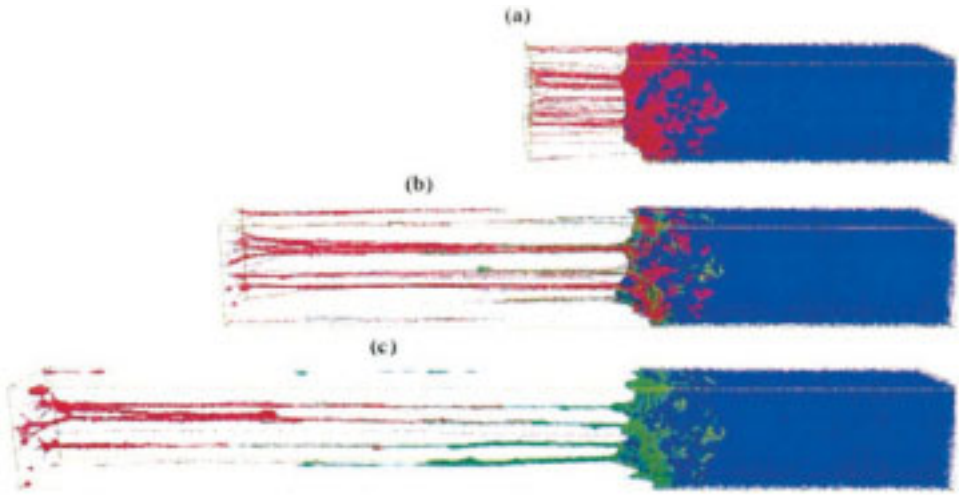


FIG. 2.2. *Breakdown of a glassy adhesive system by scission of the brush chains. The glass (formed from the untethered chains) is blue, portions of brushes still attached to the tether surface are red, and brush chains which have broken off into the glass are green. The coverage is* $\Sigma = .02/\sigma^2$. *Figure is taken from Ref. [6].*

We study the entanglement structure using primitive path analysis (PPA) [1]. A primitive path is the shortest path a chain fixed at its ends can take without crossing any other chains. In PPA, all chain ends are fixed in space and several changes are made to the interaction potential. Intrachain excluded-volume interactions are deactivated, while interchain excluded-volume interactions are retained. The system is coupled to a heat bath at $T = 0.001u_0/k_B$ so that thermal fluctuations are negligible, and the equations of motion are integrated until the chains minimize their length. Turning off intrachain interactions means that self-entanglements are not preserved. We believe their number is negligibly small for the systems considered here, but need to verify this.

Excluded volume appears nowhere in Edwards' standard concept of primitive paths - the paths are envisioned as a network of infinitely thin line segments between entanglement points (EPs). At the conclusion of a typical PPA run, however, excluded volume remains important because of the finite thickness of the chains. The percentage of beads with interchain contacts remains high, and not all of the interchain contacts are EPs. To identify individual entanglements along a chain, an addition must be made to the standard PPA procedure. At the conclusion of the standard PPA, we insert $n_{dec} - 1$ beads between each bead of the existing chains,

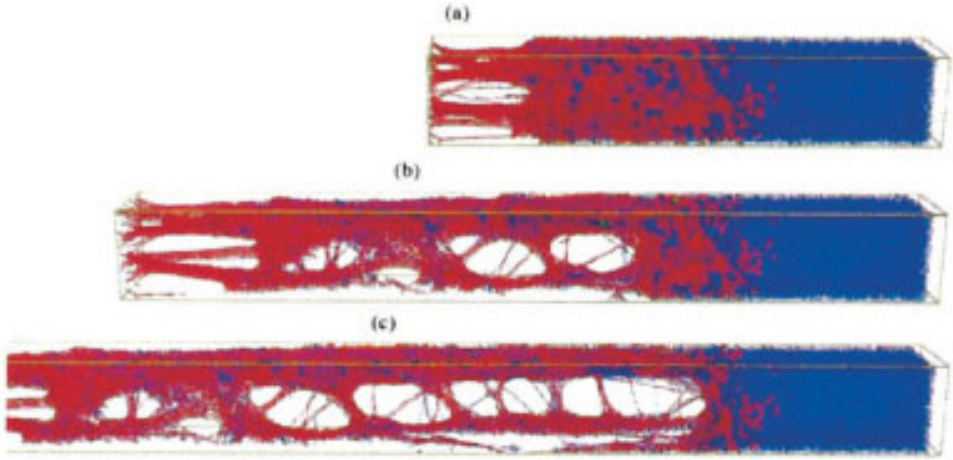


FIG. 2.3. *Crazing of a glassy adhesive system. Colors are as above. The coverage is* $\Sigma = .10/\sigma^2$. *Figure is taken from Ref. [6].*

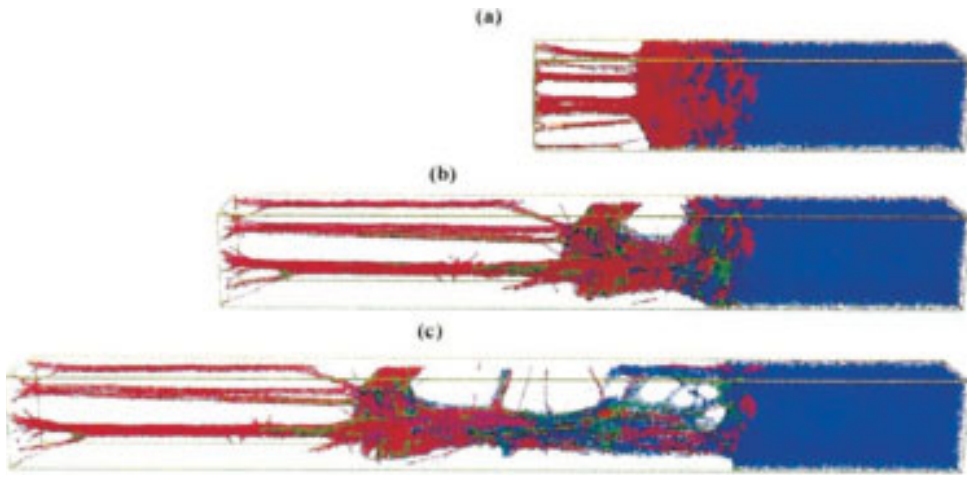


FIG. 2.4. *Mixed scission and crazing. Colors are as above. The coverage is* $\Sigma = .05/\sigma^2$. *Figure is taken from Ref. [6].*

reduce the bead diameter by a factor of n_{dec}, and allow the chains to minimize their length once again. This reduces the fraction of beads having interchain contacts by a factor of roughly n_{dec}, allowing us to identify interchain contacts as EPs.

I now discuss some preliminary results of our PPA analysis of the brush-melt systems. Figure 2.5 depicts the number of brush-brush, brush-melt, and melt-melt interchain contacts for the thin-chain, $\Sigma = .03/\sigma^2$ system. The horizontal axis in Figures 2.2-2.4 is the z-direction, and the initial length of the simulation cell is L. At this moderate coverage, the brush is considerably entangled with itself at values of z/L of up to about .15. The brush extends considerably further into the melt, however, and brush-melt entanglements exist for z/L of up to about .35. At low values of z, the number of brush-brush and brush-melt entanglements are comparable, indicating thorough overlapping between melt and brush. However, the finite coverage density

is manifest in the low number of melt-melt entanglements for z/L below about .2. At moderate values of z/L, the number of brush-melt and melt-melt entanglements are similar. This behavior is indicative of the crossover between the low and high coverage regimes. At low coverages, the brush is negligibly entangled with itself, but more strongly entangled with the melt. At high coverages, towards the phase-separated regime, the brush is more entangled with itself and less entangled with the melt.

Another finding is that the number of entanglements of type X-Y (where X and Y = brush or melt) at given value of z is proportional to $\rho_X(z)\rho_Y(z)$, the product of the relevant densities, and that the proportionality constant is the same for all three types of entanglements. This suggests that brushes and melts entangle in essentially the same way. This is surprising given the different structure of brushes and melts. While the squared end-end distance $< R^2 >$ scales as N for the random-walk- like melt chains chains, it scales as N^2 for the directed brushes.

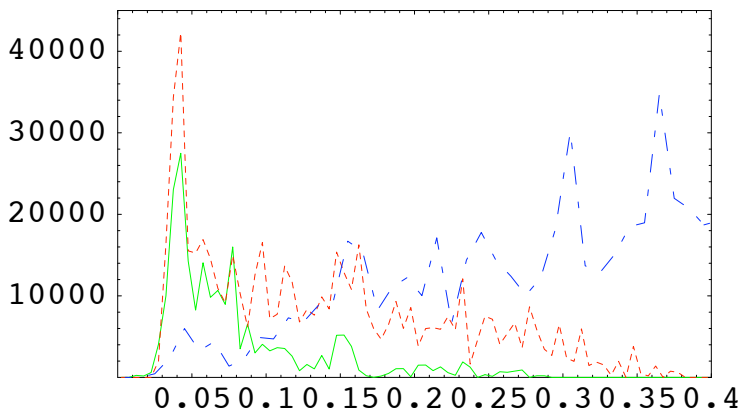


FIG. 2.5. *Entanglement of a melt-brush system for coverage* $.03/\sigma^2$. *The horizontal axis indicates the distance from the tethering surface and the vertical axis indicates the number of interchain contacts. The solid green line indicates brush-brush contacts, the dashed red line indicates brush-melt contacts, and the dot-dashed blue line indicates melt-melt contacts.*

While the data in Figure 2.5 is fairly clear and easy to interpret, it is less clear for some of the other systems we have examined. Fortunately we have a large number of initial states at the various coverages, and will continue running PPA analyses to get better statistics for the distributions of entanglement densities.

To gain a better understanding of the entanglement structure of brush-melt systems, we have also simulated "dry" brushes with no melt present. The goal is to see how much the brushes are entangled with each other. We have simulated dry brushes under two fundamentally different physical conditions. The first is "good-solvent" (GS), with purely repulsive interactions ($r_c = 2^{1/6}\sigma$) and $T = 1.0u_0/k_B$. The second is "θ-point" (TP), with relatively long-range attractive interactions ($r_c = 2.5\sigma$) and $T = 3.18u_0/k_B$. Although the interactions of real polymers are fixed, both GS and TP conditions occur in real systems.

MD Configurations from my simulations of dry GS and TP brushes are shown in Figure 2.6. The GS brush extends to a greater height and is more dispersed, while the TP brush is more condensed [2]. For this reason, as our PPA simulations have verified, the TP brushes are much more entangled. Thin-chain PPA simulations have shown that significant entanglement is not present in the GS brushes below coverage

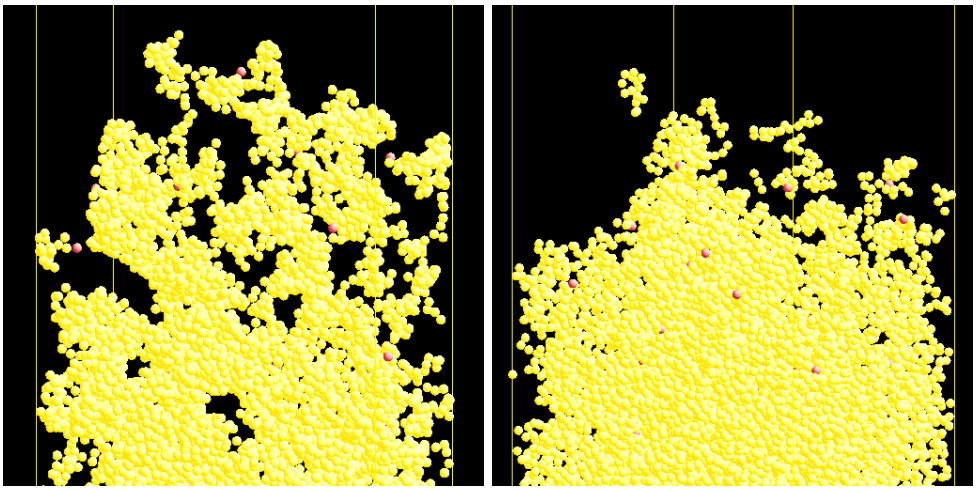


FIG. 2.6. *Images from MD simulations of brushes under good-solvent (left) and θ-solvent (right) conditions for $\rho_a = .05/\sigma^2$. Note that the brush in good solvent is more dispersed, and has the greater height. Bottom portions of brushes not shown.*

$.03/\sigma^2$, while for TP brushes it is present at coverages as low as $.01/\sigma^2$. This result is interesting in itself and is worth further investigation. We may want to simulate larger dry-brush systems (of order 1 million atoms), because statistics for the current systems are rather poor.

I am currently in the process of learning more about dry brushes, and we are hoping to determine a small set of specific physical questions to address. I will be continuing the study of entanglement in brushes after returning to Johns Hopkins. We hope to write a journal paper addressing issues in the physics of both wet and dry brushes sometime during the coming academic year. This paper will constitute part of my thesis work.

3. Acknowledgements. I would like to acknowledge the kind assistance of Sandia finite element experts Pavel Bochev, Michael Parks, and Rich Lehoucq as well as the assitance of the TAHOE team including Jon Zimmerman and Vicky Nguyen. I also would like to acknowledge the constant help of Steve Plimpton, who has proven to be an excellent mentor.

REFERENCES

[1] R. Everaers et. al., *Rheology and microscopic topology of entangled polymeric liquids*, Science, 303 (2004), pp. 823–826.

[2] G. S. Grest and M. Murat, *Stcuture of grafted polymeric brushes in solvents of varying quality: a molecular dynamics study*, Macromolecules, 26 (1993), pp. 3108–3117.

[3] K. Kremer and G. S. Grest, *Dynamics of entangled linear polymer melts - a molecular dynamics simulation*, J. Chem. Phys., 92 (1990), pp. 5057–5086.

[4] B. Luan, S. Hyun, J. F. Molinari, N. Bernstein, and M. O. Robbins, *Multiscale modeling of two dimensional contacts*, submitted, (2006).

[5] S. W. Sides, G. S. Grest, and M. J. Stevens, *Large-scale simulation of adhesion dynamics for end-grafted polymers*, Macromolecules, 35 (2002), pp. 566–573.

[6] S. W. Sides, G. S. Grest, M. J. Stevens, and S. J. Plimpton, *Effect of end-tethered polymers on surface adhesion of glassy polymers*, J. Polym. Sci. Part B - Polym. Phys., 42 (2004), pp. 199–208.

ADSORPTION OF NUCLEOTIDE MONOPHOSPHATES ON CARBON NANOTUBES

T. IACONIS* AND D. ROBINSON†

Abstract. Carbon nanotubes (CNT) are produced as a mixture with varying diameters, helicities, and electronic properties. Tubes purified with respect to these properties would be much more useful. DNA can disperse nanotubes into an aqueous medium, facilitating liquid-phase separation techniques for the purification process. It has been claimed that DNA-CNT interaction is dependent on the DNA's primary structure, suggesting specific interactions between individual nucleotides and the nanotubes. We are studying the strength of monomeric nucleotide-CNT interactions as a function of nucleotide identity, and weight ratio. After equilibrating carbon nanotubes with a given nucleotide in salt solution, we observe the depletion of nucleotides in solution due to adsorption onto the tubes by UV spectroscopy. We have found that nucleotides with larger, more electron-rich bases have a greater affinity for carbon nanotubes.

1. Introduction. Since their discovery in 1991, there has been great interest in using carbon nanotubes for a wide variety of applications. Commercially synthesized carbon nanotubes exhibit environment-sensitive electrical properties that range from metallic to semiconducting, depending on tube diameter and helicity. Their electrical properties are quite sensitive to adsorbed molecules, making them attractive as sensors and as components in molecular electronic devices. However, nanotubes are difficult to use due to their wide dispersion of geometries and conductivities, as well as a tendency to aggregate. Two critical steps to using these materials are to solubilize and disperse the tubes, and to sort them according to their properties. Recently, Zheng and coworkers [1] discovered that single-stranded DNA binds strongly to single-walled carbon nanotubes (SWNTs) and that the ssDNA effectively disperses bundled SWNTs into solution. They found that certain sequences of ssDNA are more effective than others, and that it is possible to use anion exchange chromatography to sort the SWNTs by tube diameter. The team proposed that the structure of a ssDNA / SWNT hybrid depends on the electronic character of the nanotube, which in turn depends on its diameter and whether the tube is metallic or semiconducting.

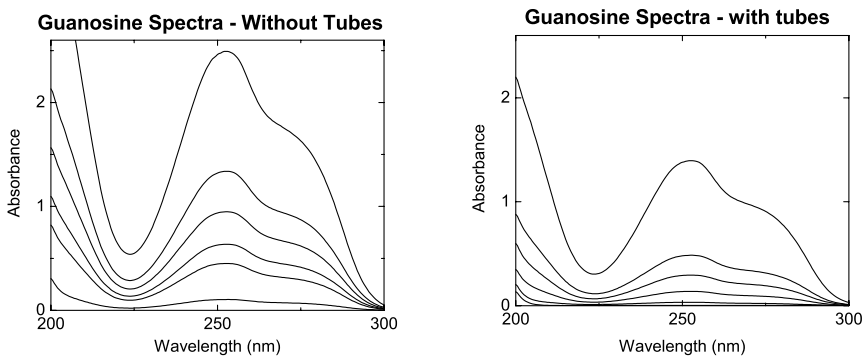
To better understand the interactions between ssDNA and SWNTs, we have studied the interactions of monomeric nucleotide monophosphates (NMPs) and SWNTs. Our goal was to determine the affinities of different NMPs for SWNTs, and explain what leads to any observed variation.

2. Methods. We began by weighing out carbon nanotubes into 4mL vials at 2-3 mg per sample, and preparing a 150 mM stock of salt solution. The salt solution consisted of 175 mg NaCl and 20mL DI water. Next we weighed out a chosen nucleotide monophosphate and added the salt solution to a 10 mg/mL nucleotide concentration. This was the stock nucleotide solution. We determined a set of weight fractions (mass nucleotides / mass CNT) to test; these values ranged approximately 0-2.5. We labeled a nanotube-containing vial for each fraction we intended to test. Dividing each weight fraction by the concentration of the nucleotide ("NMP") solution (10 mg/mL), we calculated the amount of NMP solution to add last into our samples. 0.5mL of salt solution was added to each tube vial per mg of tubes, but the volume of NMP solution to eventually add was subtracted from this amount. After the salt solution and nanotubes were combined, the calculated volumes of NMP solution were added. All

*Massachusetts Institute of Technology
†Sandia National Laboratories

vials were vortexed and the lids secured with parafilm. Samples were then sonicated in ice for three hours to equilibrate the components. The ice ensured that the samples did not heat up over the incubation period. We followed by centrifuging the vials for 5 minutes at 2000 rpm in order to settle the nanotubes with adsorbed nucleotides. To measure the amount of free nucleotides remaining in solution, 0.1mL was removed to a cuvette without redispersing the nanotubes from the sample. The remainder of the 1.5mL cuvette was filled with DI water and the UV spectrum taken from 200-400 nm using UV-Probe software. The process was repeated for all samples. A set of controls was run identically through the process with absence of CNT.

3. Results. By comparing the spectra of the controls with the spectra of the samples (Figures 3.1(a) and 3.1(b)), we see that there is a significant decrease in the absorbance levels when carbon nanotubes are present. This verifies that the nanotubes are in fact taking up a portion of the nucleotides, leaving less free in solution, and less to absorb the UV light. Also, the spectrum lines increase with increasing weight fraction for both the tubes and controls. This simply indicates that in both cases, when more nucleotides are added to the sample, more are present in solution.

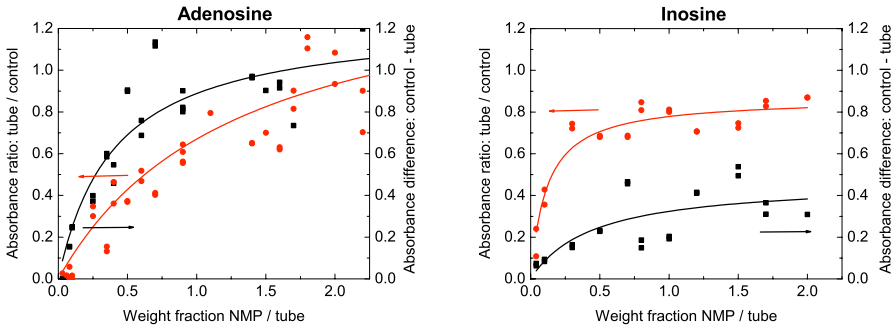


(a) (b)

FIG. 3.1. *Guanosine Spectra*

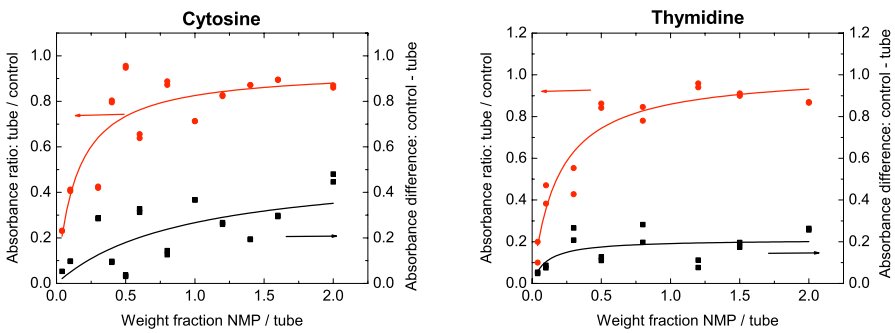
More importantly, we compared the adsorbance patterns between nucleotides. We measured this in two ways (Figures 3.2(a)-3.4(b)). Both methods compare the UV absorbance of a sample (with nanotubes) to the UV absorbance of a control (no nanotubes) for a given wavelength and weight fraction. The first measure compares these quantities as a ratio of the sample to the control. This ratio increases as we increase the weight fraction, but eventually levels off close to 1. This happens as the nanotube surfaces become full and saturated with the nucleotides. The second measure is the difference between the aforementioned quantities. This is a direct measure of the amount of nucleotides that stuck to the tubes. This quantity also increases with weight fraction because more nucleotides can adsorb when more are present. However as before, the surfaces of the nanotubes reach saturation and the curve levels off. The data is fit with curves according to the Langmuir adsorption model, as discussed below.

Comparing the difference curves and ratio curves, it is clear that guanosine and adenosine differ greatly from thymidine, uridine, cytosine, and inosine. The ratio curves are low, and the difference curves high for these two nucleotides. The latter



(a) (b)

FIG. 3.2. *Adsorbance patterns between nucleotides.*



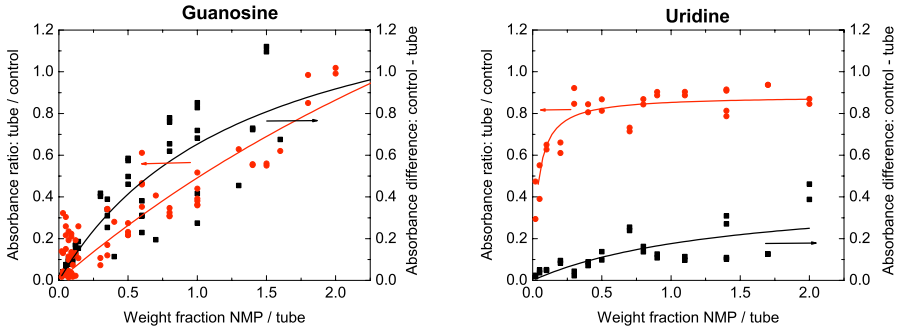
(a) (b)

FIG. 3.3. *Adsorbance patterns between nucleotides.*

four exhibit the opposite, with considerable difference between the curves. Adenosine and guanosine must therefore adsorb much better to the tubes

4. Discussion and Conclusions. We found that the tubes take up NMP increasingly with concentration, and begin to saturate at a level unique to the NMP. As shown above, guanosine and adenosine take higher concentrations to saturate. These nucleotides are adsorbing to the tubes much more strongly than uridine, thymidine, cytosine or inosine. Comparing the molecular structures of the NMPs, guanosine and adenosine are more aromatic and contain more electron donating groups than the others. Specifically they contain donating amines, and fewer withdrawing oxygens. These properties therefore cause stronger adsorption onto the carbon nanotubes.

The curves are fit to our data from the Langmuir adsorption model. However, the difference in degree is different between the nucleotides. The discrepancies with our data lead us to believe that our system has significant differences from Langmuir. The basis of the Langmuir model is coverage of a surface by a monomolecular layer, including parameters for saturation and affinity. In our situation the NMP molecules are covering the carbon nanotube surface, but the Langmuir model is not completely applicable here because it assumes a constant surface area for adsorption. It is pos-



(a) (b)

FIG. 3.4. *Adsorbance patterns between nucleotides.*

sible that with the adsorption of the NMP to the tubes, the bundles of tubes are more dispersed and expose more surface area. Nucleotides that adsorb better may also disperse better. This may account for the differing patterns we have observed. Since adenosine and guanosine continue to exhibit adsorption into higher weight fractions, they must be dispersing the nanotubes well in order to open up surface area to accommodate the molecules.

Further work could include studying the effect of temperature on the adsorption process. We will also look at oligomers and other artificial polymers that may optimize the adsorption and dispersion mechanism. Ultimately, we hope to fully understand the way that DNA-like molecules interact with carbon nanotubes, and be able to use this process to effectively separate and sort nanotubes with high yield and purity.

REFERENCES

[1] ZHENG *et al*, Nature Materials, 2 (2003), p. 338.

FUNCTIONAL METAL-ORGANIC FRAMEWORKS: SYNTHESIS OF NEW FLUORESCENT MOFS AND GROWTH OF MOF5 THIN FILMS ON SAMS

V. LIU*, C. BAUER†, M. ALLENDORF†, AND R. SIMMONS†

Abstract. Metal-Organic Frameworks (MOFs) have recently attracted interest due to their regular porosity, high surface area, and potential for application in gas and vapor storage, separation, and sensing. In our experiments, we utilized the rigidity and well-defined geometries inherent in MOFs to provide a low-density matrix for the incorporation of fluorescent linkers with potential for vapor sensing. In this paper, synthesis strategies for various MOFs, their characterization, and their fluorescence properties are discussed. Ultimately, we intend to use these materials to perform vapor sensing and to incorporate them into thin film devices.

1. Introduction. MOFs have attracted attention as a new class of nanoporous materials, with applications ranging from gas storage to sensing and separation. One of the advantages of MOFs is that the products of their synthesis can be reasonably predicted from the starting materials [1]. This synthetic methodology has allowed the use of a variety of organic molecules as linkers in the MOF crystals, and we will discuss the growth strategies of these MOFs in the following paragraphs. We would also discuss our application of these MOFs as fluorescent chemical sensors and the growth of these MOFs on surfaces functionalized with self-assembled monolayers (SAMs).

2. Synthesis Procedures. In the following paragraphs, the methods used to synthesize each type of MOF are detailed and the observations specific to each type of MOF are noted.

2.1. MOF 5/IRMOF-1.

Procedure: Weigh out appropriate amounts of materials (consult table 2) and put them into a scintillation vial. Measure out appropriate amount of DEF with a graduated cylinder and mix DEF with the materials. The solution at this point should appear slightly cloudy due to the undissolved linker material. Place a stir bar into the solution and let it sit on a stir plate for approximately 1 hr, or until everything is dissolved. The solution should then be placed into the oven at 105 ^{o}C for about 1 day.

Observations: After one day, crystal growth should be observed on side walls and the bottom of the vial. The color of the DEF should also turn to a dark yellow/light brown. The decanted DEF solution can be put back into the oven at 105 ^{o}C for continued crystal growth. It has been observed that these crystals are generally smaller than the first growth.

Fluorescence: Weakly fluorescent (color = blue-green)

Additional info: Addition of water to the reagents can result in a different material (verified by powder XRD) being produced, depending on the amount of water added. We have found that being at about 4% (by volume) water in solution, the crystals cease to form. This is similar to the calculated water sensitivity as published by J. Greathouse and M. Allendorf [1].

*University of California at Berkeley
†Sandia National Laboratories

Table 2.1

Summary of reagents used for synthesis of various MOFs. Note: Metal source can be also be Zn(NO$_3$)$_2$ cdot 4H$_2$O w/ a conversion between different molar masses.

Type	Metal Source	Linker	Solvent
MOF 5/IRMOF 1	$Zn(NO_3)_2 \cdot 6H_2O$	Terephthalic acid	DEF
IRMOF 8	$Zn(NO_3)_2 \cdot 6H_2O$	2,6-Naphthalenedicarboxylic acid	DEF
IRMOF 10	$Zn(NO_3)_2 \cdot 6H_2O$	4,4-Biphenyldicarboxylic acid	DEF
2D Stilbene MOF	$Zn(NO_3)_2 \cdot 6H_2O$	Stilbenedicarboxylic acid	DMF
3D Stilbene MOF	$Zn(NO_3)_2 \cdot 6H_2O$	Stilbenedicarboxylic acid	DEF

DEF = Diethylformamide, DMF = Dimethylformamide

Table 2.2

Summary of weights/volumes of reagents for various MOFs Note: The weights for all the above entries are based on reactions with 10mL of solvent. To synthesize material on a larger scale (i.e. with more solvent), the weight of materials required is obtained by multiplying the amounts by x/10, where x is the volume of solvent to be used.

Type	$Zn(NO_3)_2 \cdot 6H_2O$ mass (mg)	$Zn(NO_3)_2 \cdot 4H_2O$ mass (mg)	Linker mass (mg)	Molar ratio
MOF 5/ IRMOF 1	357	313.75	67	3
IRMOF 8	626	550	60	7.57
IRMOF 10	17.53	15.4	3.2	4.54
2D Stilbene MOF	119	104.6	35.67	3
3D Stilbene MOF	119	104.6	15.33	6

2.2. IRMOF 8.

Procedure: Similar to MOF 5

Observations: : Similar to MOF 5, with much smaller crystal sizes.

Fluorescence: Weakly fluorescent (color = purple)

Additional info: These crystals can also be formed at room temperature by leaving the solution out for about 2 days. Crystal sizes appear smaller but the quality of the crystals might be better.

2.3. IRMOF 10.

Procedure: Similar to MOF 5

Observations: These crystals are very small; under the microscope, the crystals look slightly spherical.

Fluorescence: Not very fluorescent (color = blue)

Additional info: The single crystal structure of this material was never obtained by the Yaghi group because of their small sizes.

2.4. 2D Stilbene MOF.

Procedure: Stilbenedicarboxylic acid is fairly difficult to dissolve in DMF. If you just add all the stilbene into DMF at once, it is likely that a large portion of the material would not be dissolved. To facilitate the synthesis, use the following method:

1. Weight out required amount of stilbenedicarboxylic acid in a scintillation vial and set it aside. Measure out the required amount of DMF, add 5-10mL of DMF into the scintillation vial, and put the rest of the DMF into a glass bottle or flask.

2. Begin by sonicating the vial for ∼1-2 minutes to break up large stilbene pieces. Afterwards, use a glass pipet to siphon a couple drops of the solution from the scintillation vial and add it to the bottle/flask. Swirl the flask until the solution appears clear. Heating the solution with a heat gun while swirling can help the stilbene pieces dissolve into the solution. Repeat this process until all the solution is in the flask. (At the end, you might want to rinse the vial with the solution for a couple times to ensure all the stilbene pieces are collected into the flask) Note that its OK for the solution to be a little cloudy at the end, as the increase in ionic strength caused by the addition of zinc nitrate will help in the dissolving process. Note: if the solution still appear cloudy after swirling/heating, or if you can see large pieces of stilbene in the solution, try sonicating it for ∼1-2 minutes.
3. Weigh out required amount of zinc nitrate into a scintillation vial, and proceed by adding the stilbene solution into the vial (if vial is too small for all the solution, you can just rinse it with the solution and pour/pipet it back to the larger container; if you do this, make sure you have collected all the zinc nitrate pieces by sonicating and rinsing the vial a couple of times)
4. At this point, the addition of Zn can help dissolve the remaining pieces of stilbene that are not yet dissolved. If solution remained cloudy, put the vial into an oven at 105 ^{o}C for around 15 minutes or until the solution looks clear.
5. Filter the solution with gravitational filtration and put the solution into an oven at 105°C overnight.

Observations: These crystals look like daggers under an optical microscope. Crystals are air-stable (due to strong coordination with DMF molecules), making them good candidates for vapor sensing and other applications. The crystals are called "2D" because of the crystals structure: the molecules form extended sheets of crystals separated by DMF molecules.

Fluorescence: These crystals are very bright, and the color looks blue. The intensity and relative peak heights depend on the size/shape of the crystal. Therefore, when the fluorescence is measured with a fluorometer, the absolute intensity in counts per-second (cps) can vary quite a bit, but the normalized shapes will be very similar.

Additional info: We have found that stilbene is easier to dissolve in DMF than DEF, if the solution looks clear before filtering, that step can be omitted.

2.5. 3D Stilbene MOF.

Procedure: Similar to 2D stilbene MOF, but use DEF instead of DMF as a solvent. However, this requires more significant sonication and may require filtration before the crystals are grown to remove significant amounts of undissolved linker.

Observations: These crystals are somewhat sensitive to air (moisture) and require inert or dry atmospheres when preparing, similar to the other IRMOFs. These were shown (by single crystal XRD) to have an IRMOF-like structure (Posessing the ZnO_4 cluster linked by stilbene) and have an interpenetrated structure.

Additional info: The interpenetrated structure reduces the measured pore volume, therefore the size of the pores are not as big as expected. The measured Langmuir surface area is approximately $700 m^2 g^{-1}$.Also, decreasing the concentration of reagents produces smaller but higher quality crystals.

2.6. Anthracene MOF. Note: As of the time of the writing of this document, we have yet to successfully synthesize high quality crystals with its linker.

Procedure: It is very difficult to dissolve anthracene in DEF or DMF; we have tried using different cosolvents of DMSO, DMF and DEF to improve the solubility of anthracene, and we have found that a ratio of 4:1 DMSO:DEF/DMF can dissolve anthracene well. Even with this mixture of solvents, we still had to use a procedure similar to the synthesis of stilbene MOFs to dissolve the anthracene.

Observations: Decomposed anthracene would form in the vial after the vial is placed in the oven at 105°C overnight. The decants of the solution are placed in the oven at 85°C, but so far we have only polycrystalline material.

2.7. MOF Storage. When the vial with the MOF with DEF is taken out of the oven, pour out the DEF into a container as soon as possible and fill the vial with DMF. The DMF would help wash out the decomposed DEF in the pores of the crystals and any unreacted salts. After leaving the crystals in DMF for >0.5 hr, pour out the DMF and pour chloroform into the vial. Repeat the chloroform rinse for 2 more times (separated by about 1 day) and the MOFs can be stored in chloroform indefinitely.

3. MOFs on SAMs. In order to incorporate MOFs into sensing devices, we need to be able to reliably grow MOFs on surfaces. Carboxylate-terminated SAMs provide a promising way to attach MOFs on surfaces by encouraging nucleation on the SAMs. The growth of SAMs on a gold surface (on Si wafer) can be incorporated in semiconductor processing, and this can help incorporate these MOFs into sensing devices.

3.1. MOF5 on SAM.

Procedure:

1. Mix 90mL of ethanol and 10mL of acetic acid, and also prepare a 100mL portion of MOF5 crystals.
2. Weight out 6mg of mercapto hexadecanol and put this into 10mL of the 10% ethanol-acetic acid mixture. This solution is then diluted by 100 fold with the ethanol-acetic acid mixture. The method we used was to add 1 mL of the mercapto solution to 99 mL of the original mixture to produce a 100mL solution.
3. Put a cleaned gold substrate into the solution overnight at room temperature.

Next day:

1. Rinse gold substrate thoroughly with ethanol
2. Decant MOF5 solution
3. Place gold substrate into MOF5 decant
4. Leave the solution with gold substrate at room temperature overnight

4. Sensor responses. 3D stilbene MOFs and a solvent are placed in a quartz cuvette. The quartz is then put into a black box and the excitation/emission is collected through a fiber-optic. Since the 3D stilbene MOFs are usually stored in chloroform, a solvent-exchange would be required if we want to test the crystals in different solvent. For solvent-exchange, a 3-time rinse with the new solvent ensures that the old solvent is mostly displaced before the MOFs are exposed to the new solvent.

Preliminary testing with 3D stilbene MOFs has shown a shift in the fluorescence spectrum when the crystals are immersed in different solvents. In the graph below, 4 solvents of varying polarity are compared. A comparison of the peak positions shows that different solvents have produced different shifts (compared to chloroform) of the

spectrum. We think this is due to the different electrostatic interaction between the solvent and the stilbene linkers in the crystals. Hexane, being a nonpolar solvent, can be reasonably expected to have a different interaction with stilbene than ethanol, which is a more polar solvent. The responses to the solvents are reversible except for the case of pyridine; we suspect this is due to the pyridine coordinating to the metal clusters of the MOF, fundamentally changing the crystal structure and leading to an irreversible change in the fluorescence spectrum. We observe a decomposition of the crystals upon exposure to pyridine.

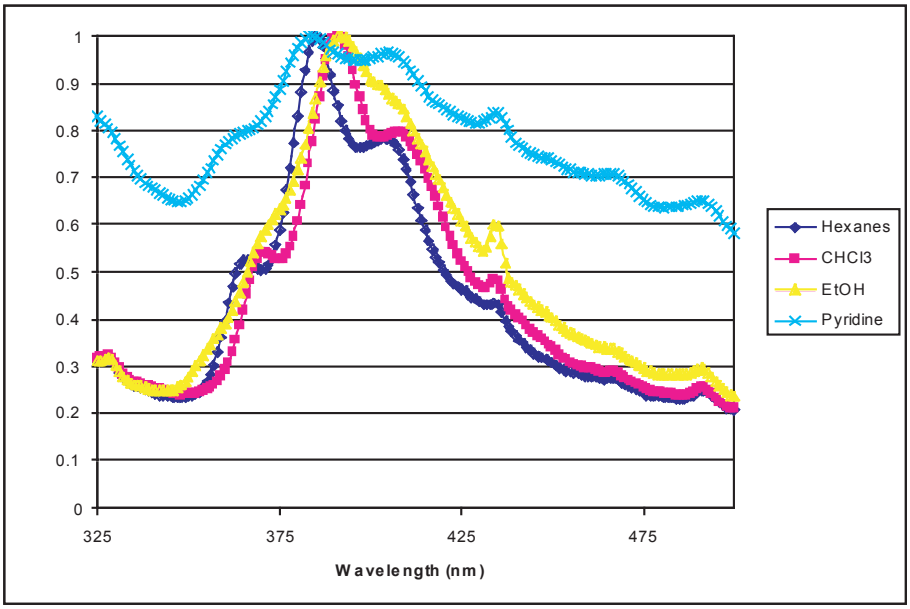


FIG. 4.1. *Fluorescence spectra of 3D stilbene MOFs in various solvents.*

5. Conclusion. In this work, we have detailed the synthetic strategies of various MOFs, characterized these new MOFs with fluorescence spectroscopy, and have achieved sensor responses of 3D stilbene MOFs in different solvents. In the future, we hope to demonstrate vapor sensing and incorporation of fluorescent MOFs onto SAMs.

REFERENCES

[1] J.A. GREATHOUSE AND M.D. ALLENDORF, J. Amer. Chem. Soc., (2006). ASAP Article Web Release Date: July 29, 2006.

CLEANING TECHNIQUES AND MECHANICAL PROPERTIES OF DIATOM FRUSTULES

T. LYNCH* AND B. SIMMONS†

Abstract. Since the production of nanoscale parts has proven to be fairly difficult by ordinary means, biomimetics is taking another approach in attempting to mimic biological systems production of intricate, microscopic structures. Further understanding the mechanics of assembling the silica nanosphere structure that composes the diatom frustule and its resulting strength will aide in the investigation of synthetic self-assembling structures. This was investigated through fluorescence microscope imaging of PDMPO doped diatom frustules, thorough cleaning the frustules using three different acids as the cleaning agent, and nanoindentation performed by a Dimension 3000 AFM. The resulting elastic moduli from nanoindentation of 7 to 640 MPa were lower than previous experiments have shown. This is believed to be caused by an inadequate indenting AFM. However, these results put us on the right track to understanding more about this biological system.

1. Introduction. The diatom outer shell, or frustule, is composed primarily of silica nanospheres and a complex matrix of proteins (silaffins), which initiate and control the production of silica within the cell and construct the frustules during cellular division. Because of the frustules' porous framework and multi-scale silica composition, they possess a strong and lightweight structure and therefore the diatom frustule has been found as a favorable system to utilize as a model system in the realization of similar silicified materials in a laboratory setting. Self-assembling silica nanospheres like these would be immensely useful, especially in further developments/innovation of thin-films and microscopic frameworks. Before a synthetic system can be produced and understood, the model system of the frustule must be thoroughly studied to learn more about the mechanisms used to produce and assemble the silica frame and its resulting strength. To further understand the capabilities of the frustule, a thorough processing technique must first be administered to remove the extra-polymeric substance, which coats certain species, and other organic materials, to leave the silica structure in a pristine condition. Once this is accomplished, mechanical property tests, such as indentation and lithography, can be conducted to gain knowledge of the hardness and elastic modulus of the frustule without having tainted results from organic matter coating the frustule surface. This experiment will explore the production and allocation of silica during the cell division process by fluorescence microscopy, continued by comparing several different acid-based cleaning techniques and their efficacy in cleaning diatoms, followed up with a comparison of the mechanical properties of the frustules as a function of species as measured with nanoindentation.

2. Experimental Materials and Methods. Diatom species *N. alba*, *T. pseudonana*, and *T. weissflogii* were cultured at Sandia National Laboratories, CA, through the expertise of Pamela Lane. The PDMPO/silica uptake experiment was conducted on *N. alba*. PDMPO, (2(4pyridyl)5((4(2dimethylaminoethylaminocarbamoyl) methoxy)phenyl) oxazole), is a fluorescent dye that traces silica movement. Diatoms are synchronized by keeping them in a saltwater medium void of silica to not allow further cellular divisions. At time equal to zero, they are exposed to a silica and PDMPO rich environment. Every increment of forty minutes, a sample was removed, exposed to ionophores, and resuspended in a silica dry medium. To image, they were placed on

*University of Colorado at Boulder
†Sandia National Laboratories

a glass slide and viewed with a mercury lamp under a green light filter at 510 nm. The fluoresced light at about 530 nm was recorded using the optics portion of the microscope.

These three species then underwent an acid-based cleaning process to remove the organic material coating the frustule exterior. Nitric acid, sulfuric acid and piranha were used as the cleaning agents. 200 μl of diatoms suspended in media, with a concentration of ~500 million cells per ml, was added to 800 μl of as received acid. Frustules were subjected to each acid for a marked period of time to determine the appropriate cleaning time per acid. Nitric acid evaluation took place at five, ten and fifteen minutes of exposure. Sulfuric acid cleaning was evaluated at two, four and six minutes of exposure. Piranha, however, was only performed once on *T. pseudonana* and *T. weissflogii* for three minutes.

Once a proper cleaning time was determined, the washed frustules were ready for AFM imaging and indentation. The frustules were mounted on glass slides prepared with poly-l-lysine. The glass slides were first washed with soap and water, and then dried. A drop of poly-l-lysine was applied to the slide, allowed to sit for one minute, and then rinsed with water. The cleaned frustules were then pipetted onto the glass surface and heated to evaporate the media and to lock the frustules in place. AFM images were acquired with a Nanoscope IIIa controlled Dimension 3000. Fat/short DNP silicon-nitride contact imaging tips with a spring constant of 0.06 N/m were used to image and indent. Once a good image was obtained, the microscope was switched to force imaging mode where the tip would be pushed into the surface of the diatom, giving back a Tip Deflection vs. Z Height curve. The raw data from this curve was extracted and converted to usable units. Following the indentation analysis procedure produced by Oliver and Pharr [1], the elastic modulus and hardness of the diatom frustule were calculated.

3. Confocal Microscopy Results. Viewing the displacement of silica during the cell division process can lead to greater progress in silica nanosphere production. In this case, the silica uptake experiment marks only the new silica that has entered into the cell. A normal cell division process for this species takes around six to eight hours and can clearly be seen in Figure 3.1. The silica is taken into the cell and

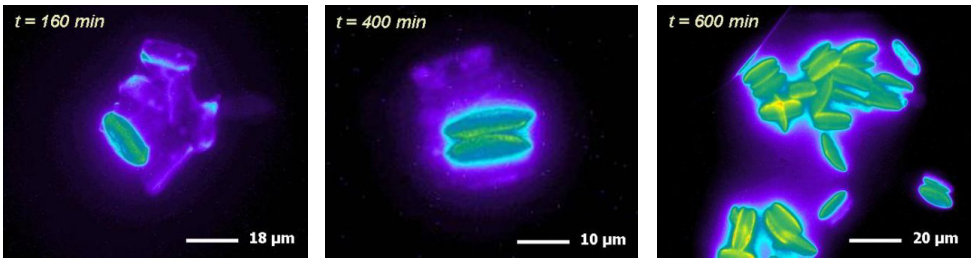


FIG. 3.1. *These images show the progression of cellular division. The left image displays the beginning of the process when the cell is pooling silica to utilize later in the production of new valves. The center image clearly shows the division process in action as the cells are moving apart with the new valves glowing green in the center of the separation. The right image is an image of post cellular division where the concentration of silica is definitely strongest in the central nodule region.*

pooled to provide an ample supply during cellular division. When the cell is ready to divide and the new valves have been formed in the silica deposition vesicle, the cell separates, places the new valves, and is finished. The new valves contain the

highest concentration of silica on the central nodule region, as has been shown to be the strongest portion of the diatom frustule from previous studies.

4. Acid-Based Cleaning Results. This process proved to be very species dependent, as no two species reacted equally to the same acid-based cleaning procedure. These results were verified through images obtained using a field emission scanning electron microscope. *T. pseudonana* seemed to fair quite well with all three cleaning procedures with piranha producing the best results by removing organic matter while still leaving a completely intact frustule as shown in Figure 4.1

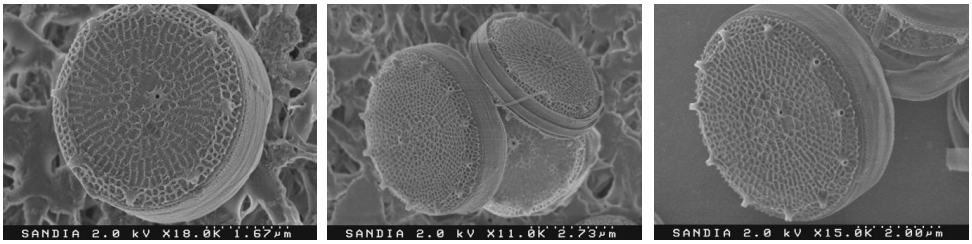


FIG. 4.1. *The cleaning process of nitric acid at 10 minutes (left), sulfuric acid at 6 minutes (center), and piranha at 3 minutes (right) performed on T. pseudonana. From these images, sulfuric acid and piranha seemed to clean most thoroughly while still leaving an intact frustule. Nitric acid did not clean as thoroughly and would begin degrading the frustule at longer times before removing all organic matter.*

T. weissflogii reacted quite differently to the acid treatments, as shown in Figure 4.2. The best cleaning procedure for *T. weissflogii* was nitric acid at 10 minutes.

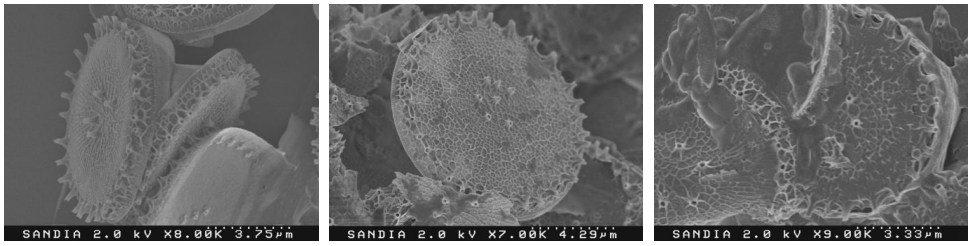


FIG. 4.2. *Scanning electron micrographs of nitric acid (left), sulfuric acid (center), and piranha (right) cleaning, respectively, performed on T. weissflogii. In this case, nitric acid produced the best result with cleaned frustules without degradation of the valves primarily. Sulfuric acid and piranha began degrading the frustule before organic material was removed.*

Sulfuric acid and piranha would destroy the girdle bands very quickly and begin degrading the frustule intensely before having removed all of the organic material. Piranha did the same, but much quicker and removing even less organic matter.

For the pennate species that was tested in addition to the previous two centric species, we gained more results showing how this cleaning technique is very species dependent. As seen in Figure 4.3, *N. alba* reacted to nitric acid in a similar manner as that of *T. pseudonana*. The best cleaning came from a slightly longer exposure to sulfuric acid at 6 minutes. Nitric acid would begin removing the girdle bands and degrade the frustule before all organic material was removed.

The questions that arise from seeing the results of the acidic cleaning processes are: does a certain acidic cleaning process determine the resulting mechanical strength

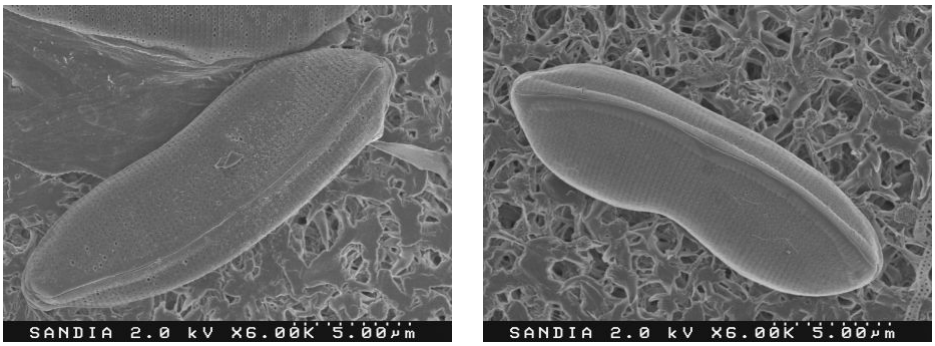


FIG. 4.3. *Scanning electron micrographs of cleaning N. alba with nitric acid at 10 minutes (left) and sulfuric acid at 6 minutes (right).*

of the frustule? and does the acidic cleaning procedure the frustule can withstand reflect its overall strength?

5. AFM Mechanical Properties Tests. With the composition of the frustules and their height protrusion of around 3 μm off of the glass slide, the best mode of scanning turned out to be contact mode with DNP silicon-nitride tips. These tips are some of the softest made, with an elastic modulus of 310 GPa, and could scan the large height changes in the diatom frustule fairly well.

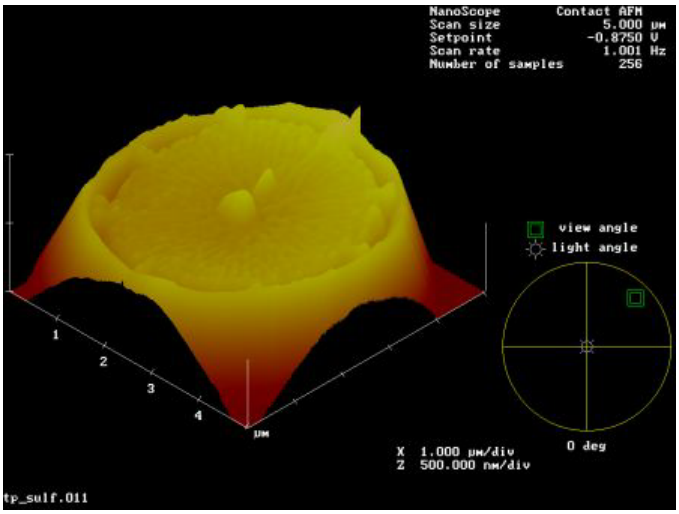


FIG. 5.1. *3D image of T. pseudonana cleaned by sulfuric acid obtained by AFM contact mode scanning.*

When a good image of the surface was produced, the microscope was transferred to force mode. The Z scan size and start height were then adjusted to optimize the contact time on the surface. From the resulting Tip Deflection vs. Z Height data, a Force vs. Indent Depth curve was produced. Using the method derived from Oliver and Pharr [1], an elastic modulus and hardness of the sample were calculated. The load, P, can be calculated by taking the spring constant of the cantilever and multiplying it by the tip deflection. This is treating the cantilever like a simple spring.

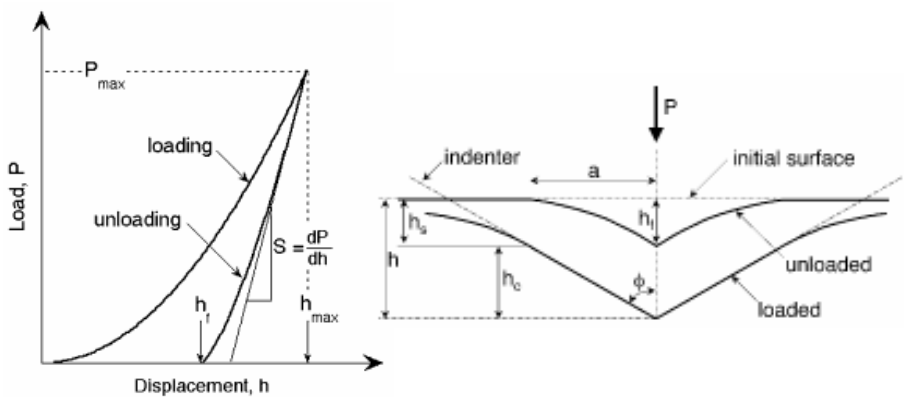


FIG. 5.2. *The loading and unloading of a sample during nanoindentation have been visualized in these images. The left image displays a proper unloading curve from which the slope and loading forces can be extracted from. The right image shows the different forms of the indent when loaded and unloaded, giving the plastically deformed indent depth.*

The slope of the unloading curve, S, can be obtained through interpolation. Starting with the plastic deformation left behind when the tip has been removed, the height of the indent depth can be defined by:

$$h_s = \epsilon \frac{P_{max}}{S} \tag{5.1}$$

where ϵ is a correction factor for the tip geometry. Using this dimension, the height at which the tip is in contact with the sample during loading is

$$h_c = h_{max} - h_s \tag{5.2}$$

where h_{max} is the height from the planar surface to the bottom of the indent when fully loaded. The cross sectional area of the indenting tip in contact with the sample is a function of that height $A = F(h_c)$. This area is then used to compute the hardness and elastic modulus of the sample

$$H = \frac{P_{max}}{A} \tag{5.3}$$

$$S = \beta \frac{2}{\sqrt{\pi}} E_{eff} \sqrt{A} \tag{5.4}$$

$$\frac{1}{E_{eff}} = \frac{1-\nu^2}{E} + \frac{1-\nu_i^2}{E_i} \tag{5.5}$$

β is a correction factor for a square based punch.

Based on extracting the slope, forces, and maximum penetrating force from the curve below and applying the above equations, the overall modulus of the diatom frustules tested in this experiment ranged from 7 to 640 MPa.

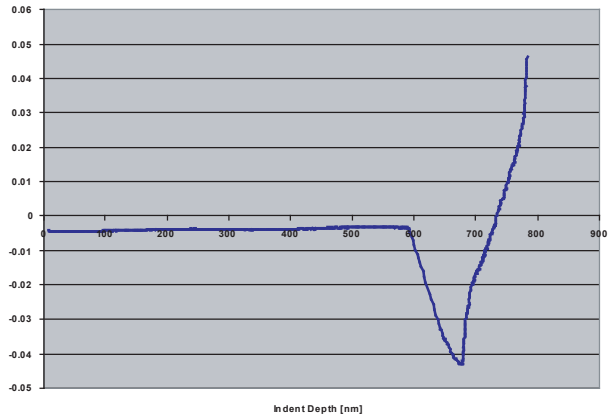


FIG. 5.3. *Tip Deflection vs. Z Height curve obtained while indenting T. weissflogii cleaned with sulfuric acid.*

6. Discussion. Following the previous works of Subhash, Yao, Bellinger and Gretz [2], they show that these results may be on the right track but in no way are to be taken as actual values for the elastic modulus of a diatom frustule. From their experiments, the frustule should have a modulus ranging from 0.6 to 2.8 GPa. Some of the results from this experiment fall into that range, but for the most part do not display a true representation of the strength of the frustules. In addition to these observations, the majority of the indentations performed did not produce a Force vs. Indent Depth curve with a positive slope returning from the maximum force applied. The bulk of the indents produced a curve more like that shown in Figure 6.1 The most

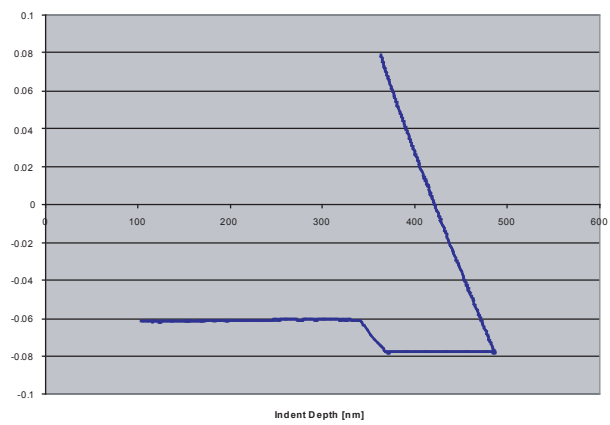


FIG. 6.1. *This is how the majority of the indent curves returned. The unloading portion is backwards implying that with a smaller force one could gain a greater indent depth.*

probable explanation for producing these curves is the machine used to perform the indents was not strong enough or built to indent with the scanning probes provided. Also, the maximum indenting force that the curves display is only around 0.08 μN, where as previous studies have shown proper indents on frustules with a force in the

range of 5 to 10 μN. With as small of a force as is produced by this microscope, it is projected to only have a plastically deformed indent of a few nm in depth. This would not show up on rescanning the frustule immediately after indenting. No indent was ever viewed. Therefore, the few indents that produced positive results may be seen as a fluke from the normal distribution of results with nonsensical interpretations. Further revisions of the process used in this test are underway and should produce real values for the elastic modulus of diatom frustules.

7. Conclusion. The use of diatom frustules in biomimetics has a strong application in the production and manipulation of silica nanospheres and microscopic frameworks. Hopefully a further understanding of the biological system will give greater knowledge in producing a more robust and plausible synthetic system. From this experiment the basics of silica deposition throughout the cell during cellular division were examined through fluorescence microscopy to reveal the strongest regions of the frustule as well as give further insight into the mechanics of the silaffins when producing silica nanospheres. In addition, the organic coating of the frustules was removed using three acid cleaning procedures, and each process was then subjected to nanoindentation to observe the effect of the cleaning procedure on the strength of the diatom frustule. Unfortunately, the indents performed in this test were not satisfactory for the purposes of determining the strength differences between the three species of diatoms, between the centric and pennate forms, and between the three different acid cleanings applied. Further tests will be conducted on these frustules using a stronger machine with a more precise indenting tip to gain actual results as to the strength of the diatom frustules.

REFERENCES

[1] W.C. Oliver and G.M. Pharr, *Measurement of hardness and elastic modulus by instrumented indentation: Advances in understanding and refinements to methodology*, J. Materials Research, (2003).

[2] G. Subhash, S. Yao, B. Bellinger, and M.R. Gretz, *Investigation of mechanical properties of diatom frstules using nanoindentation*, J. Nanoscience and Nanotechnology, 5 (2005), pp. 50–56.

A GENERALIZED CONTINUUM MODEL FOR FINITE DEFORMATION CRYSTAL PLASTICITY ACCOUNTING FOR THE INFLUENCE OF DISCLINATIONS

J. MAYEUR* AND D. BAMMANN†

1. Introduction. The goal of this research is to develop a generalized continuum model of crystal plasticity capable of predicting defect substructure development over a wide range of length scales spanning from nanometers to microns. Defect substructure evolution, as seen in highly deformed crystalline materials, is a self-organized highly cooperative process driven by the minimization of free energy with respect to certain kinematic constraints such as boundary conditions, grain boundaries, free surfaces, etc. The formation and arrangement of these structures is the result of equilibrated internal forces and moments with respect to these constraints. Properly accounting for the balance of these internal stresses (in addition to the classical continuum mechanics balance laws) is essential to predicting realistic substructure development and its influence on material response ranging from texture evolution to fragmentation and fracture processes. The methodology proposed to accomplish these behaviors within a continuum level framework is the inclusion of disclinations into the kinematics of geometrically-oriented crystal plasticity. The incorporation of disclinations provides additional rotational degrees of freedom, as well as an additional characteristic length scale with which to compete with that of dislocations in driving the evolution of the material substructure.

2. Background. Disclinations are linear, rotational defects initially conceptualized by Volterra simultaneously with the more familiar translational dislocations from elementary plasticity theory. A disclination can be visualized by considering a hollow cylinder with a cut running the length of the cylinder. A disclination is then produced by rotating the cut faces of the cylinder with respect to each other about one of the three coordinate directions defined by the cylinder axis (disclination line direction), normal to the cut plane, and the cross product of the cut-plane normal and axis direction. Each of these 3 rotations produces a different type of disclination as illustrated in Figure 2.1. Rotating about an axis parallel to the cylinder axis

FIG. 2.1. *Types of disclinations.*

produces a wedge disclination (right), whereas rotating about the axes perpendicular to the cylinder axis produces twist disclinations. Therefore, there are two distinct

*Georgia Institute of Technology
†Sandia National Laboratories

types of twist disclinations: a twist disclination with axis normal to the plane of the cut does not result in inter-penetration of material (middle), while the twist disclination with rotation axis in the plane of the cut and perpendicular to the cylinder axis does (left). While this simple geometric construction of the disclination concept is beneficial, it does not give insight as to how these defects might appear in and influence the properties of crystalline materials. Just as (translational) dislocations are required to preserve crystalline symmetries (i.e., conservation of burgers vector), disclinations must, in turn, obey the rotational symmetries of the crystal structure. Unlike liquid crystals and polymers (which do not posses long-range order), disclinations must appear in screened configurations in crystalline materials due to the large amount of strain energy associated with the presence of a full disclination. Therefore disclinations must have their stress field screened either by a surrounding dislocation distribution or by appearing as partial disclination dipoles. The manifestation of partial disclination dipoles in crystalline materials is in the form of high angle (>15 deg) grain/subgrain boundaries. At misorientation angles greater than this, the classical dislocation description of the boundary no longer applies as the individual dislocation cores become so closely spaced that they cannot be uniquely identified. The crystal boundary then takes the form of a distribution of partial disclinations (at least for tilt grain boundaries, i.e., wedge disclinations).

3. Model. In addition to adding disclinations into the kinematics of crystal plasticity, we also introduce a generalization of the classical multiplicative decomposition of finite deformation plasticity, i.e. $\mathbf{F} = \mathbf{F}^e\mathbf{F}^p$. For simplicity, the kinematic enhancements are first discussed within the framework of crystal plasticity considering dislocations only. Our argument is that an additional term, $\mathbf{F}^i$, should appear in the decomposition which represents a local compatibility restoring deformation, i.e., $\mathbf{F} = \mathbf{F}^e\mathbf{F}^i\mathbf{F}^p$, which reflects the state of internal elastic strain due to the presence of geometrically necessary dislocations. This reflects the belief that the unloading of the body through $\mathbf{F}^{e^{-1}}$ should not produce an incompatible anholonomic configuration, but rather the unloading of internal (residual) stresses in the absence of applied tractions through $\mathbf{F}^{i^{-1}}$ should produce the familiar intermediate configuration of crystal plasticity. In this point-of-view $\mathbf{F}^i\mathbf{F}^p$ is the solution to the micromechanical problem where $\mathbf{F}^p$ is the source of eigenstrains, $\mathbf{F}^i$ is due to microelastic strains resulting from the inhomogeneity of the plastic flow, and $\mathbf{F}^e$ is a superposed compatible deformation due to the applied tractions. The incorporation of disclinations is straight forward and achieved by adding a rotational component to the multiplicative decomposition such that it now takes the form $\mathbf{F} = \mathbf{F}^e\mathbf{F}^i\overline{\mathbf{R}}\mathbf{F}^p$ where all of the previous terms carry the same meaning and the $\overline{\mathbf{R}}$ captures the contribution of disclinations to the deformation history.

The proper inclusion of disclinations into the kinematics of defective crystals requires additional degrees of freedom at each continuum point other than the three displacement components. This is achieved by attaching a triad of rigid director vectors to each material point that can rotate independently of the material spin of an infinitesimal line segment. In this type of oriented continuum, geometrically necessary dislocations are defined in terms of incompatibility of the strain field, and geometrically necessary disclinations are defined in terms of incompatibility of the curvature field. This type of material falls under the class of generalized continua and is called a micropolar material. Traditionally, micropolar materials are associated with non-symmetric macroscopic Cauchy stresses, due to the presence of couple stresses stemming from the independent rotations which cause non-symmetric shear strains.

However, the model proposed here maintains the symmetry of the macroscopic Cauchy stress, and employs the micropolar balance laws to the micromechanical problem of determining the internal stresses arising from geometrically necessary dislocations and disclinations.

For simplicity, a linearized version of the model is presented below. The crux of the model revolves around the additive decomposition of the displacement field into two compatible components $\bar{\mathbf{u}}$ and $\hat{\mathbf{u}}$. The first component, $\bar{\mathbf{u}}$, is equal to the total displacement prior to the onset of plasticity, and the second component, $\hat{\mathbf{u}}$, results from plasticity and is governed by the micropolar balance equations to the micromechanical problem. In addition to the these six displacement degrees of freedom, the independent micropolar rotation, $\hat{\boldsymbol{\phi}}$, is introduced as part of the micromechanical problem. The governing equations of the proposed model are summarized below. The driving force for plasticity is the Cauchy stress, whereas the micropolar stress

Displacement decomposition:	$\mathbf{u} = \bar{\mathbf{u}} + \hat{\mathbf{u}}$
Total strain:	$\boldsymbol{\varepsilon} = sym\left(\nabla \mathbf{u}\right) = \bar{\boldsymbol{\varepsilon}} + sym\left(\hat{\boldsymbol{\varepsilon}}\right)$
Cauchy stress:	$\bar{\boldsymbol{\sigma}} = \mathbf{C} : \bar{\boldsymbol{\varepsilon}}$
Macroscopic balance laws:	$div\, \bar{\boldsymbol{\sigma}} = \mathbf{0}, \quad \bar{\boldsymbol{\sigma}} = \bar{\boldsymbol{\sigma}}^T$
Micropolar strain and curvature:	$\hat{\boldsymbol{\varepsilon}} = \hat{\mathbf{u}}\nabla + \mathbf{e} : \hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\varepsilon}}^e + \hat{\boldsymbol{\varepsilon}}^p$ $\hat{\boldsymbol{\kappa}} = \hat{\boldsymbol{\phi}}\nabla = \hat{\boldsymbol{\kappa}}^e + \hat{\boldsymbol{\kappa}}^p$
Micropolar stress and couple stress:	$\hat{\boldsymbol{\sigma}} = \mathbf{A} : \hat{\boldsymbol{\varepsilon}}^e, \quad \hat{\boldsymbol{\mu}} = \mathbf{B} : \hat{\boldsymbol{\kappa}}^e$
Strain and curvature flow rules:	$\dot{\hat{\boldsymbol{\varepsilon}}}^p = \sum_\alpha \dot{\gamma}^\alpha \bar{\mathbf{s}}^\alpha \bigotimes \bar{\mathbf{m}}^\alpha, \quad \dot{\hat{\boldsymbol{\kappa}}}^p = \sum_\alpha \dot{\lambda}^\alpha \mathbf{Q}^\alpha$
Slip and spin rates:	$\dot{\gamma}^\alpha = f\left(\bar{\boldsymbol{\sigma}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\mu}}, \boldsymbol{\alpha}, \boldsymbol{\theta}\right)$ $\dot{\lambda}^\alpha = g\left(\bar{\boldsymbol{\sigma}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\mu}}, \boldsymbol{\alpha}, \boldsymbol{\theta}\right)$
Dislocation and disclination densities:	$\alpha_{kl} = e_{kmn}\hat{\varepsilon}^e_{nl,m} - \hat{\kappa}^e_{lk} + \hat{\kappa}^e_{mm}\delta_{kl}$ $\theta_{kl} = e_{kmn}\hat{\kappa}^e_{nl,m}$
Micropolar balance laws:	$div\, \hat{\boldsymbol{\sigma}} = \mathbf{0}, \quad div\, \hat{\boldsymbol{\mu}} + \mathbf{e} : \hat{\boldsymbol{\sigma}} = \mathbf{0}$

and couple stress enter the flow rules as back stresses/moments representing the long-range internal stresses due to presence of geometrically necessary dislocations and disclinations, i.e. higher-order deformation gradients.

4. Conclusions. The proposed generalized continuum model of crystal plasticity accounting for both dislocation and disclination effects is currently a work in progress. A foundation for incorporating disclinations into the kinematics of crystal plasticity and accounting for the balance of the internal force and moment stresses due to geometrically necessary defects has been laid. Unlike the traditional micropolar material, our model retains the symmetry of the macroscopic Cauchy stress which is in line with experimental observations. It is believed that the model will have a wide-range of predictive capabilities including capturing the enhanced role of grain boundaries in the deformation of nanocrystalline materials to the fragmentation and fracture processes of severely deformed polycrystals with more conventional grain sizes. The focus of future work will be on developing the numerical methods and tools necessary to implement the model into a finite element code.

DEPOSITION OF NANOPARTICLES ON TEXTURED SURFACES THROUGH SPIN- AND DIP-COATING

H. MOORE* AND B. SIMMONS†

1. Introduction. Patterned arrangements of spherical gold colloids are crucial for the development of advanced nano-sensors and nano-electro mechanical systems (NEMS). The main methods for Au nanosphere deposition are spin coating and dip coating on chemically patterned substrates, mechanically embossed nano-channeled polymer substrates, or a combination of the two. Research this summer investigated the effect of varying the particle size, spin speeds, substrate wettability, channel width and substrate chemical functionality.

2. Results. Several different Zeonor 1060 and PMMA VS100 substrates were embossed using a Ni wafer with 200 nm width channels or a Si wafer with 600 nm channels by heating the substrates up to 240°F and then applying a 3500 lbs. load with a Carver press shown in Figure 2.1.

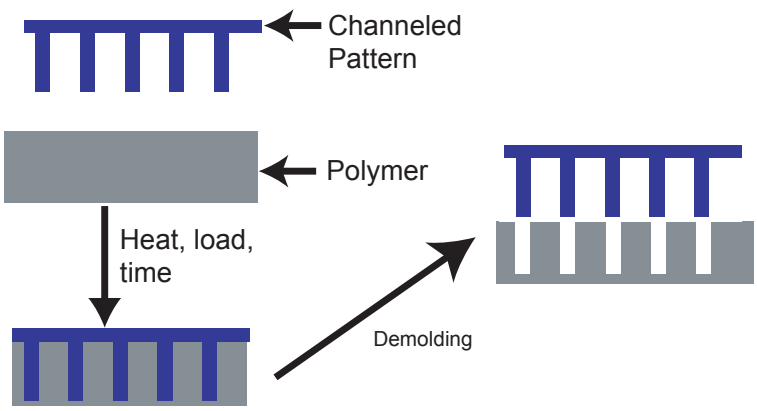


FIG. 2.1. *Nanoembossing Schematic*

Once the substrates were allowed to cool some of the Zeonor patterned substrates were sputtered with gold to create three different surface interface materials with different surface energies and wettabilities. The Zeonor interface is the most hydrophilic, gold is moderately hydrophilic and PMMA is hydrophobic. The more hydrophilic a material surfaces the smaller the contact angle between the droplet and the substrate will be upon wetting. This translated to increased substrate surface area in contact with the droplet for the same droplet volume. Theoretically, the more hydrophilic a surface is the more Au particles would come in contact with the substrate surface for the same droplet unit volume and increase the probability of successful deposition of Au particles on the substrate. Conversely, the more hydrophobic a material surface is the larger the contact angle is formed and the lower the probability of Au colloid deposition on the substrate surface.

*Georgia Institute of Technology
†Sandia National Laboratories

The three substrates were dip coated by submersing them into a solution containing 20 nm spherical Au colloids and then withdrawn mechanically at a constant rate of ∼1 mm/min. This process schematic can be seen in Figure 2.2.

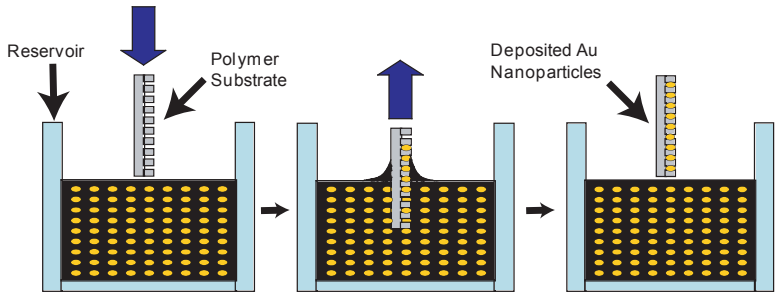


FIG. 2.2. *Schematic Dip Coating*

The withdraw angle was varied from the vertical position to almost horizontal position with unsuccessful deposition onto any of the substrates for any angle. The unsuccessful deposition is attributed to the unfavorable force balance in the free meniscus system where the surface tension and Stokes drag force dominate the particles motion causing the Au colloids to slide off the substrate with the solution. The Stockes drag force F_{drag} is given by the expression

$$F_{drag} = 6\pi U_0 \mu a, \tag{2.1}$$

where U_0 is the velocity of the particle relative to the fluid and a is the particles radius. The Stokes drag force is applied to small particles traveling at slow speeds relative to the fluid and typically associated with fluid systems with Renolds numbers less than 1.

Spin coating was the next deposition technique investigated to utilize centrifugal forces created by spin coating to fill the nano-grooves and over come the unfavorable force balance. Different solutions containing approximately uniformly sized Au spherical colloids were dropped on to the different substrates and spun at speeds varying from 250 rpm to 1500 rpm to spread the solutions across the substrates. Theoretically the Au colloids would be deposited in the nanochannels as the water is spun off of the substrate as shown in Figure 2.3. The actual results can be seen in the SEM images in Figure 4, which show a larger quantity and a higher packing density of Au particle as the particle size increased and spin speeds decreased. The increase in particle size increased the favorable centrifugal force on the particle and the decrease in spin speed decreased the unfavorable Stokes drag force, thus validating the initial hypothesis.

The solution surface tension and the substrate functionality were altered by the addition of ethanol to the Au colloid solution which decreased the solutions surface tension and oxidizing the substrates surface by using an O_2 clean. The decrease in the surface tension caused an increasing the contact angle which was thought to be favorable for reasons described above. The effects of the oxidation and addition of ethanol on Au deposition were inconclusive as can be seen Figure 2.5.

The next method for Au colloid deposition investigated was chemical fictionalization. Zeonor 200 nm channeled substrates were coated with 100 angstroms of gold then chemically patterned with either 1 gm/1mL or 5 gm/mL solutions of 1,4-benezinedithol in ethanol via microimprint lithography to functionalize the top of

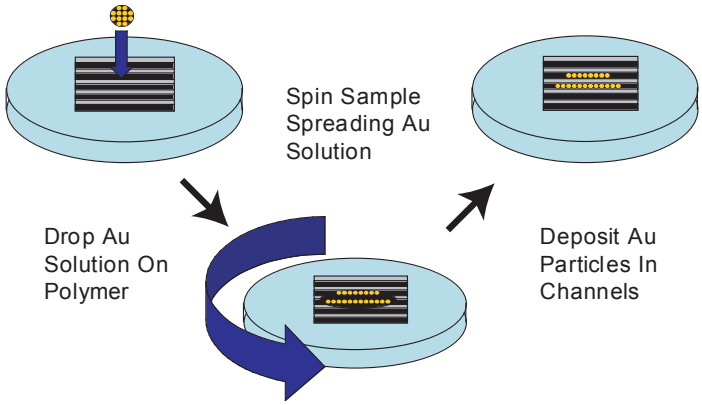


FIG. 2.3. *Spin coating schematic*

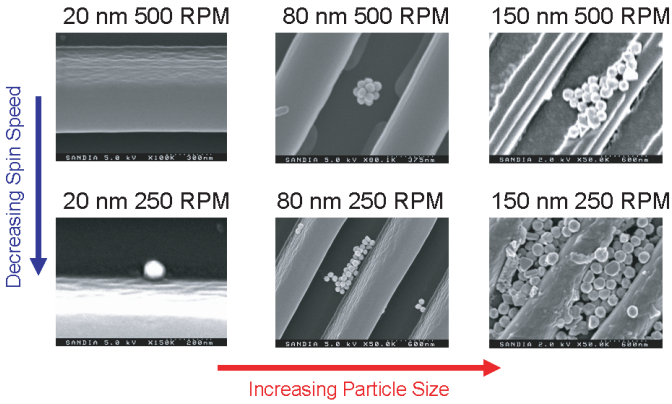


FIG. 2.4. *Spin Coating Results*

the nanochannels on the substrates surface. After dissolving 1,4-benezinedithol in ethanol, the surface of polymer stamp touched the solution to create a thin film on the stamp. The thin film solution was transferred by lightly touching the surface of the stamp to the nanochanneled gold plated substrate. Once the ethanol evaporated, the 1,4-benezinedithol remained on the tops of the channels on the substrate as displayed in Figure 2.6.

The SEM results of spinning relatively monodispersed Au colloids with diameters of 80 nm, 150 nm and 250 nm are shown in Figure 2.7.

The variation in particle size showed insignificant effects on the gold particle deposition compared to the concentration of 1,4-benzinedithol in the solution. The increase in the concentration of the 1,4-benezinedithol caused an increase in Au colloid deposition number and the packing density. The inaccuracy of the nanoimprint lithography technique caused the Au colloid deposition to be not completely restricted to the tops of the channels, however does show the feasibility of using physical and chemically patterned substrates for the directed assembly of Au colloids.

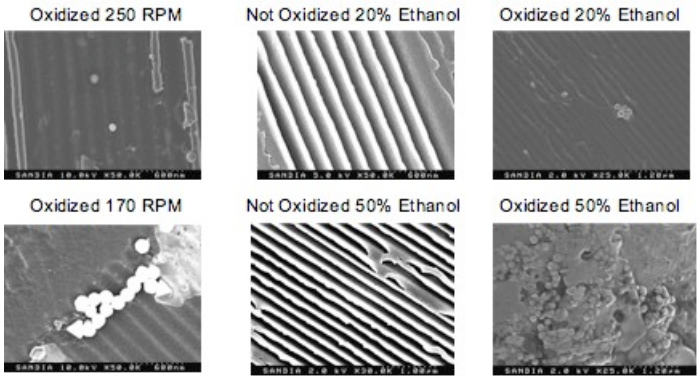


FIG. 2.5. *Oxidization of substrate surface and addition of ethanol*

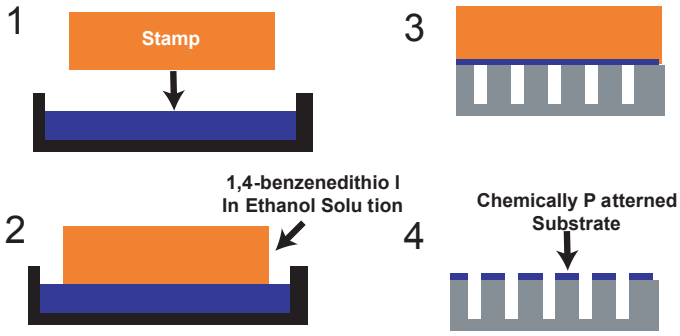


FIG. 2.6. *Nanoimprint lithography schematic*

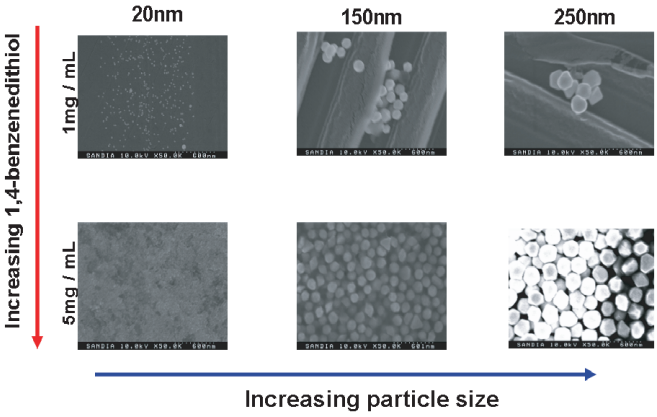


FIG. 2.7. *Chemical fictionalization with 1,4-benezinedithol results*

ATOMIC SIMULATIONS OF STRESS EVOLUTION DURING THIN FILM GROWTH

C-W. PAO*, E.B. WEBB III†, S.M. FOILES‡, J.A. FLORO §, AND D.J. SROLOVITZ¶

Abstract. As first presented by Laplace 200 years ago, curvature driven pressure, or stress, is inversely proportional to the dimension of an object. Put simply, smaller objects are more greatly affected by surface and interface stress. Understanding stress evolution during growth of nanometer scale objects as well as thin film materials is paramount to optimizing nano and mesoscale device performance. During thin film growth, many materials exhibit significant compressive stress evolution after the underlying substrate has been completely covered. During this stage of growth, a continuous film of the growing material exists and depositing adatoms are encountering a surface typically rich with defects such as atomic steps and grain boundaries. Mechanisms of stress evolution at this stage are difficult to establish given the complexity of the system and the required resolution for direct experimental observation. This work presents molecular dynamics simulation results to investigate stress evolution during later stages of thin film growth. Simulations were performed of Ni adatoms being deposited onto a film of Ni with the free surface intersected by grain boundaries. A significant percentage of deposited atoms diffuse to the grain boundary where they are rapidly incorporated into the structure below the surface. This process is demonstrated to generate significant compressive stress. Results presented show stress in the simulations is larger in magnitude than is observed in experiment. We show preliminary evidence that this discrepancy is a result of modeling perfect crystal surfaces (i.e. away from the grain boundaries). More realistic surfaces have atomic steps on the surface, which we show to trap adatoms and prevent adatom diffusion to the grain boundary, thereby lowering the magnitude of compressive stress generated for a given amount of deposited material. Simulation data will guide development of a new, predictive model for compressive stress evolution in thin films.

1. Introduction. Growth of materials via methods like chemical vapor deposition or physical vapor deposition is important to a very large number of technologies; the automotive and microelectronics industries are but two of many that are impacted by thin film science and engineering. Residual stress in small objects and thin films strongly impacts the corresponding behavior of the film. From a mechanical point of view, residual stress can result in undesired behavior in the form of bending, warping, or failing. Less intuitive is the tie that has been observed between residual stress and, for instance, ferroelectric response in some oxides [9]. Similarly, a tie has been observed between residual stress and the optical response of some semiconductor materials [10]. Being able to engineer residual stress is thus critical to maximizing the benefit we obtain from emerging nano and mesoscale devices. Predictive models that connect growth processes to resulting stress in materials would greatly reduce design costs by guiding engineers into narrower process windows, giving quicker turnaround times. Clearly such models depend upon an intimate knowledge of the fundamental mechanisms of stress evolution during growth.

Growing thin films are subject to stress evolution from a number of sources [6]. Two that are fairly well understood are stresses that evolve due to lattice mismatch and due to thermal expansion mismatch. The former refers to the effect of forcing a material to grow at a lattice spacing different from its equilibrium value and thereby building inherent strain (and stress) into the structure. Thermal expansion mismatch is, of course, most manifest when the temperature of the system makes wide traverses.

*Princeton University, cpao@princeton.edu
†Sandia National Laboratories, ebwebb@sandia.gov
‡Sandia National Laboratories, foiles@sandia.gov
§University of Virginia, floro@virginia.edu
¶Princeton University, srol@princeton.edu

Film and substrate don't deform the same with temperature change, resulting in stress build-up. While these sources of stress evolution are fairly well understood, stresses that evolve as a result of atomic scale behavior inherent to the growth process are not. An important example can be found in materials that grow in the Volmer-Weber (VW) or island mode [6]. Such materials exhibit three morphological regimes: at early time, isolated islands nucleate on the substrate and grow; at intermediate time, growing islands impinge and coalesce; at late time, a complete film covers the substrate and deposition results in film thickening. For high surface mobility materials, each of these growth process regimes can be associated with a specific stress state: compressive at early time, tensile at intermediate time, and compressive again at late time. Much has been conjectured about the atomic behavior inherent to each growth process regime and how the behavior results in observed stress. For instance, it is fairly well accepted that, during the intermediate time regime when growing islands are impinging and coalesching, they are stretching to close gaps between them and eliminate undesired surface energy [8]. While this results in tensile stress evolution, it lowers the system energy by removing surface energy contributions. Despite this fairly well accepted mechanism, questions remain. Indeed, a great deal of uncertainty still exists, particularly for the late time regime.

In the late time regime, most systems have developed significant grain structure in the film. For instance, each island growing on the surface at early time may have the same crystallographic orientation perpendicular to the surface. However, their particular orientation in the plane of the surface can vary between adjacent growing islands. As such, when they impinge and coalesce, a tilt grain boundary (GB) will be formed between them. Once the film fully coalesces and begins thickening (i.e. the late time regime), there is typically a collection of different tilt GBs intersecting the film surface. Deposited adatoms behave in some fashion to generate compressive stress at this stage. Two prevalent theories exist attempting to explain this behavior. One theory proposes that kinetic effects in conjunction with compressed surface ledges or steps act during deposition to allow adatoms to be trapped into an otherwise perfect crystalline lattice [12]. Such interstitial atoms create significant compressive stress so, if subsequent growth buried them in this state, they could be the reason for compressive stress evolution. Note that this theory does not depend upon GBs being present but does depend upon other surface defects (steps). In another model, adatoms are driven into existing GBs due to an increased chemical potential on the free surface that results from dynamic impingement of adatoms (i.e. deposition) [3]. Adatom incorporation into GBs should also produce compressive stress but the magnitude of the effect is less understood and it is even unclear if such incorporation does occur.

Herein, we present molecular dynamics simulations of deposition onto a metal surface intersected by tilt GBs. We examine atomic behavior near and far from the GBs along with corresponding stress behavior. From these results we evaluate the models discussed above. We also review necessary improvements to our calculations to further elucidate stress evolution during this complex stage of growth.

2. Procedure. Interatomic potential energy in the simulations was described via the embedded atom method (EAM) wherein the potential energy for N atoms is [4]

$$E = \sum_{i=1}^{N} \left[F_i(\rho_i) + \frac{1}{2} \sum_{j \neq i} \phi_{ij}(R) \right]. \tag{2.1}$$

Here, ρ_i is the electron density at atom i,

$$\rho_i = \sum_{j \neq i} {\rho_j}^a(R), \tag{2.2}$$

where ${\rho_j}^a(R)$ is the spherically symmetric electron density contributed by atom j, a distance R from i. $F_i(\rho_i)$ is the energy associated with embedding atom i into an electron density ρ_i; this provides a many-body nature to the model. $\phi_{ij}(R)$ is a pair potential between atoms i and j; in the EAM model used herein, it represents core/core repulsion. The many-body nature of the EAM makes it superior over pair potentials for modeling metals and it has been widely used to study bulk, surface, point defect, and alloy behavior [5]. We model Ni deposition onto a Ni(111) film surface; the parameters for our Ni EAM potential are discussed in [7]. Except where noted, all runs are performed at $T = 763$ K as this corresponds to half the melting point; temperature is maintained in the simulation via a Nose-Hoover thermostat [1]. Although many deposition processes are performed at room temperature, temporal constraints inherent in the MD method encouraged us to model an elevated temperature such that atomic diffusion and other processes would be accelerated. This enhances our ability to observe phenomena in MD time scales. Near the conclusion of our article, we address possible implications of elevated temperature.

Prior to depostion, it was necessary to form a Ni(111) free surface intersected by grain boundaries; initially, a bulk system (no free surfaces) was created. It is difficult to characterize GB structures in a film from a typical deposition experiment; however, it is often observed that, for metal systems, (111) texture emerges in the growth direction. For this reason, we chose to model a $\Sigma 79$ coincident site lattice boundary; this nomenclature indicates that, 1 in every 79 sites that would match up in a perfect crystal (i.e. no boundary) actually do match up across the boundary. Put simply, the lower the Sigma number for a coincident site lattice designation, the higher the coherency or order across that boundary. The $\Sigma 79$ boundary can be characterized as having intermediate coherency and thus represents well a fairly generic GB. To form this boundary, two half-crystals are modeled with the (111) crystallographic direction along z. One half-crystal is rotated 33.99 deg relative to the other half crystal; thus, this is a symmetric tilt boundary. The boundary planes that meet are $(-3\ -7\ 10)$ and $(3\ -10\ 7)$; these planes (i.e. the plane of the GB) are made normal to the x direction in the simulation box. Periodic boundary conditions are applied in all three dimensions (i.e. a bulk geometry is initially assumed). This implies that there are actually two GBs normal to the simulation x direction, separated from one another by half the simulation cell dimension in x and with opposite rotations. Simulation cell dimensions in x and y were held constant throughout all simulations to keep the lattice constant of the crystal (away from the GB) at its equilibritum value for the temperature studied. The dimensions of the system thus formed were 85 Å in x, 40 Å in y, and 180 Å in z. Following extensive equilibration of the GB structure in the bulk, periodic boundary conditions were removed in z, resulting in two free (111) surfaces intersected by the $\Sigma 79$ GBs. Atoms in a slab 12 Å thick in z at the 'bottom' surface, were held rigid for subsequent simulations to mimic the slab being attached to some underlying substrate (i.e. a film). Empty space was defined to exist above the 'upper' surface of the film in z and the system was extensively equilibrated in this surface (film) geometry prior to deposition [2].

Deposition was then modeled by inserting adatoms into the empty space above the film free surface in z. Adatoms were placed in a random position in x and y at a

distance above the surface slightly larger than the planar separation in (111) for Ni at this temperature. This insertion process assumes the ballistic part of atomic deposition is essentially concluded and permits more rapid equilibration of the added atom. For one case, we repeated adatom deposition, allowing atoms to ballistically deposit onto the substrate by inserting them at a distance above the surface slightly less than the interaction cut-off. As expected, these adatoms took longer to equilibrate but there were no differences in the primary conclusions of this work. As such, only simulations employing the former method are discussed herein. The time between adatom depositions was 50 ps; given the temporal constraints on MD, it is not surprising that simulation deposition flux is many orders of magnitude greater than in experiment. We address this issue in the context of our results in the following section.

3. Results and Discussion. Figure 3.1 shows a snapshot from the simulation after 100 adatoms have been deposited onto the film surface. Although all atoms are Ni, adatoms are colored differently from film atoms to assist discussion. In the view shown, the $\Sigma 79$ GBs are located in the middle of the picture and at the edges of the picture (i.e. one GB is straddling the periodic bound in x — the horizontal in the figure). Is is easy to observe that a significant percentage of the adatoms deposited have found their way to the GBs where many have incorporated into the structure below the free surface. In some cases, this adatom incorporation into the GB results in what was a film atom being forced out onto the free surface. Howerver, in many cases, this does not occur. There can also be observed a number of atoms forming a cluster away from either GB. So, some fraction of deposited atoms diffuse across the surface until they interact with one of the GBs. When they do, some percentage of these adsorb or incorporate into the film at the GB. This serves as striking evidence in support of the model for compressive stress generation that asserts GB incorporation is a plausible phenomenon. Note that we did not observe adatom trapping into the structure away from the GB. However, as pointed out in the Introduction, trapping is assumed to be assisted by neighboring steps or ledges that generate a compressive stress due to the corresponding atomic interactions. Thus, the absence of steps (at least at the start of deposition) could prevent observing this mechanism. Also, as mentioned in [12], the compressive interaction between steps may be material dependent so this mechanism may not emerge in all systems.

We wish to evaluate the stress evolution in the film during deposition and attempt to coordinate it with the GB incorporation behavior observed. In stress measurement experiments, what is actually determined is the product of the stress in the film multiplied by the mean thickness of the film (material deposited is assumed to form a uniform film with minimal roughness). To compare to this quantity directly, we calculate the stress-thickness product in our simulation; the method for doing so is covered in [11]. Data obtained are shown in Figure 3.2 and it can be seen that, over the first 100 atoms deposited, a very large compressive stress is generated in the model film. Because we can specifically track where each atom in the system is, we can easily determine how many extra atoms have incorporated into the film. A direct correlation is seen between the magnitude of compressive stress and the number of extra atoms in the film. This proves that, although processes on the surface may also be influencing stress generation, the prominent compressive stress generation mechanism in our simulations is adatom incorporation at GBs. While this is encouraging with regard to existing models, the magnitude of the response in the simulation is significantly larger than in experiment. Because data are plotted versus coverage (rather than time), this difference in stress response magnitude cannot be directly attributed to the difference

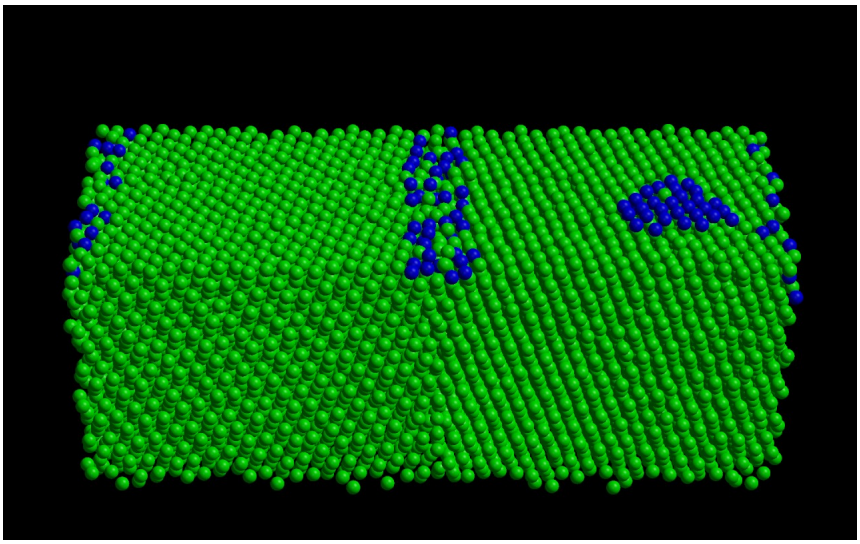


FIG. 3.1. *Snapshot from the film deposition simulations after* 100 *adatoms have been deposited. Atoms that belonged to the film prior to deposition are green and adatoms are blue.*

in deposition flux between experiment and simulation. Nonetheless, we repeated the simulation allowing only 5 ps between adatom additions to the system (i.e. an order of magnitude increase in flux). Little effect was observed on the stress response; however, it cannot be ruled out that a flux rate effect may emerge at significantly lower fluxes.

One way to assess the role of flux rate is to perform a long simulation with only one deposited adatom and to observe the behavior of that adatom in the absence of any further deposition. This permits one to assess the interaction between the adatom and the surface in a less complicated geometry. In all cases where we did this (i.e. we varied the starting random position), the deposited atom eventually diffused near the GB and very quickly incorporated into it. In fact, if the adatom was deposited within ≈ 7 Å of the center of the GB (in x), incorporation typically occurred in less than 10 ps. Furthermore, this incorporated adatom was never observed to leave the GB. In the absence of highly tedious free energy calculations (beyond the scope of this work), these observations support the notion that adatoms are more energetically favorable when incorporated into the GB as compared to being on the free surface. Thus, while a flux effect may exist, the fundamental mechanism of adatom incorporation into GBs is still expected to be a relevant contributor to compressive stress generation. What should be noted regarding the existing model is that the authors' contend adatoms are driven into GBs due to an increased surface chemical potential due to deposition flux [3]. However, our calculations with single adatoms indicate that the chemical potential in the GB may be lower than at the free surface, regardless of flux. That is, flux may enhance this relationship but it appears that adatom incorporation may be thermodynamically favored even in the absence of a flux.

Another possibility for the discrepancy between the compressive stress magnitude in experiment versus simulation is the difference in temperature (although this is unlikely since some experiments have been performed at elevated temperature and the difference exists in data from those experiments as well). Furthermore, a few single

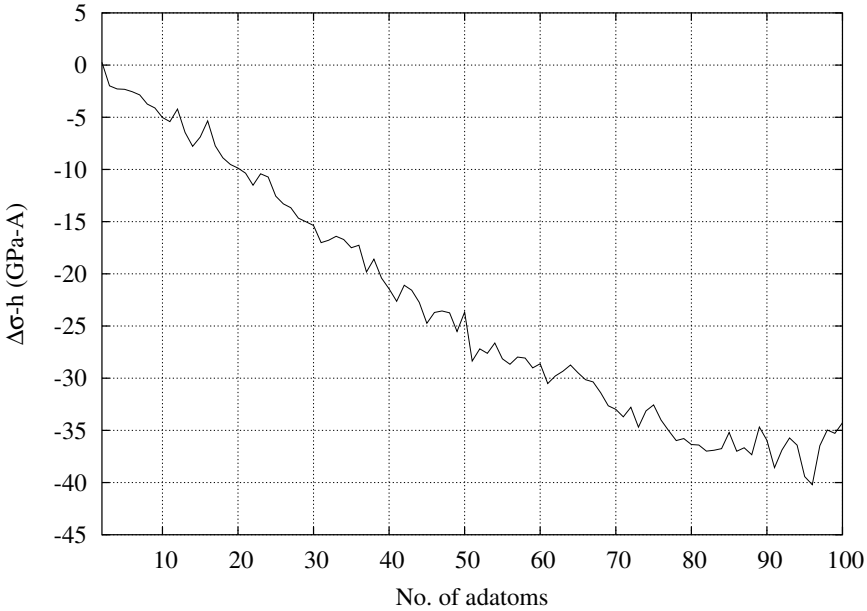


FIG. 3.2. *Calculated stress-thickness product as a function of deposition onto the model film.*

adatom simulations were performed for $T = 300$ K and, although surface diffusion was greatly reduced, atoms still incorporated into a GB once they got within ~ 7 Å of the GB center (in x). In addition, the kinetics of incorporation, once an adatom got close enough to a GB, were very similar at different temperatures. This indicates that, while diffusion to the GB will be slower at room temperature, adsorption into the GB is essentially spontaneous once the adatom interacts sufficiently with the boundary. A more likely suspect for the discrepancy between simulation and experiment stress magnitude was our use of a fairly ideal structure system. In other words, our free surface is absent of any surface defects other than the GBs. Images from thin film growth experiments reveal surface steps in relative abundance on single crystal grains. In order to assess the effect of surface steps on adatom incorporation into GBs, we performed deposition simulations onto a surface with steps on it. It is a subject of ongoing research as to whether adatoms are more energetically favorable attached to steps versus adsorbed in GBs. However, in our simulations, we observed that adatoms that diffused near to a step would readily attach at the step and, at least for the duration of our MD simulations, remain there. This indicates that surface steps may be diffusion traps for adatoms. As such, even if they would be lower in energy at a GB, adatoms are kinetically limited from every finding the GB if they encounter a surface step first.

4. Conclusions. We have used MD simulations to examine atomic behavior relevant to compressive stress generation during thin film growth. In accord with an existing theory, adatoms deposited on crystal surfaces intersected by grain boundaries

diffuse to the GBs and readily incorporate into them. Furthermore, this behavior is seen to correspond with compressive stress generation in the film. A discrepancy between experiment and simulation was noted in that the magnitude of stress generation was significantly greater in simulation than experiment. It is known that, in experimental samples, steps are present on crystal surfaces. When deposition was modeled onto a surface with crystal steps, the steps were demonstrated to act as diffusion traps, preventing adatoms from reaching a GB. This reduces adatom incorporation, and therefore compressive stress generation, for a given coverage. Future work will attempt to gather these various effects into an analytical model describing stress evolution as a function of thin film growth.

REFERENCES

[1] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon, Oxford, 1987.

[2] Authors' Note: Model grain boundary construction is a complex process involving molecular dynamics, Monte Carlo methods, and energy minimization procedures. As a detailed discussion of this is beyond the scope of this article, the interested reader is directed to S. M. Foiles (foiles@sandia.gov).

[3] E. Chason, B. W. Sheldon, L. B. Freund, J. A. Floro, and S. J. Hearne, *Origin of compressive residual stress in polycrystalline thin films*, Phys. Rev. Lett., 88 (2002), p. 156103.

[4] M. S. Daw and M. I. Baskes, Phys. Rev. B, 29 (1984), pp. 6443–6453.

[5] M. S. Daw, S. M. Foiles, and M. I. Baskes, *The embedded-atom method: a review of theory and applications*, Mater. Sci. Rep., 9 (1993), pp. 251–310.

[6] J. A. Floro, E. Chason, R. C. Cammarata, and D. J. Srolovitz, *Physical origins of intrinsic stresses in volmer-weber thin films*, MRS Bulletin, 27 (2002), pp. 19–25.

[7] S. M. Foiles and J. J. Hoyt, Acta Mater., 54 (2006), p. 3351.

[8] S. J. Hearne, S. C. Seel, J. A. Floro, C. W. Dyck, W. Fan, and S. R. J. Brueck, *Quantitative determination of tensile stress creation during island coalescence using selective-area growth*, J. Appl. Phys., 97 (2005), p. 83530.

[9] G.-F. Huang and S. Berger, *Combined effect of thickness and stress on ferroelectric behavior of thin batio$_3$ films*, J. Appl. Phys., 93 (2003), pp. 2855–2860.

[10] J. Lee, W. Gao, Z. Li, M. Hodgson, J. Metson, H. Gong, and U. Pal, *Sputtered deposited nanocrystalline zno films: a correlation between electrical, optical, and microstructural properties*, Appl. Phys. A, 80 (2005), pp. 1641–1646.

[11] C. Pao and D. J. Srolovitz, Phys. Rev. Lett., 96 (2006), p. 186103.

[12] F. Spaepen, *Interfaces and stresses in thin films*, Acta Mater., 48 (2000), pp. 31–42.

USING THE FOCUSED ION-BEAM TOOL FOR SITE-SPECIFIC ATOM-PROBE TOMOGRAPY ANALYSIS OF CU ALLOYS

J. RIESTERER* AND E. MARQUIS†

Abstract. Grain boundary triple junctions have been shown to impact the behavior of grain boundaries, but very limited experimental experiments have been carried out to understand dopant segregation at triple junctions and the driving forces at play. Characterizing triple junctions can be very challenging. Only a few limited transmission electron microscopy observations have been reported so far. Because of the three-dimensional character of triple junctions, atom probe tomography would be the technique of choice, providing atom-by-atom three-dimensional mapping of small volumes of material. We are therefore developing techniques to prepare atom probe specimens containing triple junctions and grain boundaries of known character. After annealing to promote diffusion of dopants within the Cu samples and ensure stability of the boundary structures, site-specific atom probe tips containing grain boundaries and triple junctions are prepared. The analyzed boundaries and junctions are characterized by electron backscatter diffraction (EBSD) prior to focused-ion beam milling of the areas of interest.

1. Introduction. The significance of grain boundaries and triple junctions in impurity segregation has long been known, however, not fully understood. Au is a natural impurity in Cu, while Bi segregation to Cu grain boundaries is known to cause embrittlement. Segregation of impurities to the grain boundaries may result in an impurity concentration orders of magnitude higher at the grain boundaries than in the bulk [32]. Until recently, atom-by-atom three-dimensional mapping of these small volumes was not possible. The development of atom-probe tomography (APT) has enabled the investigation of structure and chemistry at grain boundaries and triple junctions in Cu alloys. By definition, triple junctions are 3-D objects. Traditional analysis methods, such as transmission electron microscopy (TEM), are only capable of 2-D renderings. APT combines a time-of-flight mass spectrometer with a position-sensitive detector, allowing samples to be reconstructed in all three dimensions.

Careful sample preparation is the key to successful microscopy and tomography of these materials structures. Analysis of as-received, annealed and/or doped Cu may be performed after preparing suitably thin atom probe tips using a FIB. The focused ion-beam (FIB) tool allows site-specific features to be plucked from the bulk. The goal here is to measure Au and Bi segregation to grain boundaries and triple junctions with known character in Cu-alloys using the APT.

Embrittlement in the Cu-Bi system was first observed by Hampe in 1874. When Bi is introduced to Cu, the ductile-to-brittle transition is facilitated via intergranular fracture [1]. Questions remain regarding where Bi sits in Cu grain boundaries. One model proposes that Bi enhances grain boundary roughening, reducing the grain boundary adhesion, leading to embrittlement [31]. High-resolution transmission electron microscopy (HRTEM) [6, 7], scanning transmission electron microscopy (STEM) [1], Auger electron spectroscopy (AES) [5–7], energy dispersive spectroscopy (EDS) [1], and electron energy loss spectrometry (EELS) [1] have all been used to analyze grain boundary structure and segregation levels of Bi in Cu. Chang, et al [4,5] have been working on clarifying the Cu-Bi phase diagram. The position of the Cu-solidus line has been shown to dramatically change the Bi segregation levels and grain boundary structure in Cu. The solid solubility of Bi in Cu was measured to be 100 ppm at 1100 K [5]. However, segregation to the grain boundaries depends on the

*University of Minnesota
†Sandia National Laboratories

grain boundary type and the grain boundary planes [31]. Using Changs phase diagram, Divinski, et al. proposed premelting of Cu grain boundaries in the presence of Bi [6, 7].

Alber, et al. [1] did an extensive study of symmetric and asymmetric boundaries of different misorientations at different temperatures. The concentration of Bi was shown to change the grain boundary faceting. Grain boundary segregation was also shown to be driven by Bi dissolved in the bulk, rather than Bi at the surface or in precipitates. The final conclusion made was that the higher the grain boundary energy, the stronger the Bi segregation will be.

Besides being found naturally as an impurity in Cu, Au forms a stable long-period superlattice structure with Cu [15]. When concentrations approach equiatomic, Au will increase age-hardenability of Cu, especially in the presence of Ga [26]. Prokofjev investigated segregation behavior of Au in Cu boundaries of known misorientation and proposed a structural transformation of the boundaries with the change of segregation enthalpy [29, 30]. However, little is known about the 3-dimensional grain boundary structure.

The effect of triple junctions on diffusion and segregation is still not well known. Triple junctions are 3-dimensional structures and become increasingly more important as science pushes into nanoscale regimes [13]. The large volume available at triple junctions for impurity segregation and the high grain boundary energies cause triple junctions to be the preferred segregation sites [14].

2. Atom-probe tomography. APT has the highest resolution of any microscopy technique; lateral and depth resolutions of less than 1 nm may be routinely achieved [25]. Single atoms can be mapped on the needle surface spatially and chemically [17]. The technique allows microstructure, such as precipitates at grain boundaries, to be imaged [8, 25]. Point defects, dislocations and twin boundaries may also be imaged [25].

The APT combines a field ion microscope (FIM) with a time-of-flight mass spectrometer. Mủller was the first to develop FIM in 1951 [27]. The FIM draws atoms from the sample surface by using an ionizing gas and high electric fields of 20-50 V/nm to induce field evaporation [21]. The electric field controls the rate at which atoms are evaporated [27]. The technique requires samples to be conductive and in the shape of a sharp needle with tip radius of 50 nm or less. Once evaporated from the surface, the ionized atom passes through an aperture to a position-sensitive detector and the time-of-flight is measured. Figure 2.1 illustrates the operation of an APT. The original atomic position and the position struck on the detector are directly correlated [27]. The time-of-flight and mass-to-charge ratio are used to identify the atoms chemical nature. Data is reconstructed in 3-dimensions, typically as a colored-coded dot map. Each dot represents an atom in space and the color identifies the chemical nature [24].

Difficult specimen preparation, high surface-to-volume ratio and image interpretation are issues with this technique [8]. The efficiency of atoms detected ranges from 50-60% of atoms evaporated, making vacancy determination difficult. Features of interest need to be sufficiently close to the tip apex to be imaged, thus sample preparation is difficult. Traditionally, tips are fabricated via electropolishing; feature positioning using this preparation method is tedious [19]. Site-specific tips may be fabricated using a FIB tool [21].

3. Focused ion-beam tool. The ability to prepare samples in cross-section, rather than plan-view, to investigate the characteristic properties is not trivial. In traditional mechanical methods, a significant amount of polishing is usually necessary.

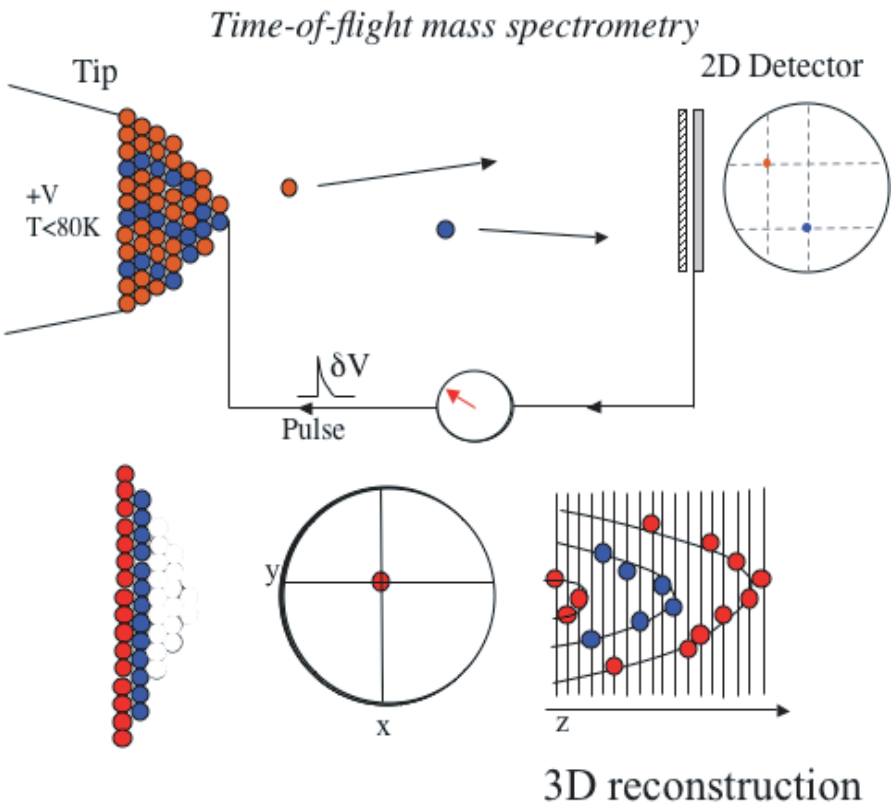


Fig. 2.1. *Schematic representation of APT operation.*

Damage to the cross-section surface is nearly always present as a result [37]. Traditional methods may take an enormous amount of time. Chemical etching is difficult and often carries with it safety concerns. Site-specific features can not be easily characterized due to the imprecision of the human hand. The FIB tool is the solution to most sample preparation issues [11]. Cross-section samples of integrated circuits [2], Si [33, 35], WC [3], FeAl [3], silica-zirconia membranes [16], glass substrates [36], and nanoparticles [28] are just a few examples of samples prepared by using ions to mill away a trench in the sample for cross-sectional viewing using the SEM or TEM [11,37].

A FIB tool is nearly identical to an SEM with one major exception. As can be in seen in figure 3.1, a liquid metal ion source, usually Ga+ liquid metal, is attached to an SEM at a 52° angle to the electron beam column [9, 11, 20]. While imaging the sample with the electron beam, the ion beam can be used to sputter away a user-defined region on the sample surface. Using successively smaller ion beam currents and spot sizes, material may be milled away in a finer, more precise manner, analogous to traditional polishing, until electron transparency is reached. When imaging with the ion source, 10-100 nm resolution may be achieved depending on the beam current and apertures used [20]. FIB tools are also equipped with Pt and/or W gas injection sources (GIS), allowing the membrane to be welded in situ to the sample grid with the ion beam.

If a site-specific TEM sample is desired, the FIB membrane may be plucked and

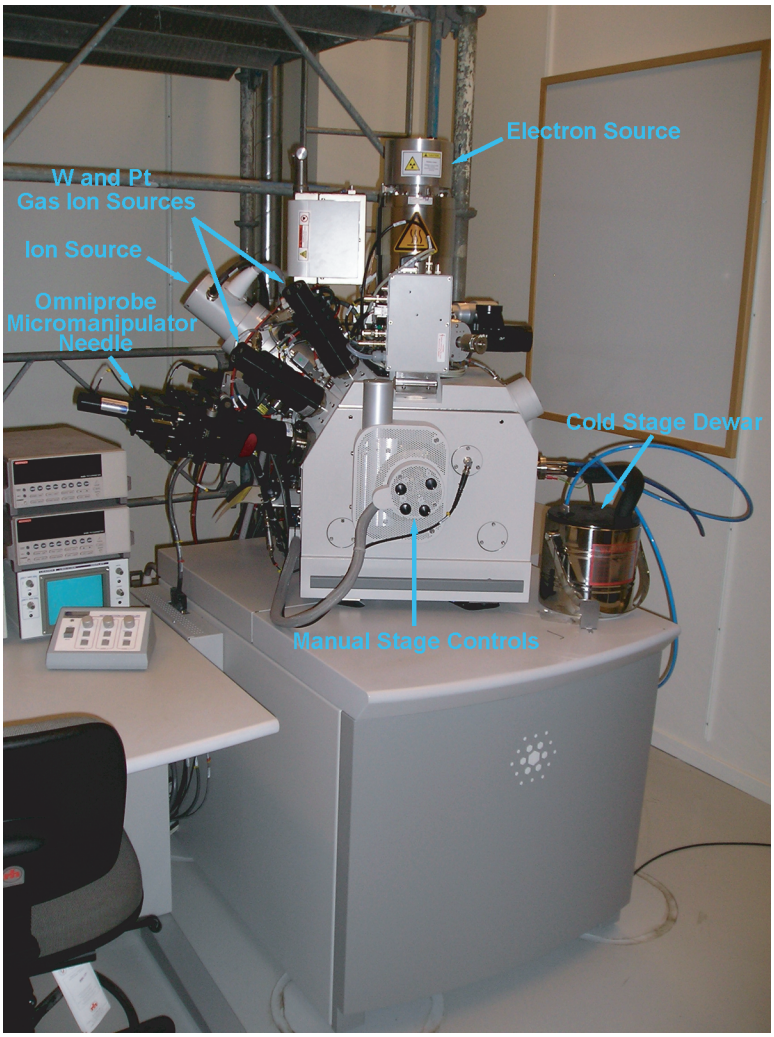


FIG. 3.1. *A common FIB tool set-up. (FEI Strata 235 DualBeam).*

placed on a sample grid using in situ or ex situ micromanipulators. This procedure is termed the lift-out technique. The self-supported sample geometry is also common for TEM samples; the material is polished traditionally and the FIB tool mills trenches [9, 20, 28]. The sample is attached to a TEM washer and has an H-shaped surface after milling.

Several other advantages and disadvantages exist. Surface damage due to Ga+ implantation may be prevented by depositing straps of Pt, W or SiO2 using the gas injection sources [9, 20]. A protective metal layer also prevents the curtain effect where striations form due to changes in sputter rate between different materials. However, when comparing FIB to broad ion beam (BIB) milling, preferential milling is reduced [20]. The lift-out method may have a very low-yield due to the precision needed when using micromanipulators.

The FIB is not just a sample preparation tool. Often times, EBSP and EDS detectors will be attached to the column of the FIB [23]. As milling proceeds, orien-

tation and composition maps may be taken. Stringing the data together may create a 3D image of the sample. Needle samples used for APT and TEM membranes may be easily made using automated routines [21, 22].

4. Experimental. Polycrystalline copper disks, (purity 99.9995%), approximately 9 mm in diameter and 0.8 mm thick were used for grain boundary segregation studies of Au and Bi dopants. The disks were encapsulated inside silica tubes under Ar atmosphere, heat treated in a Lindberg/Blue M furnace for 1 hr at 400°C and air quenched. Annealing promoted grain boundary stabilization prior to dopant diffusion and deposition. Disks were polished to a 0.5 μm finish using diamond lapping film followed by a silica slurry polish. Grain size was measured using electron backscatter diffraction (EBSD) and found to be an average of 8 μm before the initial anneal, as shown in figure 4.1.

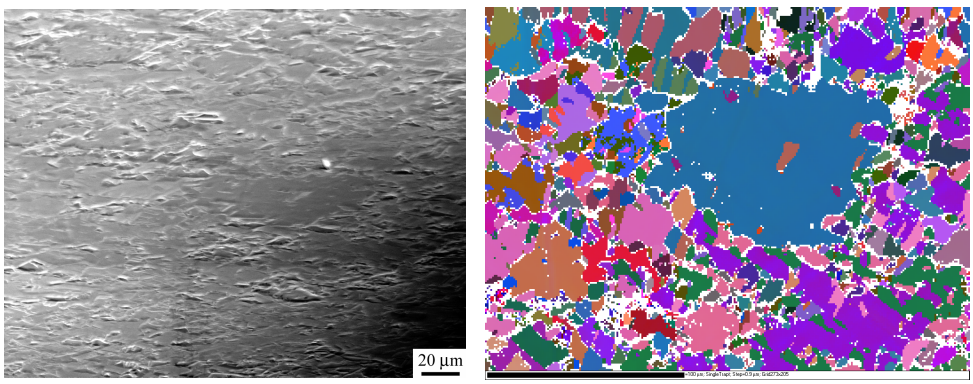


FIG. 4.1. *EBSD analysis of a pure Cu surface prior to annealing or deposition.*

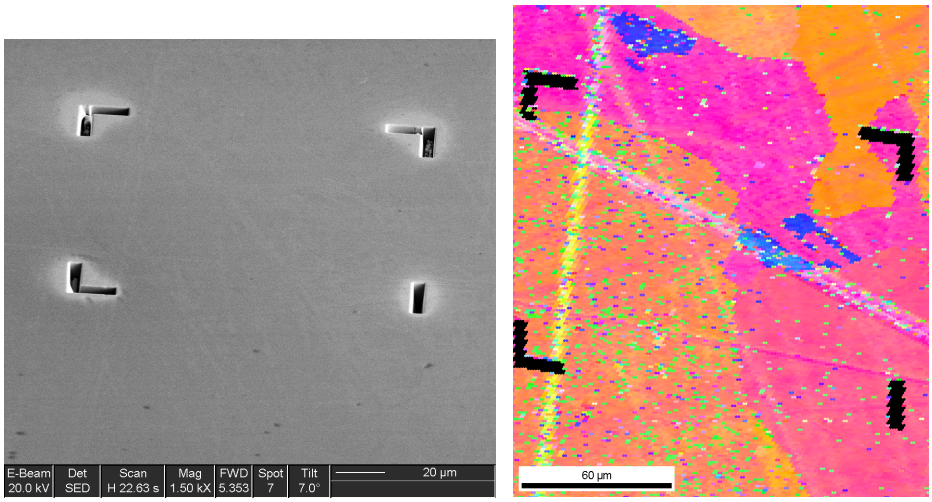
Au or Bi thin films were deposited via physical vapor deposition (PVD). The initial Cu surface was sputtered with N_2 for 30 s to remove any oxides. Bi films were capped with a thin film of Au to prevent surface oxidation. Substrate heating reached $\sim$50°C during deposition; the PVD was not equipped with a cold stage to prevent diffusion during deposition. Substrates were cleaned with an acetone/methanol rinse prior to deposition.

Disks were, again, encapsulated, annealed at 250°C and air quenched. Annealing times and temperatures were calculated using Bi and Au diffusion coefficients calculated by Gorbachev, et al. [12] Samples were kept in a continuously N_2-flushed dessicator when not in use to minimize oxidation.

Atom probe tips were made from the substrate disks using the FIB tool. An FEI Strata 235 DualBeam and a Quanta 200 3D FIB tool were used to mill thin membranes in the bulk. These membranes were plucked from the bulk via the lift-out technique and affixed to fabricated Si micro-tips using the Omniprobe needle and Pt deposition features of the microscopes. Prior to milling the main trenches, markers were milled in the surface to enable the same region to be found later. EBSD was used to identify grain boundaries and triple junctions at the surface. A Pt strap was deposited over the features of interest for identification and protection from Ga ion beam damage. After the membrane was diced and welded to the micro-tips, a series of annular millings were performed until the tip was approximately 100 nm in diameter. A Pt cap remained at the tip to protect the surface from oxidation and beam damage.

Atom-probe tomography was performed to analyze the tip contents. Tips were evaporated at 60 K using a pulse fraction of 15% and pulse rate of 200,000 Hz in an Imago LEAP 3000 APT. Using the micro-array geometry, several tips from the same grain boundary may be analyzed in series.

5. APT Tip Preparation. Samples are initially mounted to a 45° angle holder. Markers in the substrate are milled using 3000 pA current at 7° tilt. (Sample surface is positioned at 52° to the electron column.) Samples are moved to 70° with respect to the electron beam; EBSD maps produced using TSL or HKL detectors and aid in locating triple junctions. A representative map is shown in figure 5.1. The surface is



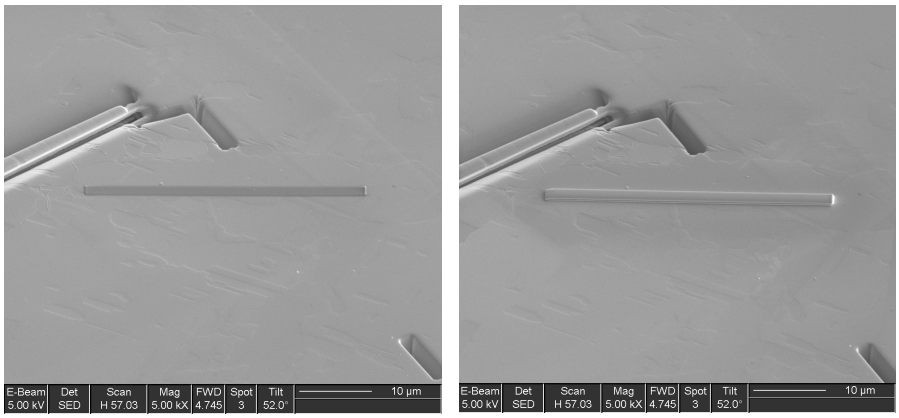
(a) (b)

FIG. 5.1. *(a) SEM image at 70° tilt of region of interest, (b) EBSD map of same region.*

tilted back to 52° and a protective Pt strip is deposited over the triple junctions and grain boundaries of interest. The electron beam is used to deposit ∼50 nm Pt before the ion beam deposits ∼400 nm Pt, and can be seen in figure 5.2. Use of the electron beam before the ion beam allows the sample to be protected from ion-beam damage. Substrates are remounted to a flat holder with additional spaces for electropolished tips and micro-arrays, which preserve samples when using the Omniprobe needle by avoiding chamber venting. The region of interest can be found again using the markers milled for EBSD.

Two milling recipes have been used for membrane preparation. Both require the stage to be tilted 30° to the ion beam (22° with respect to the electron beam), and have the same procedure. The first recipe called for large trenches to be milled next to the area of interest, while thin slices were milled in the second recipe. The second recipe proved to be much more time-efficient and subjected the sample to less Ga+ bombardment. Therefore, only the second, more refined, recipe will be described here.

After a grain boundary is chosen and Pt deposition is done in situ, the sample is tilted 22° and small slits are milled on either side of the Pt, shown in figure 5.3(a). The sample membrane is in the shape of an isosceles triangle, where the bottom has been cut free and is attached at the sides. One side is milled free from the bulk, forming a cantilever, as in figure 5.3(b). The Omniprobe micromanipulator is brought into the chamber such that it just touches the sample, shown in figure 5.3(c).



(a) (b)

Fig. 5.2. *(a) Electron-beam deposited Pt, (b) ion-beam deposited Pt.*

The micromanipulator is welded to the sample surface using the Pt deposition, the membrane is cut free from the bulk and the membrane is lifted from the surface. Figure 5.3(d) shows a membrane free from the bulk and welded to the Omniprobe.

The membrane may be diced into smaller sections and affixed to a Si coupon with micro-tips prepared via lithography. Figure 5.4(a) shows how the end of the membrane is brought to just touch a micro-tip. Pt welds are used to hold the sample to the micro-tip without cutting the sample free from the Omniprobe. The section is milled away from the rest of the membrane and more tips can be made from the section until the membrane is completely used. The welded sections are sharpened into tips using a sequence of annular milling. Smaller annuli and beam currents are used until the final tip, shown in figure 5.4(c), is less than 100 nm in diameter. The tips can then be taken to the APT and analyzed. Preliminary data is shown in figure 5.4(d).

6. Conclusions. While the goal of measuring dopant concentration levels at triple junctions and comparing to levels at adjacent grain boundaries has yet to be reached, a solid start has been made. The sample preparation method has been fine-tuned and can now be used to confirm diffusion coefficient measurements and segregation levels in the future. Stresses in the sample needle resulting from the high electric field used in APT can cause embrittled Cu to fracture during analysis before sufficient data is obtained [34]. A method to prevent fracture needs to be looked at.

In the sample preparation experiments, the Pt deposited in the FIB was found not evaporate in the APT. A strong C signal was collected, but no Pt. Specimen fracture occurred very early during analysis due to the high strains on the sample by trying to evaporate the Pt. Work is being conducted to determine how to sufficiently protect the surface from ion bombardment.

The effect of Ga+ ion bombardment is a topic of future research. The same effects on Cu grain boundaries need to be looked at with Ga as the segregant. Ga is soluble in Cu and for the same 3-dimensional considerations, the grain boundary segregation behavior needs to be investigated [10, 18]. Little is known about the influence of Ga introduced by the FIB on Cu grain boundaries and may be investigated using the method outlined here. According to Sigle, et al. ion-beam thinning introduces dislocation loops, which disrupts the diffusion process.

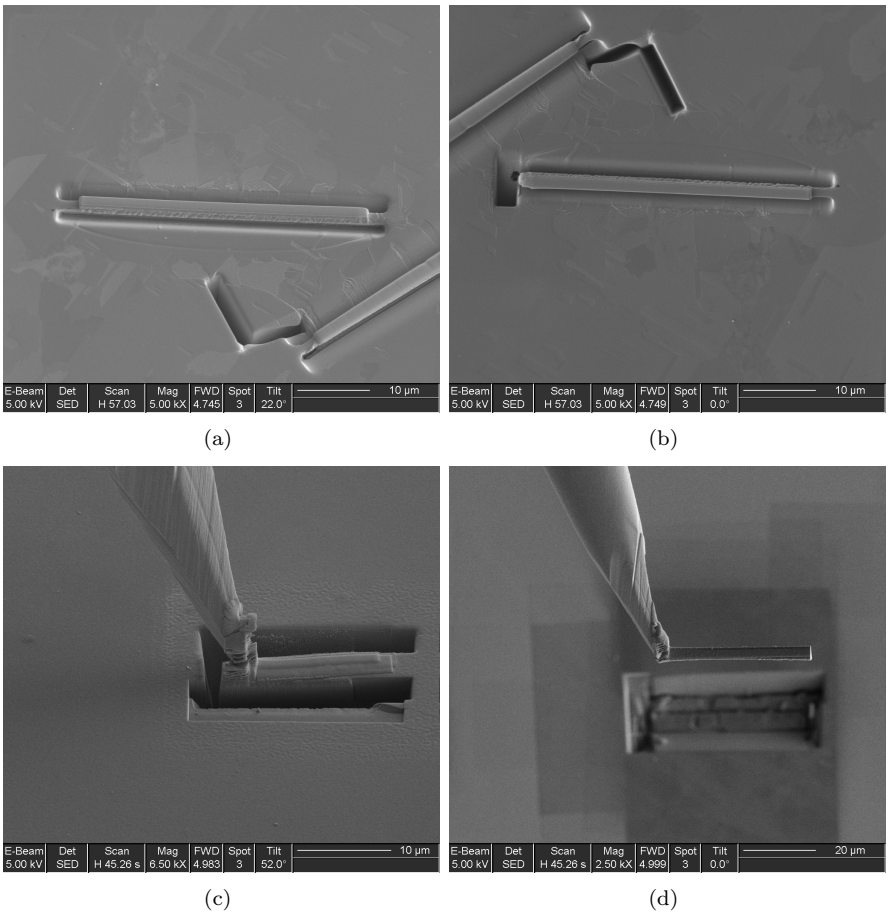


FIG. 5.3. *(a) Trench milling, (b) side cut creating cantilever, (c) Omniprobe needle touching membrane, and (d) membrane lifted from bulk on Omniprobe needle.*

7. Acknowledgements. The authors acknowledge support of the National Center for Electron Microscopy, Lawrence Berkeley Lab, which is supported by the U.S. Department of Energy under Contract #DE-AC02-05CH11231. Drs. Andrew Minor and Velimir Radmilovic assisted in FIB operation.

Prof. C. Barry Carter contributed as the graduate student advisor at the University of Minnesota. Dr. Stuart McKernan, at the University of Minnesota, assisted with the EBSD analysis. Douglas Medlin, Joseph Michael, Mark Homer, Nancy Yang, Andy Gardea and Jeffrey Chames from Sandia National Laboratories provided advice and equipment access. Assistance and FIB access was given by Dr. Gregory Thompson and Chandan Srivastava at the University of Alabama in Tuscaloosa.

REFERENCES

[1] U. ALBER, H. MÜLLEJANS, AND M. RÜHLE, *Bismuth segregation at copper grain boundaries*, Acta Mater., 47 (1999), pp. 4047–4060.

[2] F. ALTMANN AND D. KATZER, *Cross-sectional specimen preparation from ics downside for sem and tem-failure analyses using focused ion beam etching*, Thin Solid Films, 343-344 (1999),

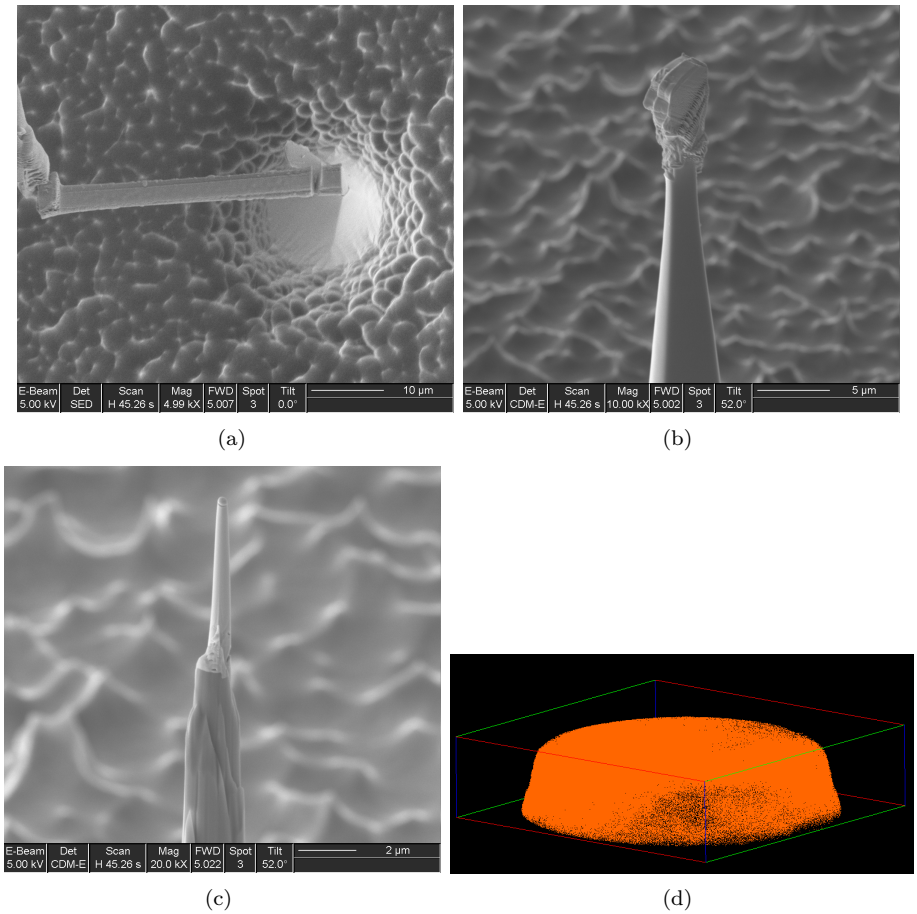


FIG. 5.4. *(a) Membranes are welded to micro-tips and diced into sections, (b) typical membrane section before annular milling, (c) a finished APT tip, and (d) preliminary APT data.*

pp. 609–611.

[3] J.M. CAIRNEY AND P.R. MUNROE, *Preparation of transmission electron microscope specimens from feal and wc powders using focused-ion beam milling*, Materials Characterization, 46 (2001), pp. 297–304.

[4] L.-S. CHANG, E. RABKIN, B.B. STRAUMAL, B. BARETZKY, AND W. GUST, *Thermodynamic aspects of the grain boundary segregation in cu(bi) alloys*, Acta Mater, 47 (1999), pp. 4041–4046.

[5] L.-S. CHANG, E. RABKIN, B. STRAUMAL, P. LEJČEK, S. HOFMANN, AND W. GUST, *Temperature dependence of the grain boundary segregation of bi in cu polycrystals*, Scr. Mater., 37 (1997), pp. 79–735.

[6] S. DIVINSKI, M. LOHMANN, AND C. HERZIG, *Grain boundary diffusion and segregation of bi in cu: radiotracer measurements in b and c diffusion regimes*, Acta Mater., 52 (2004), pp. 3973–3982.

[7] ———, *Grain-boundary melting phase transition in the cu-bi system*, Phys. Rev. B, 71 (2005), p. 104104.

[8] H.C. EATON AND R. J. BAYUSICK, *Observation of grain-boundary migration using field ion microscopy*, Appl. Phys. Lett., 32 (1978), pp. 115–117.

[9] *et al.* F.A. STEVIE, *Application of focused ion beam lift-out specimen preparation to tem, sem, stem, aes and sims analysis*, Surf. Interface Anal., 31 (2001), pp. 345–351.

[10] V.V. GAL AND P.L. GRUZIN, *A study of the spread and surface diffusion of mercury and gallium on the surfaces of copper single crystals*, Fiz. Metal. Metalloved, 30 (1970), pp. 796–799.

[11] J. Goldstein, D. Newbury, D. Joy, C. Lyman, P. Echlin, E. Lifshin, L. Sawyer, and J. Michael, *Scanning Electron Microscopy and X-ray Microanalysis*, Kluwer Academic/Plenum Publishers, New York, 2003.

[12] V.A. Gorbachev, S.M. Klotsman, Y.A. Rabovskii, V.K. Talinskiy, and A. N. Timofeev, *Impurity diffusion in copper: Diffusion of gold, lead and bismuth in copper*, Phys. Met. Metall., 44 (1977), pp. 214–217.

[13] H.J. H·ofler, R.S. Averback, H. Hahn, and H. Gleiter, *Diffusion of bismuth and gold in nanocrystalline copper*, J. Appl. Phys., 74 (1993), pp. 3832–3839.

[14] V.A. Ivanov, A.S. Ostrovsky, A.L. Peteline, and S. A. Peteline, *Exact solution of triple junction diffusion problem*, Def. Dif. Forum, 156 (1998), pp. 223–228.

[15] H. H. Kart, M. Tomak, and T. Çağin, *Thermal and mechanical properties of cu-au intermetallic alloys*, Modelling Simul. Mater. Sci. Eng., 13 (2005), pp. 657–669.

[16] T. Kato, Y. Sasaki, K. Osada, T. Hirayama, and H. Saka, *Transmission electron microscopy studies of microstructures of silica-zirconia membranes for gas separation*, Surf. Interface Anal., 31 (2001), pp. 409–414.

[17] E. Keehan, *Effect of Microstructure on Mechanical Properties of High Strength Steel Weld Metals*, PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, 2004.

[18] S.M. Klotsman, Y.A. Rabovskii, V.K. Talinskiy, and A. N. Timofeev, *Volume diffusion of gallium-67 and germanium-68 in copper*, Fiz. Metal. Metalloved., 31 (1971), pp. 429–431.

[19] B.W. Krakauer and D.N. Seidman, *Systematic procedures for atom-probe field-ion microscopy studies of grain boundary segregation*, Rev. Sci. Instrum., 63 (1992), pp. 4071–4079.

[20] R.M. Langford and A. K. Petford-Long, *Preparation of transmission electron microscopy cross-section specimens using focused ion beam milling*, J. Vac. Sci. Technol. A, 19 (2001), pp. 2186–2193.

[21] D.J. Larson, D.T. Foord, A.K. Petford-Long, T.C. Anthony, I.M. Rozdilsky, A. Cerezo, and G.W.D. Smith, *Focused ion-beam milling for field-ion specimen preparation: preliminary investigations*, Ultramicroscopy, 75 (1998), pp. 147–159.

[22] D.J. Larson, D.T. Foord, A. K. Petford-Long, H. Liew, M.G. Blamire, A. Cerezo, and G.W.D. Smith, *Field-ion specimen preparation using focused ion-beam milling*, Ultramicroscopy, 79 (1999), pp. 287–293.

[23] D.M. Longo, J.M. Howe, and W.C. Johnson, *Development of a focused ion beam (fib) technique to minimize x-ray fluorescence during energy dispersive x-ray spectroscopy (eds) of fib specimens in the transmission electron microscope (tem)*, Ultramicroscopy, 80 (1999), pp. 69–84.

[24] M. K. Miller, *The development of atom probe field-ion microscopy*, Mater. Charact., 44 (2000), pp. 11–27.

[25] L.E. Murr, *Electron and Ion Microscopy and Microanalysis*, Marcel Dekker, Inc., New York, 1982.

[26] R. Ouchida, T. Shiraishi, M. Nakagawa, and M. Ohta, *Effects of au/cu ration and gallium content on the low-temperature age-hardening in au-cu-ga alloys*, J. Mats. Sci., 30 (1995), pp. 3863–3866.

[27] J.A. Panitz, *The archetypical atom-probe*, Mater. Charact., 44 (2000), pp. 3–10.

[28] C.R. Perrey, C.B. Carter, J.R. Michael, P.G. Kotula, E.A. Stach, and V. R. Radmilovic, *Using the fib to characterize nanoparticle materials*, J. Micros., 214 (2004), pp. 222–236.

[29] S.I. Prokofjev, *Diffusion of au along ¡100¿ symmetrical tilt boundaries in copper: Grain-boundary roughening?*, Def. Dif. Forum, 194-199 (2001), pp. 1337–1342.

[30] ———, *Effect of diffusant segregation on the misorientation dependences of the characteristics of grain-boundary diffusion: Ni and au in copper*, Def. Dif. Forum, 194-199 (2001), pp. 1141–1146.

[31] W. Sigle, L.-S. Chang, and W. Gust, *On the correlation between grain-boundary segregation, faceting and embrittlement in bi-doped cu*, Phil. Mag. A, 82 (2002), pp. 1595–1608.

[32] T. Surholt and C. Herzig, *Grain boundary self- and solute diffusion and segregation in general large angle grain boundaries in copper*, Def. Dif. Forum, 143-147 (1997), pp. 1391–1396.

[33] M. Tanaka, K. Furuya, and T. Saito, *Tem observation of structural differences between two types of ni silicide/si thin films caused by fib irradiation*, Thin Solid Films, 319 (1998), pp. 101–105.

[34] M. Thuvander and H.O. Andrén, *Apfim studies of grain and phase boundaries: A review*, Mater. Charact., 44 (2000), pp. 87–100.

[35] K. Tsujimoto, S. Tsuji, K. Kuroda, and H. Saka, *Transmission electron microscopy of*

thin-film transistors on glass substrates, Thin Solid Films, 319 (1998), pp. 106–109.

[36] S. Wannaparhun, S. Scal, K. Scammon, V. Desai, and Z. Rahman, *Physics of insulating specimen preparation for non-charging auger electron spectroscopy*, J. Phys. D, 34 (2001), pp. 3319–3326.

[37] D.B. Williams and C.B. Carter, *Transmission Electron Microscopy: A Textbook for Materials Science*, Plenum, New York, 1996.

MODELING POLYCRYSTALLINE MECHANICS VIA THE EXTENDED FINITE ELEMENT METHOD

JESSICA SANDERS*, THOMAS VOTH†, AND JOSHUA ROBBINS†

Abstract. There are many challenges associated with finite element modeling of polycrystalline structures at length-scales on order of grain-size. For example, at the micro-scale material behavior is modified relative to the macro-scale response by a (generally random) distribution of "grains" and the associated grain interfaces. Hence, assumptions about bulk material behavior may no longer be appropriate. In simulations where modeling at the micro-scale is of interest (e.g. micro-machining applications [8]) capturing this behavior by explicit meshing of grain boundary interfaces can be inefficient and impractical. In either case, models for sub-grid mechanics (e.g. grain boundary sliding) are needed [6, 11].

The eXtended Finite Element Method (X-FEM; cf. [2, 9]) and the Generalized Finite Element Method (GFEM; cf. [4]) provide a variational framework to represent boundaries and interfaces without fitting a conforming mesh. Instead, internal features are represented with discontinuous enrichment of the standard finite element basis. An extension of the method by Simone et al. [11], specifically applies these ideas to the problem of polycrystalline materials.

In this paper an implementation of the generalized finite element method is demonstrated. The technique, based on that of [11], is designed to capture polycrystalline grain behavior independent of the background mesh. Some results are presented which demonstrate the efficacy of the method in terms of i) accuracy (patch test and convergence) and ii) independence of discontinuity location (grain boundary) on background mesh topology. Finally, new techniques for treating inter-granular physics in the context of GFEM/X-FEM are briefly discussed.

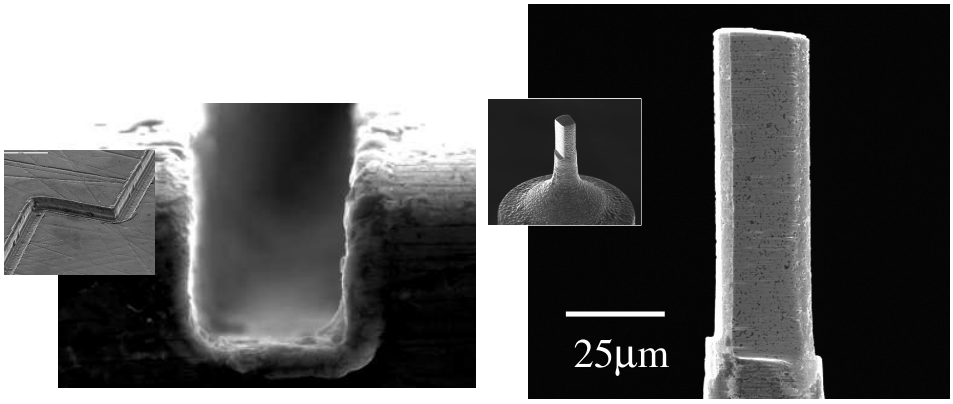
1. Introduction. Material removal (machining) processes at the macro-scale represent a substantial portion of the metal shaping industry, accounting for approximately \$5Bn annual investment world-wide. Indeed, nearly all metal forming processes involve some machining component as it allows significantly increased i) tolerances and ii) freedom in final part shape [1].

Significant sophistication and detailed understanding in macro-machining processes and their underlying physics have been realized over the last several decades. Unfortunately, this effort has not translated to the micro/meso-scale process [8]. Manufacturing of micro/meso-scale parts (e.g. micro-machines and MEMs packaging) has generally relied on additive/subtractive lithography and etching techniques developed for the electronics industry. Although valuable, these techniques are limited in part geometry that may be manufactured (they are inherently planar $2\frac{1}{2}$-D) and materials that may be considered [7]. Micro-machining, like macro-machining, is theoretically unlimited in terms of the geometries and materials to which it may be applied. Hence, the technology promises increased freedom in terms micro-machine design.

Although machining of small parts offers significant advantages, the process is quite different from the macro-scale process. One substantial difference is that process length-scales (e.g. tool tip radius and depth-of-cut) are on the order of material grain-size. Hence, to appropriately model the cutting process it is necessary to model the discrete structure of the grains. One approach is to model the grains as a continuum (with a finite element approach, for example) and to capture the inter-granular effects via some sub-grid model. However, in simulations where modeling at the micro-scale is of interest (e.g. micro-machining applications [8]) capturing this behavior by explicit meshing of grain boundary interfaces can be inefficient and impractical.

The eXtended Finite Element Method (X-FEM; cf. [2, 9]) and the Generalized

*Duke University
†Sandia National Laboratories



(a) (b)

FIG. 1.1. *Images of (a) a machined part and (b) a micro-machining tool (end-mill). Courtesy of D. Gill, Org. 2450*

Finite Element Method (GFEM; cf. [4]) provide a variational framework to represent boundaries and interfaces without fitting a conforming mesh. Instead, internal features are represented with discontinuous enrichment of the standard finite element basis. An extension of the method by Simone et al. [11], specifically applies these ideas to the problem of polycrystalline materials.

The goal of this paper is to present some first steps towards modeling of the complex micro/meso-scale machining process. Specifically, we describe the theory and implementation of a generalized finite element approach for polycrystalline materials characterized by homogeneous grains. The sub-grid mechanics associated with the grain/grain interfaces are modeled via an applied traction condition. As noted earlier, an important aspect of the GFEM/X-FEM introduced here is that grain boundaries need not align with the underlying mesh boundaries - a significant advantage in terms of, for example, stochastic analysis of the impact of crystal topology on large-scale mechanical response [3]. We note that this approach provides a framework though which complex granular interface models (obtained via experiment or molecular dynamics simulations, for example) may be incorporated.

In the following sections, we begin with a description of our problem followed by a derivation of the generalized finite element method as implemented here. We then demonstrate the efficacy of the method in terms of i) accuracy (patch test and convergence) and ii) independence of discontinuity location (grain boundary) on background mesh topology. Finally, new techniques for treating inter-granular physics in the context of GFEM/X-FEM are briefly discussed.

2. Problem Formulation. A description of our model problem and its discretization follows. Unless otherwise noted, the GFEM discretization presented here follows that outlined in [11].

2.1. Problem domain. We consider a body, Ω, in a two-dimensional euclidean space. The body is comprised of N discrete and non-overlapping sub-domains, or grains, as shown in Figure 2.1. The boundary of the body is Γ and is divided into two non-overlapping regions, Γ_d and Γ_n, over which displacement and traction boundary conditions are applied respectively. The set of grain boundaries not included in Γ is shown as φ. Mathematically, the body is defined as the union of all N grains,

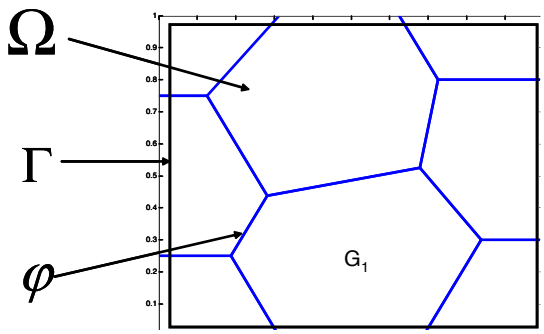


FIG. 2.1. *Illustration of model polycrystalline domain.*

$$\Omega = \bigcup_{n=1}^{N} G_n \tag{2.1}$$

2.2. Model equations. We are interested in finding a displacement field over the body which satisfies the time-independent model equations including force equilibrium over the whole body,

$$\sigma_{ij,j} + f_i = 0 \text{ in } \Omega \tag{2.2}$$

displacement boundary conditions on the boundary Γ_d,

$$u_i = g_i \text{ in } \Gamma_d \tag{2.3}$$

tractions over the boundary Γ_n,

$$\sigma_{ij} n_j = h_i \text{ in } \Gamma_n \tag{2.4}$$

and a material constitutive relationship

$$\sigma_{ij} = C_{ijkl} u_{(k,l)} \tag{2.5}$$

where strains are assumed to be small.

Additionally, some constraint must be applied at the grain boundary interfaces, the choice of which will affect the final form of the variational equations.

2.3. Variational form. Thus far, all of the derivation, including the model equations, parallels standard linear finite element problem formulation. The first point of departure for X-FEM is in the definition of the solution space. In our case, we are interested in a solutions in terms of displacements over Ω. To derive the variational form of the equations on which the finite element solution will be based, it is necessary to define a suitable form of the displacement solution, $u(x)$. Whereas traditional finite element formulation assumes a C^0 continuous space for the solution, the X-FEM formulation allows the displacement field to be discontinuous over the body. The displacement field in this case takes the form,

$$u(x) = \hat{u}(x) + \sum_{n=1}^{N} H_n(x) \tilde{u}_i(x) \tag{2.6}$$

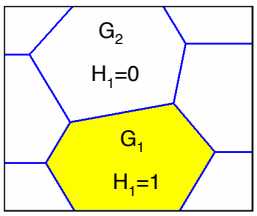


FIG. 2.2. *Schematic view of Heaviside definition.*

In Equation 2.6, H is a Heaviside function defined by the geometry of the grain structure and takes the form,

$$H(x) = \begin{cases} 1 \text{ if } x \in G_n \\ 0 \text{ if } x \notin G_n \end{cases} \tag{2.7}$$

as shown schematically in Figure 2.2.

Using the approximation for displacement in Equation 2.6, we multiply Equation 2.2 by an appropriate weighting function of the same form, integrate, and apply the appropriate boundary conditions. The result is the variational form of the equations. Note that there are multiple variational equations, one for the entire body (coarse-scale; Equation 2.8) and an additional equation for every grain (fine-scale: Equation 2.9).

$$\int_{\Omega} \hat{w}_{(i,j)} \sigma_{ij} d\Omega = \int_{\Omega} \hat{w}_i f_i d\Omega + \int_{\Gamma_n} \hat{w}_i t_i d\Gamma_n \tag{2.8}$$

$$\int_{G_n} \tilde{w}_{(i,j)_n} \sigma_{ij} dG_n + \text{ grain boundary terms } = \int_{G_n} \tilde{w}_{i_n} f_i dG_n + \int_{\Gamma_{G_n}} \tilde{w}_{i_n} t_i d\Gamma_{G_n} \tag{2.9}$$

2.4. Enriched FEM Approach. To solve the coupled variational problems, we apply a finite element discretization to the domain. With the X-FEM approach, it is not necessary to fit a finite element mesh to the grain boundaries, φ. The background can be any uniform or unstructured mesh with elements appropriate to the kind of material deformation that is expected. Elements that are cut by a grain boundary must support a discontinuous displacement approximation in the same way that it is defined over the domain. Consequently, the X-FEM approximation to the displacement field takes the same discontinuous form as the overall displacement, with the addition of the standard finite element shape functions, $\phi(x)$.

$$u_h(x) = \phi_A(x)\hat{u}_A + \sum_{n=1}^{N} H_n(x)\phi_A(x)\tilde{u}_{A_n} \tag{2.10}$$

An example of the displacement field over an element that crosses a grain boundary interface between grains 1 and 2 (see Fig. 2.2) might be,

$$u_h(x) = \phi_A(x)\hat{u}_A + H_1(x)\phi_A(x)\tilde{u}_{A_1} + H_2(x)\phi_A(x)\tilde{u}_{A_2} \tag{2.11}$$

for $A = 1...n_{en}$ where n_{en} is the number of nodes in the element.

However, there is an immediate problem in that the set of H is linearly dependent over the element. Specifically, we have that,

$$H_2 = 1 - H_1 \tag{2.12}$$

For this reason, only one Heaviside "enrichment" need (or should) be implemented over that node,

$$u_h(x) = \phi_A(x)\hat{u}_A + H_1(x)\phi_A(x)\tilde{u}_{A_1} \tag{2.13}$$

The use of H_2 is an equally valid choice. Similarly, over elements where there are no discontinuities present, the traditional finite element approximations are sufficient.

Nodes on elements which are cut by boundaries receive extra degrees of freedom, i.e., $\tilde{u}_{A_1}$ in Equation 2.13, as a result of the discontinuous approximation and are referred to as "enriched". The integration of an element that has been cut by a grain boundary has to be approached differently than the integration of a standard finite element. To resolve the displacement field appropriately, a cut element must be decomposed into integrable regions on either side of the discontinuity. A suitable Gaussian quadrature can then be applied to the element subregions.

2.5. Grain interface model. After enrichment, cut finite elements have additional degrees of freedom at the enriched nodes which are allowed to move independently. This is the mechanism that allows discontinuous displacement fields across boundaries, but it also means that if no constraints are applied at grain boundaries, the grains are free to separate from each other. There is on-going research into the mechanical nature of grain interactions at the boundaries. In the finite element context, a model must be chosen to represent the interactions.

Here, we have implemented two relatively simple models for grain boundary interactions. The first is a perfect tie between the grains. The second is a perfect tie normal to grain boundaries and no constraint in the tangential direction. The second choice is a rough model of minimally viscous grain boundary sliding. For this implementation, a penalty method enforces the constraint in the affected elements. Additional terms are added to the stiffness of the granular variational equations (equation 2.9).

In the case of perfect tying, the term that is added to the variational equation in place of the previous term only referred to as "grain boundary terms" is

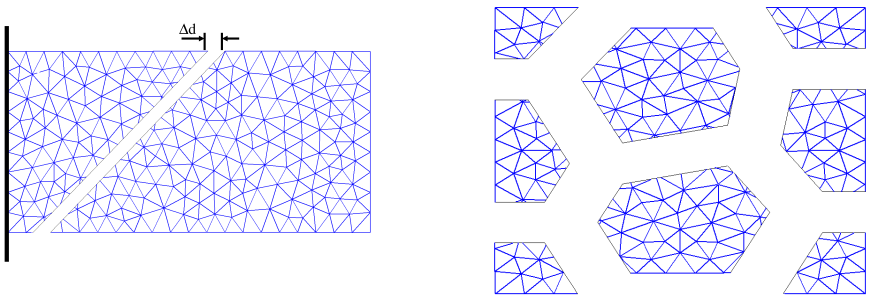
$$\varepsilon \int_{\varphi} [[\tilde{w}]] d\varphi \int_{\varphi} [[\tilde{w}]] d\varphi \tag{2.14}$$

In the case of tying only in the normal direction, the term is very similar, with the inclusion of the dot product with a grain normal.

$$\varepsilon \int_{\varphi} [[\tilde{w}]] \cdot \vec{n} d\varphi \int_{\varphi} [[\tilde{w}]] \cdot \vec{n} d\varphi \tag{2.15}$$

In both cases, the displacement jump in an element across a grain boundary is the term being "penalized" to be zero, and it is defined as,

$$\int_s [[\tilde{u}]] ds = \int_s \tilde{u}^+ ds - \int_s \tilde{u}^- ds \tag{2.16}$$



(a) (b)

FIG. 3.1. *Results to stress free problem.*

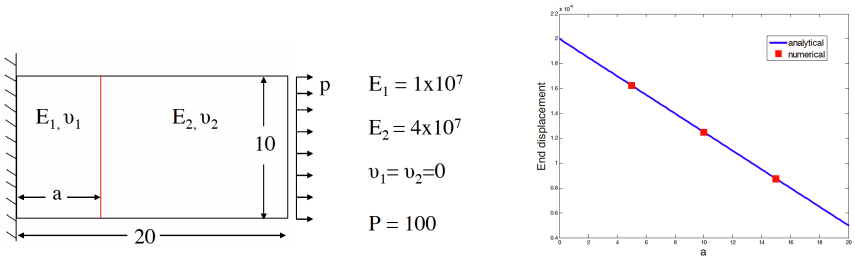
In addition to traditional issues of convergence and ill-conditioning, it has been shown ([10]) that the use of penalty method to enforce intra-element interface tractions (in the context of Partition Of Unity; POU methods) may result in artificial oscillations in these tractions. Simone notes that this issue, which is similar in nature to that for interface elements, is not well understood [10]. More distressing is that, while these oscillations may often be controlled for interface elements via an appropriate choice of interface integration scheme, such an approach is only effective for special mesh/interface configurations when applied in the context of POU/X-FEM methods. Indeed, Simone showed that oscillations will always be present when a penalty is employed and linear triangle elements are used [10]. For these reasons, more sophisticated enforcement methodologies are desirable and are currently under investigation.

3. Numerical Examples. The preceding method was implemented in a 2-dimensional eXtended Finite Element Code in the MATLAB framework. Only small strains and linear materials were considered. Linear triangle finite elements were used as the background mesh, and grain structure was produced through a process of Voronoi tessellation. The open source mesher GMSH [5] was used to produce the mesh. A number of verification problems were developed to test the capability of the code and algorithm.

The first verification problems were run with no grain boundary interfaces, and rigid body displacements applied to grains so that they separated. The ability to freely separate is demonstrated by Figure 3.1. All results are completely stress free, as should be expected.

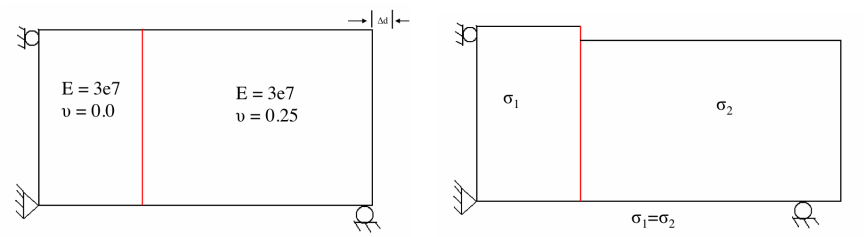
The first test problem is essentially a patch test in which a linear displacement field is applied to a rectangular body. The body is comprised of two grains, as shown in Figure 3.2, separated by a vertical interface constrained by a perfect tie. A constant pressure on one end and a fixed displacement condition on the other produce a linear displacement field. The end displacement of the body can be analytically deduced as a function of the position of the grain interface in the body and the properties of each grain. Several numerical simulations were run with different interface locations. Figure 3.2 also demonstrates the comparison between the analytical solution and the numerical simulations. All numerical simulations recovered the analytical solution to machine precision.

The second problem is a test of the grain boundary sliding algorithm. It also



(a) (b)

FIG. 3.2. *Patch test on a fully tied interface showing (a) problem domain and (b) predicted and analytical solutions.*



(a) (b)

FIG. 3.3. *Patch test on a fully tied interface*

involves a rectangular body with two grains, but in this case one grain has a 0 Poisson's ratio and the second has a Poisson's ratio of 0.25 (Figure 3.3). With the appropriate constraints, an end displacement was applied to the second grain. There is also an analytical solution for the problem. The second grain should contract in the vertical direction while the first does not. No tangential tractions should be created at the boundary, and the stress should be constant in both elements. Again, the numerical solution reproduces the solution exactly (Figure 3.3).

Both of the first two test problems considered only linear displacement fields. With a high enough penalty number, the correct answer can be recovered to machine precision regardless of mesh density.

Next, A problem without a linear displacement field was designed in order to test convergence of the method with mesh density. With a penalty number scaled as $O(1/h^2)$ a convergence rate approximately equal to 2 was recovered (Figure 3.4). However, such scaling of the penalty number can easily destroy matrix conditioning, which provides another incentive to move to more robust numerical techniques for grain boundary constraints.

Finally, to demonstrate the technique on a more complicated problem, a simulation was run with a more complicated grain structure and grain boundary sliding (Figure 3.5). Though there is no analytical solution to the problem, certain expected features emerged, such as a lack of tangential traction along grain boundaries and a reasonable displacement field.

4. Conclusions. The purpose of this work was to begin addressing issues associated with finite element modeling of polycrystalline materials at length-scales

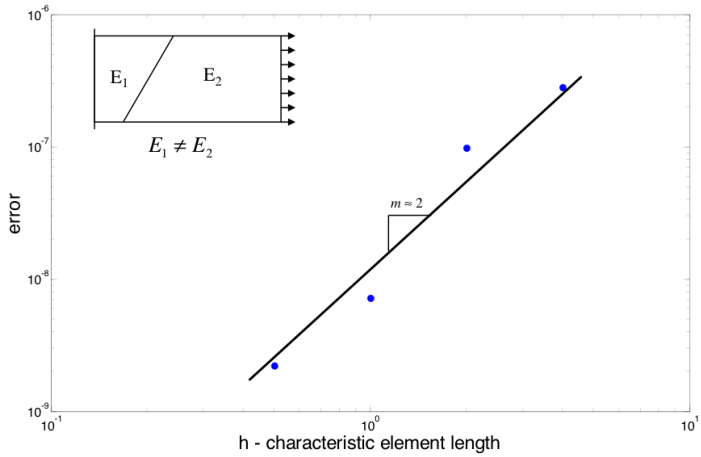


FIG. 3.4. *Convergence of tip-displacement for a multi-material beam with Heaviside enrichment and fully tied interface.*

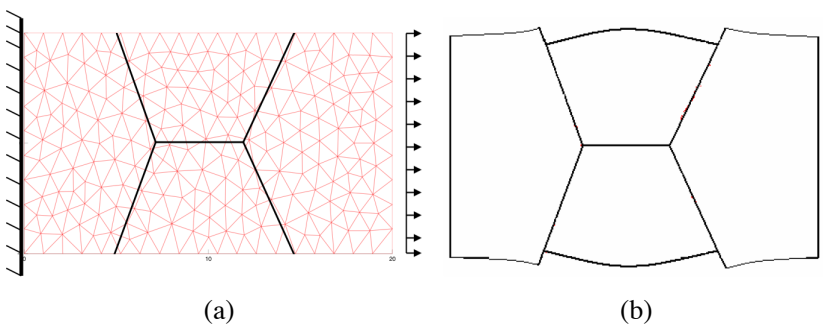


FIG. 3.5. *Multiple grain (a) geometry, mesh and applied loadings, and (b) (exaggerated) displacements for normally-tied, tangentially free interface conditions.*

on order of grain-size. We have investigated a technique for representing the polycrystalline grain structure on a background mesh using the eXtended finite element method. Specifically, we demonstrated a 2-dimensional, linear implementation of the method in a MATLAB based GFEM/X-FEM code. We have described the theory and implementation of a generalized finite element approach for polycrystalline materials characterized by homogeneous grains. The sub-grid mechanics associated with the grain/grain interfaces were modeled via an applied traction condition. Numerical examples were presented, including verification patch tests for the two grain boundary behavioral models implemented. Additional examples demonstrated the more complicated abilities of the method and implementation.

Through the implementation presented here we have shown that X-FEM can represent interfaces (with both strong and weak discontinuities) with a non-conforming mesh. Some of the advantages of the method include the trivialization of the meshing process, and establishing a natural framework for the introduction of the crack-propagation problem.

There are many future directions to be considered. To realize the goal of a full micro-machining simulation, research and algorithm development for X-FEM in the areas of,

- inter-/intra-granular failure
- large deformation mechanics, grain re-contact
- remap in the context of X-FEM
- interface reconstruction and X-FEM

are being pursued.

5. Acknowledgments. The authors would like to acknowledge the contribution of their research partners at Duke University, Professor John Dolbow and Professor Tod Laursen. In addition, we would like to thank Professor Armando Duarte for his visit to Sandia Labs. The first author would like to thank Dr. Martin Heinstein for his patience with her questions and many helpful conversations. She would also like to thank her parents for the free rent and large dinners over the summer.

REFERENCES

[1] T. Childs, K. Maekawa, T. Obikawa, and Y. Yamane, *Metal Machining: Theory and Applications*, John Wiley & Sons Inc., 2000.

[2] J. Dolbow, N. Moes, and T. Belytschko, *Discontinuous enrichment in finite elements with a partition of unity method*, Finite Element Methods in Analysis and Design, 36 (2000), pp. 235–260.

[3] C. Duarte, *A Generalized Finite Element Method for polycrystals and three-dimensional branched discontinuities*. Sandia National Laboratories Seminar, August, 11 2006.

[4] C. Duarte, I. Babuška, and J. Oden, *Generalized finite element methods for three dimensional structural mechanics problems*, 77 (2000), pp. 215–232.

[5] C. Geuzaine and J.-F. Remacle, *Gmsh Reference Manual: The documentation for Gmsh 1.65*, 2006.

[6] F. Ghahremani, *Effect of grain boundary sliding on anelasticity of polycrystals*, International Journal of Solids and Structures, 16 (1980), pp. 825–845.

[7] S. Kalpakjian, *Manufacturing Processes for Engineering Materials*, Addison-Wesley Publishing Company, 4 ed., 2002.

[8] X. Liu, R. E. DeVor, S. G. Kapoor, and K. F. Ehmann, *The mechanics of machining at the microscale: Assessment of the current state of the science*, Journal of Manufacturing Science and Engineering, 126 (2004), pp. 666–678.

[9] N. Moès, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing*, International Journal for Numerical Methods in Engineering, 46 (1999), pp. 131–150.

[10] A. Simone, *Partition of unity-based discontinuous elements for interface phenomena: computational issues*, Communications in Numerical Methods in Engineering, 20 (2004), pp. 465–478.

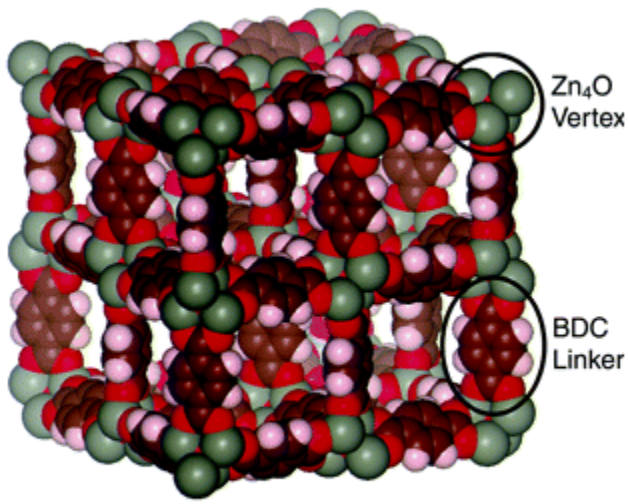
[11] A. Simone, C. Duarte, and E. V. der Giessen, *Generalized Finite Element Method for polycrystals with discontinuous grain boundaries*, International Journal for Numerical Methods in Engineering, 67 (2006), pp. 1122–1145.

MECHANICAL PROPERTIES OF METAL-ORGANIC FRAMEWORK-5 FROM FIRST PRINCIPLE CALCULATIONS

M. SHINDEL*, M. ALLENDORF , AND R. STUMPF†

1. Introduction. Metal-organic frameworks (MOFs) are an exciting new class of materials that have potential applications in areas such as catalysis, gas storage, and sensing technologies. In order to facilitate the development of devices that incorporate this type of material, it is important to be able to model the materials interaction with various substrates. This, in part, requires intimate knowledge of the materials mechanical properties, namely the elastic constants, which relate stress and strain within the system. To date, very little experimental data has been collected on the mechanical properties of MOFs. To circumvent this lack of information, first principle calculations have been made on MOF-5 to determine the systems elastic constants.

MOFs generally consist of a network of inorganic clusters interconnected by organic linkers [14]. MOF-5 employs 1,4-benzenedicarboxylate (BDC) to link together Zn4O clusters. As shown in Figure 1.1, the primitive unit cell of MOF-5 is rhombo-



MOF-5 [Ref. 10] : Zn = gray, O = red, C = brown, H = purple

FIG. 1.1. *The structure of metal-organic frameworks*

hedral, while the conventional unit cell, which is four times larger than the primitive cell, is face centered cubic. The BDC linkers form the cube edges, while the Zn4O clusters act as vertices.

2. Details of Calculations. it Ab initio methods are a powerful set of tools that allow for predictions of a material systems mechanical properties. In this work, the Vienna Ab-initio Simulation Package (VASP) [5] has been used to calculate the equilibrium volume, bulk modulus, and second-order elastic constants of MOF-5.

*University of California at Irvine
†Sandia National Laboratories

VASP uses density functional theory (DFT) [3] and a plane-wave basis set to make electronic structure calculations. A geometrical input is coupled with information about the electronic structure of the atomic species present, to calculate the total energy of a given system. In this work, calculations were made under both the local density approximation (LDA) [4], using the Ceperley-Alder functional [1] as parameterized by Perdew and Wang, and the generalized gradient approximation (GGA) [12] with the Perdew-Burke-Ernzerhof (PBE) [4] exchange-correlation functional. Results from LDA were found with both Vanderbilt ultra-soft (USPP) [15] and projector-augmented wave (PAW) [6] pseudo-potentials. Only the PAW method was used with GGA calculations. All ionic relaxations were done with the conjugate-gradient algorithm [13] included in the VASP package. The MOF-5 is quite computationally taxing to study via the methods presented here. To reduce the computational load only a single k-point, the Γ point, was used for integration over the Brillouin zone. Prior results have shown that increasing the size of the k-point mesh does not have a significant impact on the resulting MOF-system energy [9]. Furthermore, calculations were performed on the 106 atom primitive, rhombohedral cell, as opposed to the conventional cubic (fcc) cell which is four times as large (424 atoms). In all calculations the residual atomic forces were reduced below 0.01 meV/Å.

3. Procedure and Results.

3.1. Fitting to an Equation of State. To determine the optimal structure for MOF-5, VASP was used to calculate the system energy at various cell volumes. During these calculations the cell shape was held constant while the lattice parameter was varied. The lattice constants used ranged from 5% above and below the experimentally determined value [8]. In total, six data points were collected. Once the energy versus volume data had been obtained, it was fit to the empirical Murnaghan equation of state (EOS) [10],

$$E(V) = E_o + \frac{B_o V}{B_o'}\left(\frac{(V_o/V)^{B_o'}}{B_o' - 1} + 1\right) \tag{3.1}$$

This allowed for the determination of both the equilibrium cell volume (V_o) and the bulk modulus (B_o). B_o is a measure of isothermal compressibility.

The values of the parameters calculated from the fit are shown in tables 3.1 and 3.2, along with results from prior studies.

Table 3.1
MOF-5 Equilibrium Volume (Å^3)

	This Work	Ref. [8]	Ref. [9]
LDA/USPP	4196		
LDA/PAW	4206	4338	
GGA/PAW	4458		4464
Experimental		4335	4349

The USPP and PAW volume results agree well with each other and they differ from the results in [8] by ∼3%. The two experimental volumes differ by a negligible amount. The volume found with GGA differs from the experimental results by less than 3%. It should be noted that the Mattesini results [8] were obtained with a much more approximate software package (Siesta) [11], and a force cut-off five times larger than that used here.

TABLE 3.2
MOF-5 Bulk Modulus (Mbar)

	This Work	Ref. [8]
LDA/USPP	0.1797	
LDA/PAW	0.1838	0.1702
GGA/PAW	0.1543	

3.2. Elastic Constants. The elastic constants of MOF-5 were determined in a similar fashion to the way in which the equilibrium volume and bulk modulus were obtained. Energy can be described as a function of strain via a taylor series expansion [2],

$$E\left(V,\boldsymbol{\alpha}\right)=E\left(V_o,\mathbf{0}\right)+V_o\left(\sum_i\sigma_i\alpha_i\xi_i+\frac{1}{2}\sum_{i,j}C_{ij}\alpha_i\xi_i\alpha_j\xi_j\right). \tag{3.2}$$

If only small strains are applied to the system, so as to remain in the elastic regime, the taylor series can be truncated after the second order term. A designed strain is appled to the systems geometric basis,

$$\mathbf{A}'=\left(\mathbf{I}+\boldsymbol{\varepsilon}\right)\cdot\mathbf{A}, \tag{3.3}$$

thereby distorting the shape of the unit cell. The $\boldsymbol{\varepsilon}$ term in equation 3.3 is the strain matrix, $\mathbf{A}$ is the set of basis vectors and $\mathbf{I}$ is the 3x3 identity matrix. After straining the MOF-5 unit cell, VASP was used to calculate the systems energy at several magnitudes of the applied strain. The data was then fit to equation 3.2 and the relevant elastic constants were obtained from the second order coefficient [7]. A more detailed description of this method can be found in [7]. The number of distinct elastic coefficients that need to be determined is dictated by the symmetry of the system under investigation. For cubic systems there are only three individual elastic constants: C_{11}, C_{12}, and C_{44}. The value of C_{44} was obtained directly from the fitting procedure, after applying an Orthorhombic strain [7] to the unit cell. The value of the shear modulus C' = C_{11}-C_{12}, was found by applying a tetragonal shear [7] to the unit cell. The individual values of these elastic constants were found by combining the value of the shear modulus with the relationship between elastic constants, C_{11} and C_{12}, and the bulk modulus [7]. The results are shown in table 3.3. The elastic constants

TABLE 3.3
MOF-5 Elastic Constants (Mbar)

	C_{11}	C_{12}	C_{44}
This Work (LDA/USPP)	0.2234	0.1578	0.0637
This Work (LDA/PAW)	0.2263	0.1619	0.062
This Work (GGA/PAW)	0.1982	0.1323	0.0668
Ref. [8]	0.2152	0.1477	0.0754

produced here are in good agreement with those generated by Mattesini [8]. However, during the course of this investigation, it was realized that the applied strains were designed for cubic cells. However, the calculations employed the primitive form of the MOF-5 unit cell, which has a rhombohedral geometry. The presence of this error was

confirmed by the fact that the MOF-5 structure did not assume the expected geometry when distorted by the strain matrices given in [7]. Because the structure was strained in an incorrect fashion, it is probable that the tabulated values are actually linear combinations of the three elastic constants, rather than the constants themselves.

4. Future Work. Prior to further theoretical or experimental work, more correct values of the elastic constants must be obtained by performing calculations on the conventional (cubic) MOF-5 cell. The values of these constants should also be determined using the stress tensor computed by VASP (for a given strain) and the linear stress/strain relationship,

$$\sigma_i = \sum_j C_{ij}\varepsilon_j. \tag{4.1}$$

Once credible values for the elastic constants have obtained, the interactions between MOF-5 and various substrates can be investigated in order to find a favorable surface on which MOF-5 can be grown. It may also be of interest to examine any changes in physical or mechanical properties that may result from interactions between MOF-5 and any guest molecules absorbed into the structure.

REFERENCES

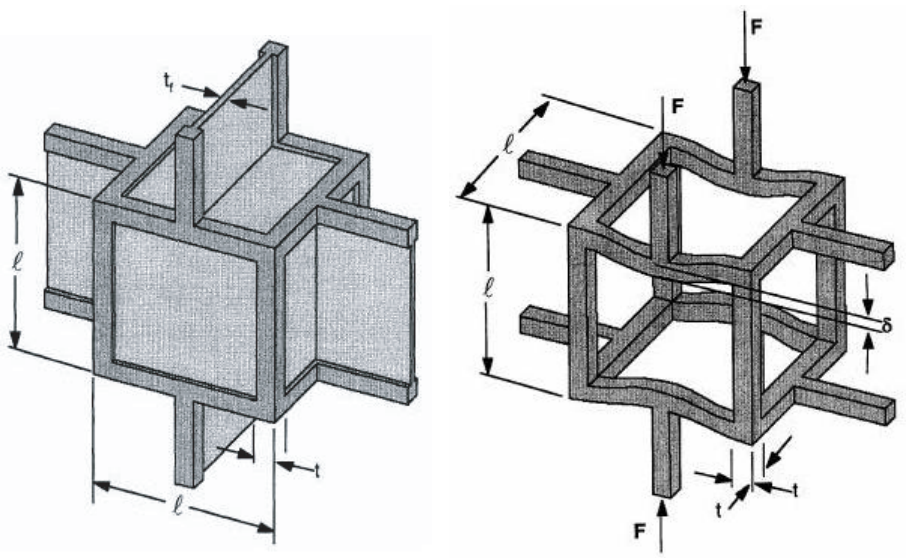
[1] D.M. Ceperley and B.J. Alder, Phys. Rev. Lett., 45 (1980).
[2] L. Fast, J.M. Wills, B. Johansson, and O. Eriksson, Phys. Rev. B, 51 (1995).
[3] P. Hohenberg and W. Kohn, Phys. Rev. B, 136 (1964).
[4] W. Kohn and L. J. Sham, Phys. Rev. A, 140 (1965).
[5] G. Kresse and J. Furthmüller, Comput. Mater. Sci., 6 (1996).
[6] G. Kresse and D. Joubert, Phys. Rev. B, 59 (1999).
[7] M. Mattesini, R. Ahuja, and B. Johansson, Phys. Rev. B, 68 (2003).
[8] M. Mattesini, J.M. Soler, and F. Ynduráin, Physical Rev. B, 73 (2006).
[9] T. Meuller and G. Ceder, J. Phys. Chem. B, 109 (2005).
[10] F.D. Murnaghan, in The Compressibility of Media under Extreme Pressures, vol. 30, 1944, pp. 244–247.
[11] P. Ordejón, E. Artacho, and J. M. Soler, Phys Rev. B, 53 (1996).
[12] J.P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett., 77 (1996).
[13] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, New York, 1986.
[14] J. Rowsell and O. Yaghi, Microporous and Mesoporous Materials, 73 (2004).
[15] D. Vanderbilt, Phys. Rev. B, 41 (1990).

NANOFILLER AND CHEMICAL STRENGTHENING OF POLYURETHANE FOAMS

B. WEISSMAN* AND A. VANCE†

Abstract. Polymer foams produced at Sandia have traditionally been used as impact protection material for the sensitive electronic components in nuclear missile systems. However, a recent market opening has shown that Sandias TufFoam may also be an ideal material for surfboard blanks. In attempting to create a more appealing and marketable material, this summers project is aimed at using nanomaterial fillers to strengthen TufFoam. Of the variety of nanomaterials tested, Cloisite30B at 2 wt% loading is the most effective, increasing TufFoams collapse strength by 8.6%. The epoxy resin Epon826 is a potent chemical strengthener, improving collapse strength by 10.6%. In addition to these discoveries, the characterization of high-shear mixed foams with Cloisite30B is also undertaken.

1. Introduction. TufFoam is a foam formulation invented by researchers at Sandia that was originally intended to replace older foams as an insulating and shock absorbing material for nuclear weapons electronics. When Clark Foam, the leading supplier of surfboard blanks, closed in December 2005 it was realized that TufFoam could potentially fill the large market share they left behind. TufFoam is a rigid closed-cell polyurethane foam (Figure 1.1) that is structurally sound, thermally insulating, and electrically insulating. Unlike previous foams it is created with environmentally



(a) Closed Cell (b) Open cell, deflecting under an applied force

FIG. 1.1. *Schematic diagrams of foam unit cell structures [3]*

friendly chemicals, replacing carcinogenic TDI (toluene diisocyanate) with Isonate181 and CFC blowing agents with water-blown CO_2. In addition to withstanding quasi-static compressive loads, TufFoam exhibits excellent durability and fracture resistance under impact conditions. The commercial availability of its constituent chemicals suggests that the manufacture of TufFoam is scalable for industrial applications.

*Brown University
†Sandia National Laboratories

The chemical constituents of TufFoam are a hydroxide-rich polyol, surfactant, an amine catalyst, water, and an isocyanate. When all of the compounds are combined, foaming and polymerization reactions occur simultaneously, as shown below [3]: In

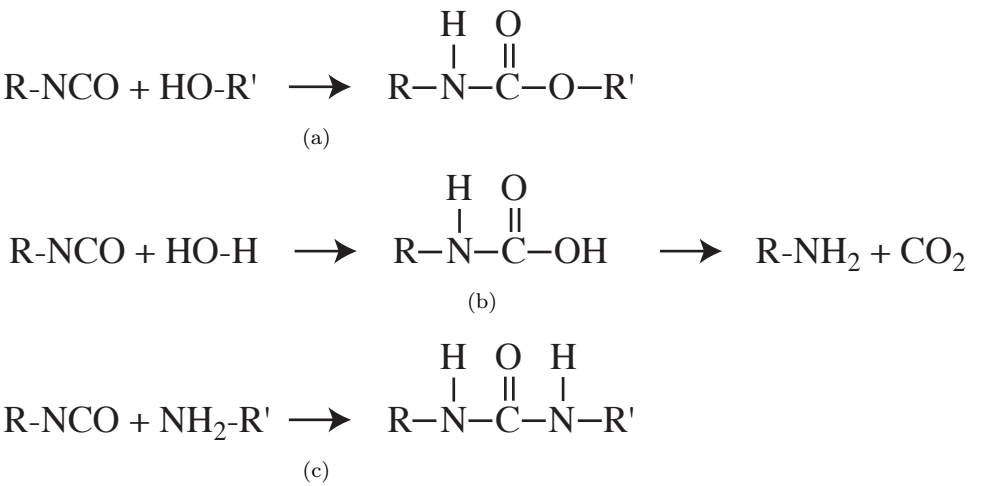


FIG. 1.2. *Foaming and polymerization reactions that occur to form TufFoam.*

reaction 1.2(a), also known as the "gel" step, the isocyanate and a hydroxyl group from the polyol form urethanes. Simultaneously, in the "blow" step of reaction 1.2(b), an isocyanate reacts with water to form an amine and carbon dioxide. The amine reacts with another isocyanate in reaction 1.2(c) to form urea linkages, while the CO_2 from 1.2(b) causes the foam to expand. This expansion is balanced by the polymerization of the reaction products of 1.2(a) and 1.2(c), causing the foam to harden as it rises [3].

The chemical formulation of TufFoam produces foam that is strong and durable. Thus, when adding filler materials it is important to avoid altering this stable formulation. Potential nanofillers were chosen on the basis of desirable physical characteristics and chemical compatibility with the TufFoam constituents. Nanoparticles were explored in lieu of micro-particles due to their favorable surface area to volume ratios and their success in other polymer compounds [1,2,4–9]. High-shear mixing of certain nanomaterials was investigated as a means of exfoliation and particle dispersion.

2. Materials and Procedure. The isocyanate source, Isonate181, and the polyol Voranol490 were both purchased from Dow Chemical Corp. The surfactant, DC-193, and the DABCO 33LV catalyst were both acquired from Air Products and Chemicals, Inc. Distilled water was obtained from a laboratory distillation source.

A variety of nanomaterials were tested throughout the course of these experiments. Fumed silica and Cloisite30B were obtained from Cab-O-Sil and Southern Clay Co., respectively, while aluminum oxide nanopowder, aluminum oxide nanopowder whiskers, and multiwall carbon nanotubes were from Aldrich. Zyvex provided NanoSolve compounds consisting of 5% SWNT in Epon826 and 10% MWNT in Epon826. Additional Epon826 epoxy resin was purchased from E.V. Roberts.

To produce foams with nanomaterial inclusions, two basic procedures were followed. One is a standard procedure, and the other is complicated by the incorporation of high-shear mixing of select ingredients. The former involves hand-mixing the Voranol and nanofiller of choice into a 500-mL cup, followed by addition of surfactant,

catalyst, and water and more hand-mixing. The Isonate is then added, followed by a minute-long period of mechanically agitated mixing using a stirrer head at 700 rpm. The appropriate amount of liquid mixture is quickly poured and clamped into a 60°C pre-heated steel mold. After a 30 minute hardening period the mold and clamp are transferred to a 60°C oven for at least four hours.

To create shear-mixed foams, the Voranol and nanomaterial are first mixed for 30 minutes in a 250-mL cup with a Jelenko dental vacuum mixer. An appropriate mass of this mixture is transferred to a 500-mL cup, and the aforementioned procedure is followed.

After heat treatment, foam cylinders 2" tall and 1.6" in diameter are cored from the bulk sample, ensuring that no foam within $\frac{1}{4}$" of the outer bulk skin is included in these samples. Quasi-static compression testing is performed at a constant strain rate on a Satec Materials testing machine to generate stress-strain curves. By manually selecting points on the linear regime, local maxima, and minima of these curves, the software determines the elastic modulus, yield stress, and collapse stress of each sample. Three to four samples per batch are tested in such a manner.

3. Results and Discussion. The first objective of this undertaking was to determine which nanomaterials were best suited for strengthening TufFoam. Six foams, containing 1 wt% alumina nano-powder, 1 wt% alumina nano-whiskers, 0.5 wt% fumed silica, 0.5 wt% Cloisite30B, 0.1 wt% MWNT, and 0.1 wt% SWNT respectively, were produced by the methods described above. Using collapse stress as a measure of the foam strength, Figure 3.1 was produced.

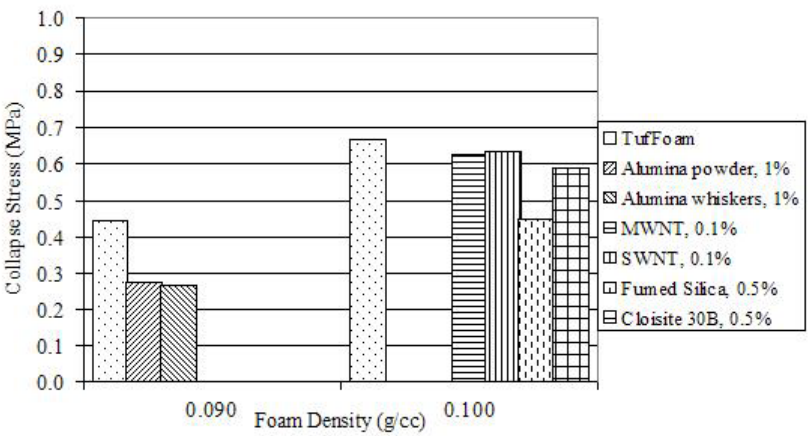
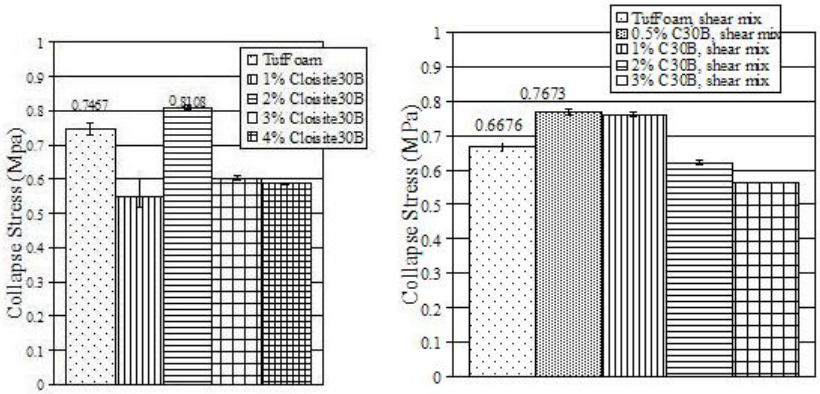


Fig. 3.1. *Effect of nanomaterial identity on collapse stress*

Since foam density plays a very large role in strength, each foam in Figure 3.1 contains a reference TufFoam sample at the same density. Comparing each sample to its respective control reveals that both alumina materials and the silica material are inferior strengtheners and actually cause a weakening of the foam. This weakening is most likely caused by interruption of the polymer matrix by the filler material. For a material to be a successful strengthener, the benefits it offers must outweigh this matrix disruption. Although the nanotubes and Cloisite also weaken the foam, they weaken it to a smaller degree and thus warrant further investigation.

To better understand the strengthening mechanism of Cloisite30B, it was tested in

foams at weight loadings of 1%, 2%, 3%, and 4%. The results of compression testing are shown in Figure 3.2(a). In addition, different Cloisite30B samples underwent



(a) Normal Cloisite30B composites. (b) High-shear mixed composites.

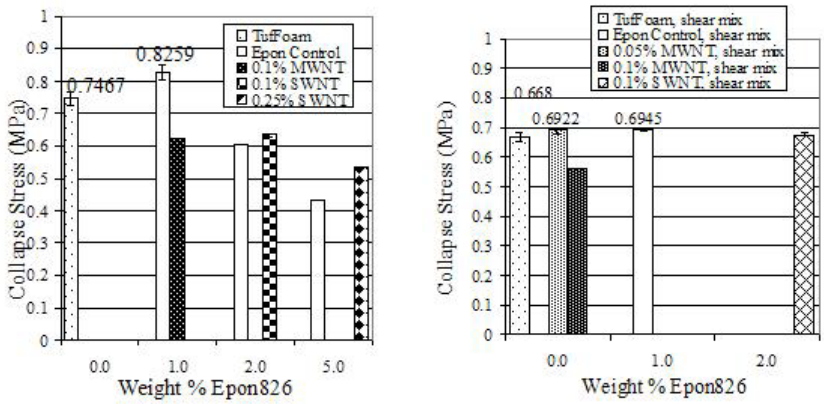
FIG. 3.2. *Effects of Cloisite30B and high-shear mixing on collapse stress.*

high-shear mixing in an effort to fully exfoliate the clay nanosheets. Exfoliation would theoretically cause a lower weight loading to produce a significant strength increase, as uniform dispersion of Cloisite would improve cross-linking and physical strengthening efficiency. The results of compression testing of the high- shear mixed samples are shown in Figure 3.2(b). These foams and all foams discussed henceforth in this report have densities of 0.100 ± 0.002 g/cc.

The normal Cloisite30B testing reveals that at a loading of 2 wt%, the collapse stress of TufFoam is increased by 8.6%. Once again, too much or too little filler causes matrix degradation without sufficient supplemental strengthening. In the high-shear mixed samples, maximal strength is attained by the 0.5 wt% loaded sample. However, high-shear mixing decreases the strength of all samples, exemplified by the TufFoam control growing weaker by 11.8%. One possible explanation for this decrease in strength is that the process of mixing imparts heat to the mixture, which decreases viscosity. This warmer, less viscous solution could react differently with the other chemicals or behave differently in the cure process. The nano-filled foams could also be weakened by physical destruction of their fillers, but TEM studies are required for verification.

A similar range of foams were created for testing the SWNT and MWNT materials. However, the nanotubes available were mixed with Epon826, meaning that a certain weight loading of nanotubes also included a greater fraction of the epoxy. As such, control samples which contained the corresponding amount of epoxy (and no nanotubes) were prepared for comparison. Results from subsequent compression testing are shown in Figure 3.3(a). After this preliminary testing, dry MWNT were obtained and incorporated into foams via high-shear mixing. Compression data from these shear-mixed samples, along with shear-mixed epoxy control and shear-mixed SWNT/epoxy combination foams, are shown in Figure 3.3(b).

The normal CNT materials exhibit no strength improvement over the original TufFoam formulation. However, the Epon control with 1 wt% epoxy unexpectedly shows a collapse stress 10.6% greater than that of TufFoam. This phenomenon explains the observed behavior: small amounts of epoxy strengthen the polymer matrix, while



(a) Normal CNT/Epon Composites. (b) High-shear mixed CNT composites.

FIG. 3.3. *Effects of MWNT, SWNT, Epon826, and high-shear mixing on collapse stress.*

larger amounts weaken it. In samples with small amounts of epoxy (such as the 0.1% MWNT with 1% Epon), the nanotubes serve as a weakening agent. In foams that contain larger quantities of epoxy (such as the 0.25 SWNT with 5% Epon), the nanotubes strengthen the mixture. In this manner, the nanotubes act as a moderator and bring either high- or low-strength materials toward an intermediate value. Once again, none of the high-shear mixed samples are stronger than unmixed TufFoam, but a small improvement is observed for the shear-mixed 0.1 wt% SWNT material compared to the normal sample.

With the observation that small amounts of epoxy improved collapse strength, it was hypothesized that an epoxy/Cloisite composite foam would enhance strength even more. According to Jia et al [5] the addition of organophilic montmorillonite to interpenetrating polymer networks (IPNs) of polyurethane and epoxy resin strengthens the composite. The sequential IPN formation is thought to exfoliate and disperse the nanoclay sheets, improving strength at a lower filler concentration. This work was completed for non-foamed polymer composites, and the same principles were assumed to be applicable to foamed systems. To fully test this assumption, TufFoam with varying degrees of Epon826 and Cloisite30B fillers were produced and tested. The compression results are shown below in Figure 3.4.

No drastic strength improvements were found to occur in the Epon/Cloisite combination system. In fact, none of the combinatorial foams produced in this manner matched the strength of the separated fillers. In order to fully understand why this occurs, more detailed characterization of the foams is required. Transmission electron microscopy (TEM) and wide-angle X-ray diffraction (WAXD) studies are needed to illustrate the degree of clay dispersion to verify that the IPN exfoliation effect doesnt work in foamed systems.

In addition to collapse stress values, the stress-strain curve produced in compression testing is a useful tool to evaluate foam strength and integrity. Figure 3.5 contains the stress-strain curves for the TufFoam baseline and the three strongest foams produced in these experiments. Each foam in this figure displays a similar stress-strain curve, implying that fracture toughness and impact strength were not compromised in creating a stronger foam. The difference in collapse strength is evidenced by the

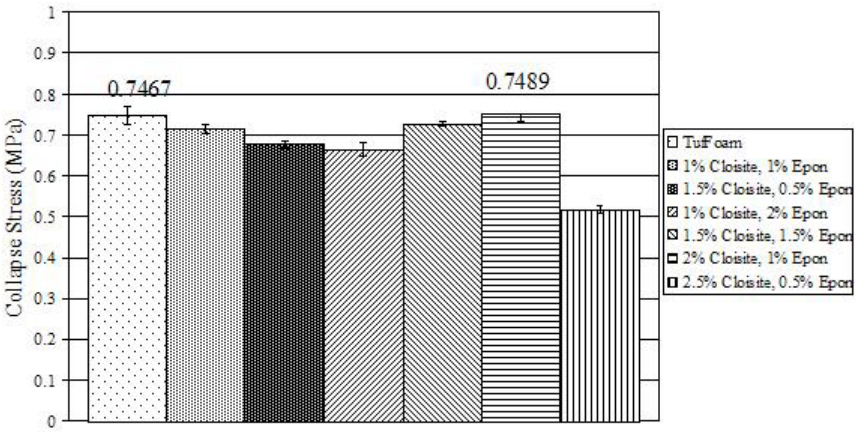


FIG. 3.4. *Effect of various Epon826/Cloisite30B filler combination on collapse stress.*

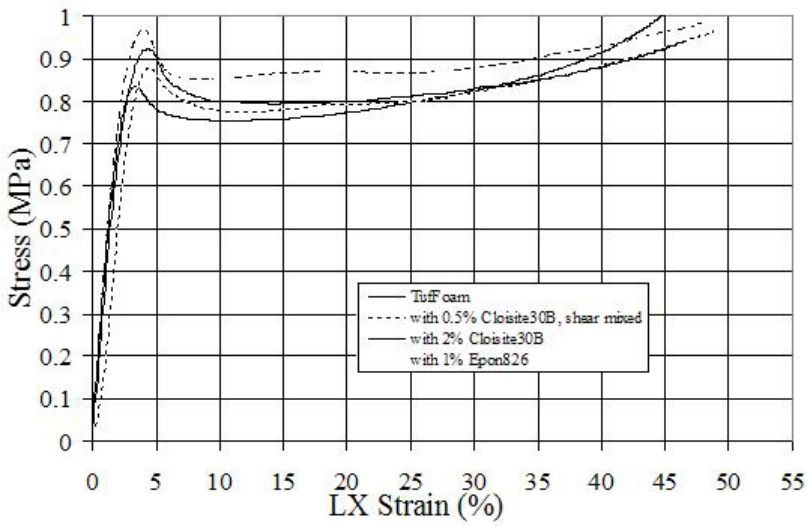
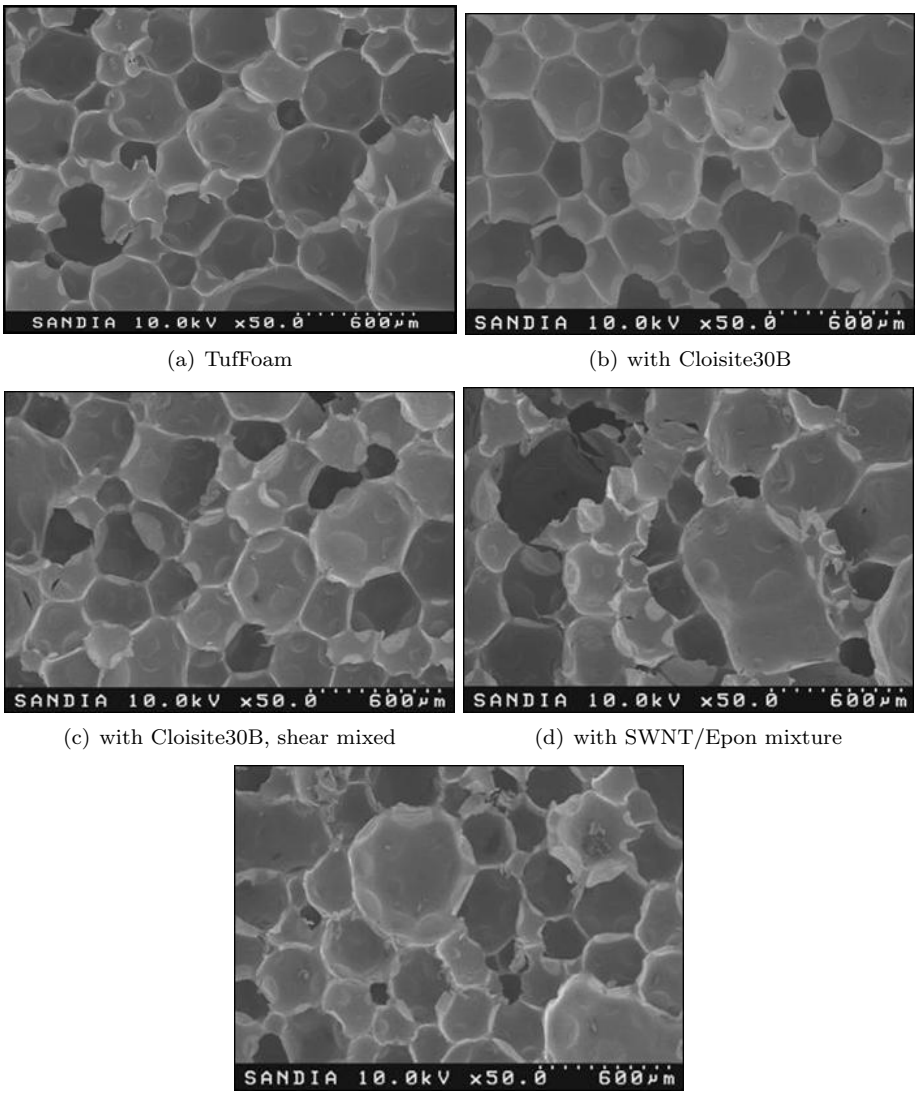


FIG. 3.5. *Stress-strain curves for the three strongest foams produced and a TufFoam control.*

difference in peak and trough height of each curve.

The final characterization performed on these foams was Scanning Electron Microscopy (SEM). While SEM provided general information about foam structure and cell size, it was not detailed enough to yield any insight into nanomaterial exfoliation or polymer matrix disruption. The images presented in Figure 3.6 show that the nanofilled foams have a similar structure and cell size to the TufFoam control. The images also suggest that high-shear mixed foams (or at least the SWNT/Epon foam) exhibit a higher frequency of smaller cells.

4. Conclusions. Through careful and systematic testing of various nano-fillers, materials and loading conditions were discovered that improve the collapse strength of standard TufFoam. The nanoclay Cloisite30B at a 2 wt% loading was found to



(a) TufFoam (b) with Cloisite30B

(c) with Cloisite30B, shear mixed (d) with SWNT/Epon mixture

(e) with SWNT/Epon mixture, shear mixed

FIG. 3.6. *SEM images depicting foam structure and cell size.*

increase collapse strength by 8.6%, while the epoxy resin Epon826 at a 1 wt% loading increased collapse strength by 10.6%. High-shear mixing was found to decrease the collapse strength of nearly all materials tested, but if the inherent thermal effects of mixing can be overcome, shear-mixed Cloisite30B at a 0.5 wt% loading may prove to be even stronger than the epoxy-containing foam. The IPN dispersion of organophilic montmorillonite discovered by Jia et al [5] was found to be inapplicable to foamed systems, but TEM and WAXD studies are required to verify this theory with confidence. SEM characterization verified a normal foam structure and hinted at a cell shrinking phenomenon due to shear-mixing.

REFERENCES

[1] X. Cao, *Polyurethane/clay nanocomposites foams: processing, structure, and properties*, Polymer, 46 (2006), pp. 775–783.

[2] W. Chen, *Carbon nanotube-reinforced polyurethane composite fibers*, Composites Science and Technology, (2006). Article in Press accepted 25 January 2006.

[3] S.H. Goods, Tech. Report SAND99-8200, Sandia National Laboratories, 1999. Unlimited Release.

[4] I. Javni, *Effect of nano- and micro-silica fillers on polyurethane foam properties*, Journal of Cellular Plastics, 38 (2002), pp. 229–239.

[5] Q. Jia, *Synthesis, characterization and properties of organoclay-modified polyurethane/epoxy interpenetrating polymer network nanocomposites*, Polymer International, 55 (2006), pp. 257–264.

[6] L.J. Lee, *Polymer nanocomposite foams*, Composites Science and Technology, 65 (2005), pp. 2344–2363.

[7] M. Mackay, *General strategies for nanoparticle dispersion*, Science, 311 (2006), pp. 1740–1743.

[8] Y. Zhang, *Rigid interpenetrating polymer network foams prepared from a rosin-based polyurethane and an epoxy resin*, Journal of Applied Polymer Science, 69 (1998), pp. 271–281.

[9] J. Zhu, *Reinforcing epoxy polymer composites through covalent integration of functionalized nanotubes*, Advanced Functional Materials, 14 (2004), pp. 643–648.

Nano/Micro-Fluidics

The articles in this section focus on the fundamental understanding of nanoscale fluid flow as well as novel applications of micro- and nanofluidic devices that underpin innovative portable equipment for the detection of biological and chemical agents. In Dickson *et al.* advanced continuation algorithms (LOCA) are used to investigate and map out the influence of continuum parameters, such as density and temperature, on the probability distributions on atoms in fluid systems through solution of the Ornstein–Zernike equations. Fettig & Collis develop a numerical simulation capability for modeling metathesis polymerization for the self-healing of materials. This process occurs at the micro-scale and is modeled by a combined Stokes model for the bulk fluid flow with a Lagrangian particle tracking for the catalyst that sets the polymer viscosity. A systematic verification process was implemented using the method of manufactured solutions and the resulting simulation tool was driven by `DAKOTA` to perform parameters studies for the healing process. Harder & Bochev document their progress on understanding the theory behind electrokinetically induced micro- and nanoflows as well as implementation of a simulation tool to model the motion of particles in nanochannels. Kennedy and Moody use detailed experimental measurements combined with analytical solutions to investigate the interfacial fracture toughness of gold films on silicon substrates. Kopacz, Nguyen, and Wagner explore the immersed finite element method as a tool for modeling complex fluid flows in deforming geometries typical of microfluidic systems. Rhieu, Huber, & Pennathur present a combined simulation and experimental study of the diffusivve transport of fluorescein in a nanofluidic device. Through this approach, they were able to estimate the zeta potential within a nanofluidic T-shaped channel. Terrel and Long evaluate a level-set approach to topologic optimization for the design of microfluidic devices. The section closes with a paper by von Winckel, Romero and Coutsias dealing with specialized, multiscale approaches to modeling micro-chromatographs. A common thread between these articles is the use of advanced computational and experimental techniques, often together, to advance the state-of-the-art in nano/micro fluidics.

S. Scott Collis

October 30, 2006

DISTRIBUTIONS OF ATOMS IN FLUIDS USING LOCA

K.I. DICKSON*, C.T. KELLEY†, B.M. PETTITT†, A.G. SALINGER‡, AND J.J. HOWARD‡

Abstract. The Ornstein-Zernike (OZ) equations [1] can be used to find probability distributions of atoms in fluid states where the primary unknowns are the radial pair correlation function and the direct correlation function. In order to specify conditions of the state, certain parameters such as density and temperature are chosen. While one can find a single solution to the OZ equations corresponding to particular parameter values, here we use LOCA (Sandia developed continuation software) in order to investigate solution behavior as the parameters density and temperature vary.

1. Introduction. The Ornstein-Zernike (OZ) equations [1] can be used to find probability distributions of atoms in fluid states. These equations are a set of nonlinear coupled integral equations with two unknowns. The system is closed with an algebraic constraint relating the two unknowns. Such a constraint is called an approximate closure relation, for which there are various options [2]. The one considered at this time is called the HNC equation. We now describe the equations of interest in more detail, followed by their application to numerical continuation.

1.1. The Equations. The Ornstein-Zernike equations are given by

$$h(r) = c(r) + \rho(h * c)(r)$$

where

$$(h * c)(r) = \int_{\Re^3} c(\|\mathbf{r} - \mathbf{r}'\|)h(\|\mathbf{r}'\|)d\mathbf{r}'.$$

Here,

- $\mathbf{r} \in \Re^3$,
- $r = \|\mathbf{r}\|$ is the distance of $\mathbf{r}$ from the origin,
- ρ is density, and
- $h, c \in C[0, \infty)$ are the unknowns.

The unknown h is called the radial pair correlation function and is an experimental observable from, for example, an X-ray or neutron diffraction experiment on fluids. The unknown c is called the direct correlation function defined by the above equations. The closure equation of choice here is the HNC equation

$$\exp(-u(r)/(TK_B) + h(r) - c(r)) - h(r) - 1 = 0, \qquad 0 \le r \le \infty.$$

In this equation, u is called the pair potential between particles. The standard u used is the Lennard-Jones potential

$$u(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right). \tag{1.1}$$

Additionally,

- T is absolute temperature (Kelvin),
- K_B is Boltzmann's constant (kcal/Kelvin),

*North Carolina State University
†University of Houston
‡Sandia National Laboratories

- ϵ is the well depth of the potential (kcal/mol), and
- σ is essentially the diameter of the particles (Angstroms).

In order to solve the OZ equations together with the HNC closure relation, one must specify certain parameter values, in this case, values for ϵ, σ, ρ, and T. However it is possible to get a sense of how solution behavior changes as one or more parameter values vary. This process is called numerical continuation.

1.2. Numerical Continuation and LOCA. Consider a set of non-linear, parameter dependent equations of the form

$$G(x, \lambda) = 0 \tag{1.2}$$

where $x \in \Re^N$ is the unknown and $\lambda \in \Re$ is a real number parameter (although in general, there may be more than one parameter). The idea of numerical continuation is to find solutions x corresponding to various values of λ and to investigate the solution behavior as λ varies. In particular, one is often interested in detecting solution paths that undergo bifurcations. In the context of the OZ equations, we can think of correlation functions h and c as the unknown x and the parameters ρ and T as λ. We would like to understand what happens to the solutions h and c as the parameters ρ and T vary.

There are many established numerical techniques and software dedicated to solving numerical continuation problems like the one just described. Trilinos (overseen by Mike Heroux, 1414) is a collection of free software packages written in C++, each one designed by a Sandia development team. The packages may stand alone, but are designed to integrate with one another. NOX (Nonlinear Object-Oriented Solutions) is the nonlinear solver package headed by Roger Pawlowski (1416), Tammy Kolda (8962), and others. Within NOX is a package called LOCA (Library of Continuation Algorithms) developed by Andrew Salinger, Eric Phipps, Brett Bader, and Russ Hooper (1416). LOCA and NOX are the primary packages responsible for the results in §3. For more on Trilinos including download information, visit

`http://software.sandia.gov/Trilinos`.

1.3. Paper Layout. In §2, we first formulate the OZ equations into a problem of the form (1.2) and then describe the problem's discretization. In §3 we look at solutions to our problem using Trilinos, followed by a description of future goals in §4. Finally, §5 offers concluding remarks.

2. Solving the OZ/HNC Equations. In order to solve the OZ/HNC equations, we first express the OZ equations as a compact fixed point problem in the form of (1.2). We then discretize the resulting problem with the trapezoid rule. This gives us the format for the continuation problem to be solved using NOX and LOCA in §3.

2.1. The Fixed Point Problem. Recall that the OZ equations are given by

$$h(r) = c(r) + \rho(h * c)(r) \tag{2.1}$$

where

$$(h * c)(r) = \int_{\Re^3} c(\|\mathbf{r} - \mathbf{r}'\|)h(\|\mathbf{r}'\|)d\mathbf{r}'. \tag{2.2}$$

The convolution $h * c$ can be computed with only one-dimensional integrals using the spherical-Bessel transform. Assuming h decays sufficiently rapidly, we define

$$\hat{h}(k) = \mathcal{H}(h)(k) = 4\pi \int_0^\infty \frac{sin(kr)}{kr} h(r) r^2 dr$$

and

$$h(r) = \mathcal{H}^{-1}(\hat{h})(r) = \frac{1}{2\pi^2}\int_0^\infty \frac{sin(kr)}{kr}\hat{h}(k)k^2 dk.$$

We compute $h * c$ by discretizing the formula (see §2.2)

$$h * c = \mathcal{H}^{-1}(\hat{h}\hat{c}) \tag{2.3}$$

where $\hat{h}\hat{c}$ denotes the pointwise product of functions. Transforming (2.1) gives

$$\hat{h} = \hat{c} - \rho\hat{h}\hat{c}$$

so that, given c, we can compute h as

$$h = H(c) = \mathcal{H}^{-1}\left(\frac{\hat{c}}{1-\rho\hat{c}}\right).$$

Now we use the HNC closure

$$\exp(-u(r)/(TK_B) + h(r) - c(r)) - h(r) - 1 = 0, \qquad 0 \le r \le \infty. \tag{2.4}$$

to recover the fixed point map. Here u is the Lennard-Jones potential defined in (1.1). Having computed $h = H(c)$, we define

$$y = h - c$$

so $h = y + c$, and the HNC closure becomes

$$\exp(-u(r)/(TK_B) + y(r)) - y(r) - 1 = c(r).$$

Substituting $H(c) - c$ in for y gives

$$\mathcal{K}(c) = \exp(-u(r)/(TK_B) + (H(c) - c)(r)) - (H(c) - c)(r) - 1$$

where $\mathcal{K}$ is a compact fixed point map in c [1] with $c = \mathcal{K}(c)$ at a solution. The final residual is

$$G(c) = c - \mathcal{K}(c).$$

Since h can be recovered once c is known, and recalling the parameters involved in the problem, we now seek to solve

$$G(c, h, \sigma, \epsilon, T, \rho) = 0. \tag{2.5}$$

As seen in §3, we choose σ and ϵ to be constant and allow T and ρ to vary for the continuation study.

2.2. The Discretization. As alluded to in §2.1, we discretize the convolution in (2.3) by discretizing the transfom and then defining the discrete convolution as the inverse transform of the product of the transforms. We discretize the transform variable in a way that allows a fast Fourier transform (FFT) to evaluate the transform and its inverse.

First, we truncate the radial variable at L so that $r \in [0, L]$. The nodes in r are defined as

$$\{r_i^\delta\}_{i=1}^N$$

where $r_i^\delta = (i-1)\delta$ and $\delta = L/(N-1)$ is the mesh width.

The nodes in the transform variable are $k_j = (j-1)\delta_k$ where $\delta_k = \pi/L$ (so $\delta_k\delta = \pi/(N-1)$). We define, for $2 \le j \le N-1$ and $v \in \Re^N$,

$$\begin{aligned}\hat{v}_j &= \mathcal{H}(v)_j \\ &= \tfrac{4\pi\delta^2}{(j-1)\delta_k}\sum\nolimits_{i=2}^{N-1}(i-1)v_i\sin((i-1)(j-1)\delta_k\delta) \\ &= \tfrac{4\pi\delta^3(N-1)}{j-1}\sum\nolimits_{i=2}^{N-1}(i-1)v_i\sin((i-1)(j-1)\pi/(N-1)).\end{aligned}$$

Then, for $2 \le i \le N-1$,

$$\mathcal{H}^{-1}(\hat{v})_i = \frac{1}{2(i-1)\pi\delta^3}\sum\nolimits_{j=2}^{N-1}k\hat{v}_j\sin((i-1)(j-1)\pi/(N-1)).$$

Finally, define for $2 \le i \le N-1$,

$$(u*v)_i = \mathcal{H}^{-1}(\hat{u}\hat{v})$$

where $\hat{u}\hat{v}$ denotes the component-wise product. We set $(u*v)_N = 0$ and define $(u*v)_1$ by linear interpolation

$$(u*v)_1 = 2(u*v)_2 - (u*v)_3.$$

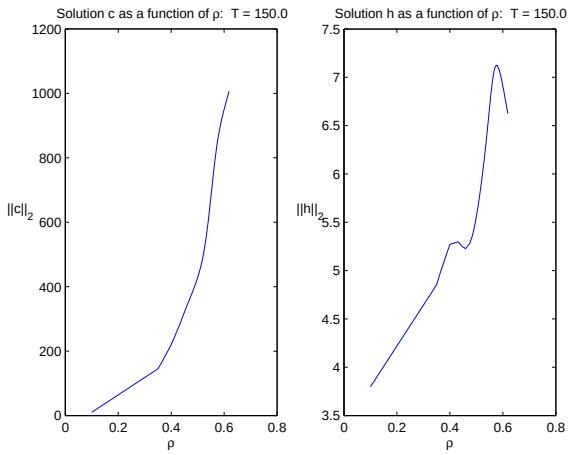
We use a fast sine transform to compute the transform and inverse transform.

3. Results. Now that we have defined and discretized the continuation problem (2.5), we are ready to solve it with the help of Trilinos.

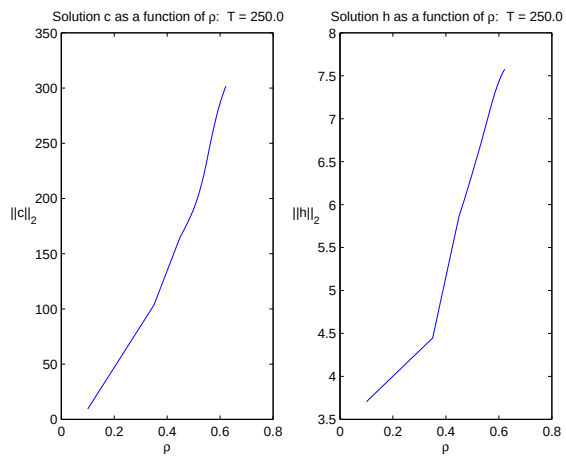
In order to solve the problem in LOCA, we were required to write two major components of C++ code. The first of the two is an interface between LOCA and the problem itself. The primary purpose for this is to define options for the continuation (LOCA), the nonlinear solver (NOX), and the linear solver (AztecOO, also a package in Trilinos). Specific solver methods, convergence tolerances, and maximum iterations are only a few examples of solver options that must be specified in this part of the code. Here we used pseudo-arclength as the continuation strategy, Newton's method with line search as the nonlinear solver, and matrix-free GMRES to solve for the Newton step. Once again, we allow temperature T and density ρ to be the continuation parameters in our study. As seen in the figures to follow, LOCA holds one parameter constant while continuing in the other. In general, the continuation parameter switches if a bifurcation is detected.

The second necessary piece of code is the one that defines the problem. In general, this is where the residual (1.2) is coded and queried by the solvers. For the problem of interest, it is the place that implements the compact fixed point problem and discretization from §2. Other methods can also be defined here such as a method to print out the solution with each continuation step or a method that defines a Jacobian matrix if the user wishes to supply one.

After coding, compiling, and running with Trilinos, we obtained the following plots. Figures 3.1(a) and 3.1(b) display continuation in ρ for two sample values of T and Figures 3.2(a), 3.2(b) display continuation in T for two values of ρ. Recalling that h and c are the unknowns for our problem, we plot the varying parameter against the L^2 norm of the solutions h and c. Additionally, we choose $\epsilon = .1$, $\sigma = 2.0$, $K = .002$, $N = 65$, and $L = 9.0$. Note that in Figures 3.1(a) and 3.1(b), Newton's method did not converge in the maximum number of iterations allotted around the values of $\rho = .35$ and $\rho = .4$. Thus the data is not plotted, causing the sharpness in the curve at these points.



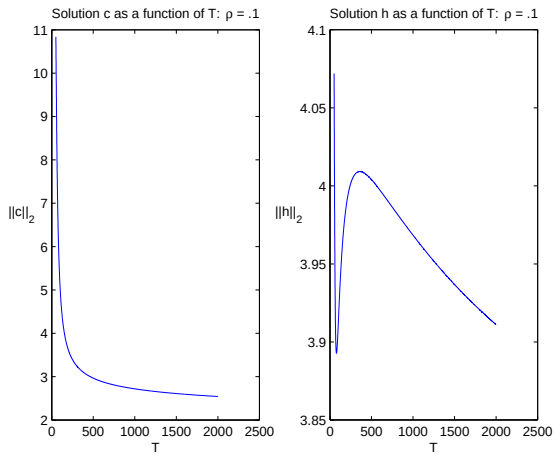
(a) $T = 150.0$



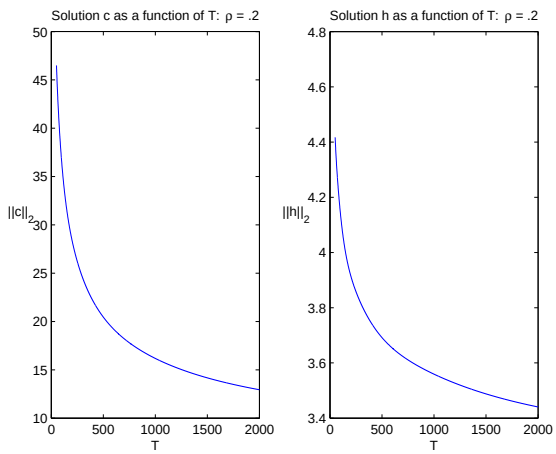
(b) $T = 250.0$

FIG. 3.1. *Continuation in ρ*

4. Future Work. The problem presented in this paper solves the OZ equations (2.1) together with the HNC closure (2.4) as the values of T and ρ vary. As mentioned in §1, HNC is not the only closure relation that can be considered. In particular, [2]



(a) $\rho = .1$



(b) $\rho = .2$

FIG. 3.2. *Continuation in T*

presents an "interpolating" closure of the form

$$\exp(-u(r)/(TK_B))\left(-a+(a+1)exp\left(\frac{h(r)-c(r)}{a+1}\right)\right)-h(r)-1=0, \qquad 0 \le r \le \infty.$$

Here, the value of a must be chosen to minimize excess chemical potential. While we do not present the expression for chemical potential at this time, one can see that within the continuation problem for the new closure resides an optimization problem that selects the appropriate value for a. The next stage of this research is to implement the new closure relation in LOCA and perform the same parameter study as done for the HNC closure.

5. Conclusions. Solving the Ornstein-Zernike equations together with the HNC closure relation gives probability distributions of atoms in fluid states under certain

conditions. We have implemented a continuation study using LOCA to look at solution behavior as the temperature and density of the state change. In the future, we wish to exchange the HNC closure for the closure presented in [2] for the continuation study.

REFERENCES

[1] C. T. Kelley and B. Montgomery Pettitt, *A fast algorithm for the Ornstein-Zernike equations*, J. Comp. Phys., 197 (2004), pp. 491–591.

[2] M.Marucho and B. Montgomery Pettitt, *An optimized theory for simple and molecular fluids*, 2006. preprint.

STOKES FLOW WITH POLYMERIZATION

JOHN FETTIG* AND S. SCOTT COLLIS†

Abstract. The flow of a fluid undergoing a ring opening metathesis polymerization is of particular interest in the design of microvascular networks for self-healing materials. In order to guide the experimentation and design process a finite element code has been developed to find and optimize a suitable set of parameters. The flow in the microscale is modeled using Stokes flow of an incompressible fluid coupled to the convection and diffusion of the catalyst in the fluid. The polymerization of monomer is approximated through a rise in the viscosity of the fluid, which is accomplished through Lagrangian particle tracking in the finite element mesh. In this paper we will discuss the implementation and verification of the computer code, as well as ongoing parameter studies and parameter optimization which utilize the computer code `DAKOTA` . Phase plot diagrams which identify the regimes in which healing occurs show that for certain parameter ranges no healing occurs in the test setups, which indicates that careful tuning of the experimental setup is required to achieve the goal of a microvascular self-healing material.

1. Introduction. The concept of a composite material that is capable of responding autonomically to damage was demonstrated in [9] with a polymer material capable of recovering 75% toughness in response to a crack-initiated heal event. This research in the field of polymeric composite systems, where materials are susceptible to virtually undetectable damage mechanisms and where repair of damaged parts can require both costly procedures and highly specialized technicians, is an important step in making these systems both inexpensive to maintain and reliable in cyclic thermal and mechanical stress loading environments. Thus the one-time healing system holds great promise if it can lead to a material capable of repeated, or continuous, self-healing.

The idea of embedding a microvascular system inside materials which is capable of continuous delivery of a healing agent draws inspiration from examples in nature such as the human circulatory system. A healing agent, in this case a monomer such as dicyclopentadiene (DCPD), flows through microvascular channels which run in network through the material which is embedded with a catalyst. The catalyst, in this case Grubbs catalyst, catalyzes a ring-opening metathesis polymerization of the monomer which produces a highly crosslinked polymer [6] [7]. The hydrodynamics of such a polymerizing flow will play a vital role in the determination of how such a system will respond to multiple damage events. The design of a self-limiting reaction which both heals regions of damage while leaving intact a delivery system to heal future damage is important in ensuring the longevity of such a system.

In this paper, we explore the design of this system through computer simulation. First, we give a brief summary of the equations which govern this problem. A computer code, `muflow` , which combines an Eulerian frame finite element solver with a Lagrangian particle method is examined to verify the order of accuracy of the calculation. A parameter study utilizing the DAKOTA toolkit along with this computer code is detailed in order to identify the flow regimes and constitutive models which result in successful healing.

2. Governing Equations.

2.1. Stokes Flow. The flow of healing agent is governed by the incompressible Stokes flow equations [5]. Here the viscosity term, $\nu = \mu/\rho$, is written in the general

*University of Illinois, jfettig@uiuc.edu
†Sandia National Laboratories, sscoll@sandia.gov

case where viscosity is not constant in space or time. We assume that the viscosity, however, does not change drastically in the timestep taken by the overall solver, and so we approximate the unsteady Stokes equations with the following steady equations

$$\nabla \cdot \mathbf{u} = 0$$
$$-\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) = 0$$

In order to calculate viscosity, which is a Lagrangian quantity, we must integrate a fluid particle over time in contact with a catalyst field, ϕ. We define a minimum concentration $\phi_{\min}$, below which polymerization does not occur, for example

$$\mu(\mathbf{x}(\xi), t) = F\left(\int_0^t \max(\phi(\xi,\tau) - \phi_{\min}, 0)\, d\tau\right)$$
$$F(t) = e^t\,.$$

The weak form of the Stokes flow equations is as follows:

$$\int_\Omega \nabla \mathbf{v} : (\nu \nabla \mathbf{u})\, d\Omega - \int_\Omega \frac{1}{\rho} p \cdot \nabla \mathbf{v}\, d\Omega = \int_{\partial\Omega} \mathbf{v} \cdot \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n}\right)\, d\Gamma$$
$$\int_\Omega q \nabla \cdot \mathbf{u}\, d\Omega = 0$$

Where $\mathbf{v}$ and q are suitably chosen test-functions. In discrete form, this corresponds to the following linear algebra problem:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{g} \end{Bmatrix}$$

Where $\mathbf{A}$ is the vector-Laplacian matrix and $\mathbf{B}$ is the divergence matrix [3]. The vectors $\mathbf{f}$ and $\mathbf{g}$ are boundary condition terms.

2.2. Convection-Diffusion of Catalyst. The viscosity calculation depends on the presence of catalyst in the healing agent. This catalyst concentration, ϕ, is governed by the advection-diffusion equation [10]:

$$\frac{\partial \phi}{\partial t} = -\left(\mathbf{u} \cdot \nabla\right)\phi + \nabla \cdot (k \nabla \phi)$$

Here, the diffusion constant is related to the viscosity through the Schmidt number, which is the ratio of the diffusion of momentum to the diffusion of mass:

$$k = \frac{1}{\mathrm{Sc}\ \mu}$$

The weak form is given by:

$$\int_\Omega v \frac{\partial \phi}{\partial t}\, d\Omega = -\int_\Omega v\left(\mathbf{u} \cdot \nabla\right)\phi\, d\Omega + \int_{\partial\Omega} v(k\nabla\phi) \cdot \mathbf{n}\, d\Gamma - \int_\Omega \nabla v \cdot (k \nabla \phi)\ d\Omega$$

In the discrete solution to this problem, time is advanced through finite-differencing. In particular, we use a Crank-Nicolson method in combination with finite elements

to produce the following linear algebra problem:

$$[M]\left\{\frac{\phi_{n+1}-\phi_n}{\Delta t}\right\}+[K+C]\left\{\frac{\phi_{n+1}+\phi_n}{2}\right\}=\{\ \mathbf{f}\ \}$$

where

$$\begin{aligned}
[M]_{ij} &= \int_\Omega \psi_i\psi_j \\
[K]_{ij} &= \int_\Omega \nabla\psi_i\cdot(k\nabla\psi_j) \\
[C]_{ij} &= \int_\Omega \psi_i(\mathbf{u}\cdot\nabla\psi_j) \\
\{f\}_i &= \sum_{j\in\partial\Omega}\phi_j\int_\Omega \psi_i(k\nabla\psi_j)\cdot\mathbf{n}
\end{aligned}$$

2.3. Boundary Conditions. The boundary conditions on the convection-diffusion equation must take into account the behavior of the system we are trying to model. Namely, on the solid wall boundaries we should have no diffusion or convection, and on the inflow and outflow we would like the catalyst to convect and diffuse away to infinity. The first of these conditions is straightforward, since we enforce a no-slip boundary condition on the Stokes flow and can additionally set the boundary integral in the weak form of the convection-diffusion equation to zero.

To achieve the effect of the catalyst diffusing away to infinity at both the inflow and the outflow, we choose to implement infinite elements [1] [4]. The infinite elements allow us to set the boundary conditions on the diffusion at infinity, rather than at the end of our finite domain. This reduces the amount of edge effect we incur by truncating the domain modeled, and makes the model more physically accurate.

3. Verification.

3.1. Method of Manufactured Solutions. First, we define a solution for $\mathbf{u}$, p, μ, and ϕ. Through substitution into the governing equations for our problem, we find source terms that correspond to this particular solution [8]:

$$\begin{aligned}
\nabla\cdot\mathbf{u} &= Q_1 \\
\frac{\partial\mathbf{u}}{\partial t}+\frac{1}{\rho}\nabla p-\nabla\cdot(\nu\nabla\mathbf{u}) &= \mathbf{Q}_2 \\
\frac{\partial\phi}{\partial t}+(\mathbf{u}\cdot\nabla)\,\phi-\nabla\cdot(k\nabla\phi) &= Q_3
\end{aligned}$$

The relationship between μ and ϕ is chosen to simplify the coupling between the Stokes flow equation and the convection-diffusion equation:

$$\begin{aligned}
\mu &= f\left(\int_0^t \phi(\xi(\tau),\tau)\,d\tau\right) && \text{where} \\
f(x) &= x && \text{so that} \\
\phi(\xi(t),t) &= \frac{\partial\mu}{\partial t}
\end{aligned}$$

TABLE 3.1
Velocity convergence data on the rectangular geometry

Number of elements	u-l_2 norm	ratio	order	v-l_2 norm	ratio	order
56	1.28356e-03			2.85123e-03		
224	8.68444e-05	14.78	3.89	1.90155e-04	14.99	3.91
896	5.64300e-06	15.39	3.94	1.22694e-05	15.50	3.95
3584	3.69152e-07	15.29	3.93	7.82259e-07	15.68	3.97

TABLE 3.2
Pressure convergence data on the rectangular geometry

Number of elements	p-l_2 norm	ratio	order
56	2.69353e-03		
224	6.89240e-04	3.91	1.97
896	1.68112e-04	4.10	2.04
3584	4.14434e-05	4.06	2.02

where $\xi(t)$ is the position of a fluid particle at time t. Thus, by specifying $\mathbf{u}$, p, and μ, we have fully specified the problem. We choose $\rho = 1$, and recall that $k = 1/(\text{Sc}\ \mu)$

3.2. Stokes Solve. We begin by verifying the individual solvers (Stokes and convection-diffusion) separately to verify their independent order-of-accuracy before testing their combined behavior. For the Stokes solver, we choose the following solution:

$$\begin{aligned} u &= \sin(x+y) \\ v &= -\sin(x+y) \\ p &= \sin(x^2+y^2) \\ \mu &= \mu_o + \mu_o \sin(x+y) \end{aligned}$$

Two domains were investigated. The first domain chosen is $-0.1 < x < .7$, and $.3 < y < 1.0$. This avoids any possible symmetry in the solution, which would potentially hide a mistake in the solver. With a refinement ratio is 2, we find super-convergent behavior in the velocity and optimally convergent behavior in the pressure field. See Tables 3.1 and 3.2 for the velocity and pressure convergence data.

The second domain is depicted in Figure 3.1. It is a curved geometry, chosen to reduce the super-convergent behavior in the velocity and test the solver on non-regular meshes. The convergence results shown in Figure 3.3 and 3.4 still show super-convergent behavior in the velocity, however the convergence has been reduced from order 4 to 3.5.

4. Computational experiments. First, the code was run on a set of parameters which produced greater than 80% healing on two different domains. The results for this are shown in Figure 4.1. Shown are the initial catalyst concentrations, the initial total velocities, the final catalyst concentrations, and the final viscosities.

Using the finite element code developed to simulate the microvascular fluid flow problem, `muflow` , we explore the parameter space by driving it with the `DAKOTA` [2] toolkit. The domain specified is a rectangular region. The parameters that are varied by `DAKOTA` include the aspect ratio of the channel, the driving pressure, the Schmidt

TABLE 3.3
Velocity convergence data on the curved geometry

Number of elements	u-l_2 norm	ratio	order	v-l_2 norm	ratio	order
117	2.93910e-04			5.53162e-04		
468	4.00229e-05	7.34	2.88	8.34950e-05	6.63	2.73
1872	3.22303e-06	12.42	3.63	6.54367e-06	12.76	3.67
7488	2.70671e-07	11.91	3.57	5.53374e-07	11.83	3.56

TABLE 3.4
Pressure convergence data on the curved geometry

Number of elements	p-l_2 norm	ratio	order
117	8.00469e-04		
468	4.38057e-04	1.83	0.87
1872	4.37066e-05	10.02	3.33
7488	7.34302e-06	5.95	2.57

number, and the constant in the model for the viscosity, $F(t) = \exp(ct)$. The initial condition for the catalyst was chosen as:

$$\phi(x, y, 0) = e^{-5(x-x_c)^2} e^{-5(y-y_c)^2}$$

where x_c, and y_c indicate the location of maximum concentration.

Before the parameter study began, the Schmidt number was first tuned so that with zero flow in a channel of width 2 and length 8, the channel would heal to 80% with the maximum catalyst location at the center of the channel in both directions. As a first attempt, we use a zero boundary condition on the inflow, which makes 100% healing difficult.

The amount of the channel that is healed is determined using the formula

$$\%\text{healed} = \frac{1}{\mu_{max} A} \int_\Omega \mu(x, y, t)\, d\Omega$$

where μ_{max} is the maximum viscosity allowed and A is the area of the channel.

The results are shown as phase plots for various values of c and Sc , with black indicating that greater than 80% of the channel healed. Figure 4.2 shows a few other values of c (=muexp) and Sc (=dparam). Figure 4.2(a) shows the initial study done by varying the driving pressure and channel width using the values of c and Sc found above. In this case, there is a "sweet spot" for which healing occurs. In the region of small channels and low pressure, no healing occurs because too much catalyst diffuses out of the inflow. In the region of large channels and high pressure, too much catalyst is advected out of the channel.

Figure 4.2(b) depicts a set of parameters for which the "sweet spot" has been pushed into a lower pressure region. Again there is a region below and above the healed region for which no healing occurs. Figure 4.2(c) depicts a set of parameters for which healing occurs down to zero pressure and small channels, but no healing occurs in the region of high pressure and large channels. Finally, Figure 4.2(d) depicts a region of parameter space where the entire set of pressures and channels heal.

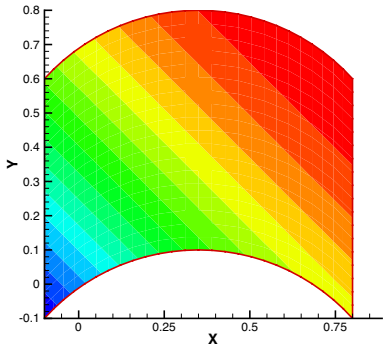


FIG. 3.1. *The curved geometry used in verification studies, show with the u velocity contours.*

5. Conclusions and Future Work. A finite element code which utilizes Lagrangian particle tracking has been developed to model the flow of healing agent in a self-healing system. The code has been verified to be at least third order convergent in velocity and second order convergent in the pressure solution coming from the Stokes solver. It remains to be verified on the convection-diffusion, but should also be third order accurate since the code is highly similar.

Using this code we have undergone parameter studies using DAKOTA, and have demonstrated the potential to use the optimization routines in DAKOTA to guide the design of the self-healing system. This combined experimentation and simulation is on-going.

REFERENCES

[1] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and Applications of Finite Element Analysis*, Wiley, fourth ed., 2001.

[2] Michael S. Eldred, Anthony A. Giunta, , Bart G. van Bloemen Waanders, Jr. Steven F. Wojtkiewicz, William E. Hart, and Mario P. Alleva, *DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis version 3.1 users manual*, Sandia Report SAND2001-3796, (2003).

[3] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, 2005.

[4] J. M. M. C. Marques and D. R. J. Owen, *Infinite elements in quasi-static materially nonlinear problems*, Computers & Structures, 18 (1984), pp. 739–751.

[5] R. L. Panton, *Incompressible Flow*, Wiley, third ed., 2005.

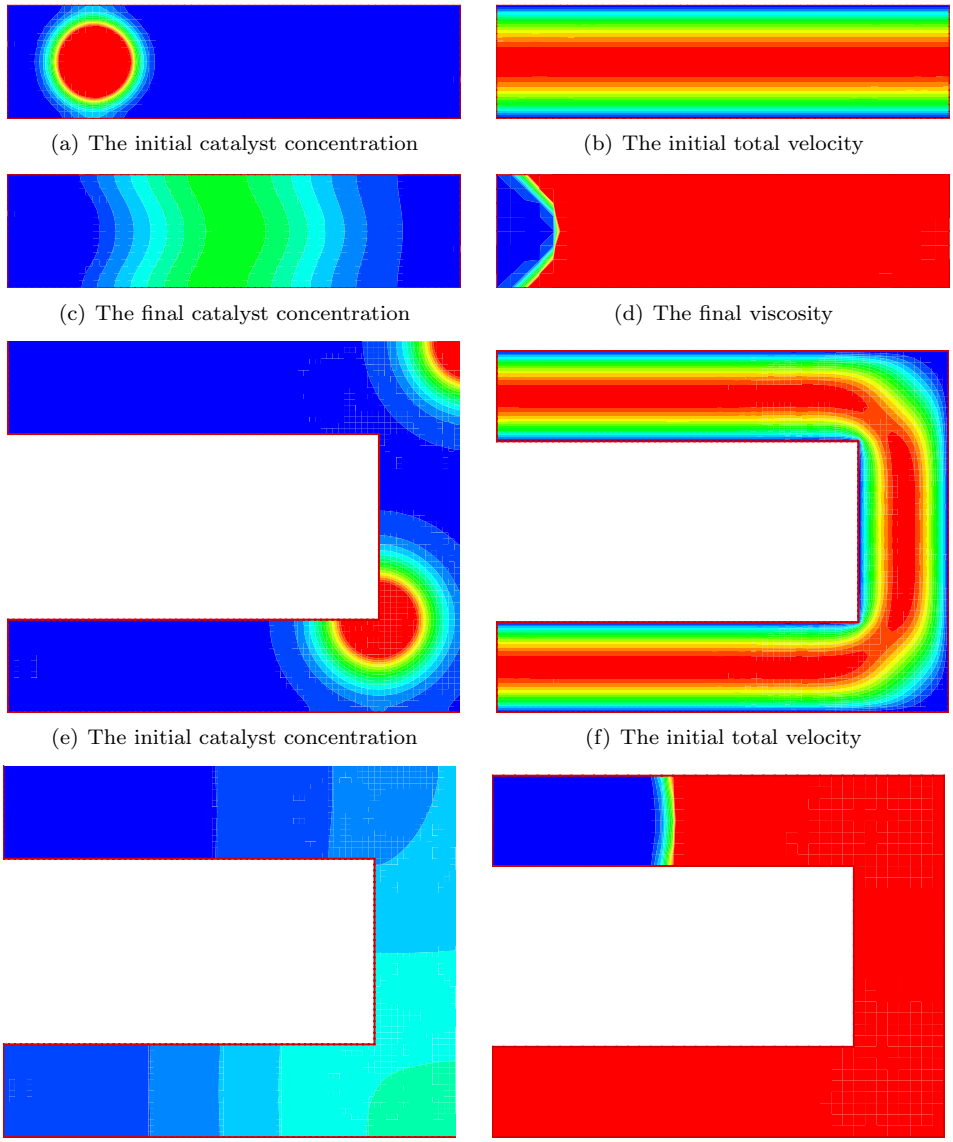
[6] J.D. Rule and J.S. Moore, *Romp reactivity of endo- and exo-dicyclopentadiene*, Macromolecules, 35 (2002), pp. 7878–7882.

[7] J.D. Rule, N.R. Sottos, S.R. White, and J.S. Moore, *The chemistry of self-healing polymers*, Education in Chemistry, 42 (2005), pp. 130–132.

[8] K. Salari and P. Knupp, *Code verification by the method of manufactured solutions*, Sandia Report SAND2000-1444, (2000).

[9] S.R. White, N.R. Sottos, P.H. Geubelle, J.S. Moore, M.R. Kessler, S.R. Sriram, E.N. Brown, and S. Viswanathan, *Autonomic healing of polymer composites*, Nature, (2001), pp. 794–797.

[10] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method, Volume 3: Fluid Dynamics*, Butterworth-Heinemann, fifth ed., 2000.



(a) The initial catalyst concentration

(b) The initial total velocity

(c) The final catalyst concentration

(d) The final viscosity

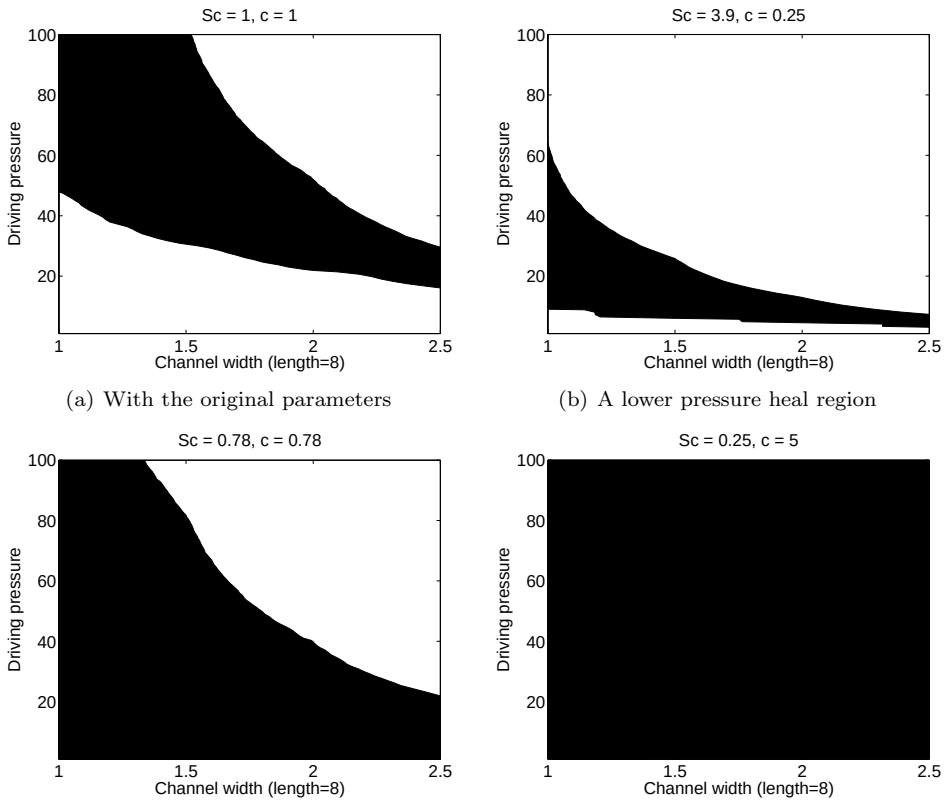
(e) The initial catalyst concentration

(f) The initial total velocity

(g) The final catalyst concentration

(h) The final viscosity

FIG. 4.1. *Two different domains with healing*



(a) With the original parameters

(b) A lower pressure heal region

(c) More than half of the region of parameter space heals

(d) The entire range of pressures and channels heal

FIG. 4.2. *Three characteristic phase diagrams*

IMPLEMENTATION OF A FINITE ELEMENT CODE IN SOLVING A COUPLED PROBLEM IN MICROFLOWS

CHRISTOPHER HARDER* AND PAVEL BOCHEV†

Abstract. Herein presented is an accumulation of information relevant to a collaboration involving Pavel Bochev and summer intern Christopher Harder. There were two major goals for the work here

1. Begin to understand the theory behind electrokinetically induced micro- and nanoflows.
2. Begin an implementation of a solver (in Matlab) which would ultimately be used to provide numerical simulations to UNM of electrophoretic motion of particles in nanochannels.

Concerning item number two, the first step was considered to be the implementation of a solver which would produce results for a pared down set of equations that could be compared to an analytical solution in the literature [9] [8]. Contained in this paper is the basic theory behind microflows and the assumptions to pare down the full set of equations in order to obtain the problem considered in [9]. Following this are the basics behind the implementation of the solver and results showing that the solver is working for the equations under consideration.

1. Introduction. Microfluidic systems arise in many applications in areas ranging from the medical field to denfense. One area of interest within the realm of these types of flows is electrokinetic flows. These flows "are important for micro- and nanoscale transport applications [3]." For one, since these flows are induced by the presence of electric fields, no moving parts are required. The benefit of this is made obvious when the difficulty of manufacturing and maintaining systems involving micropumps is considered [3]. Also, it is seen that electrokinetic flows have numerous advatages over pressure driven flows. Because electrokinetic flows are "plug-like," there is a "reduced sample species dispersion as compared to the velocity gradients associated with pressure-driven flows [2]."

Our main interest is in the numerical simulation of electrokinetic flows in cylindrical pores that contain a charged particle (approximated by a rigid sphere). These pores are filled with an electrolyte fluid and an electric field is applied. Due to what are termed the **Electric Double Layers** (EDL), fluid flow is induced.

There are two major pieces of the following paper. First, we briefly explore the theoretical underpinnings which describe electrokinetic flows. The equations which result are nonlinear and strongly coupled. It is the numerical solution of this system, with the geometry described above, that is the ultimate goal of our project. However, we would like to begin by tackling the problem at an easier level. Under what is termed the thin EDL approximation, we may neglect certain of these equations and solve a system involving the Laplace and Stokes equations. This system is coupled only through the boundary conditions on the Stokes equations. Thus, the resulting equations may be solved in series, taking the solution for the first and using it in a boundary condition on the second. Finally, instead of jumping right in and finding a solution with a sphere placed in the flow, we seek a solution to the system of equations in the case that we are simply observing flow in a cylinder induced by the electric field.

The second part of the paper deals with practical issues that arise when developing a code from the "ground up." Our implementation of a solver for the coupled Laplace-Stokes system will employ $Q2$ and $Q2/Q1$ interpolations. We will discuss the method we used to partition the cylinder into hexahedral cells. Further, we will touch on

*University of Colorado at Denver
†Sandia National Labs

how we determined the location of the nodes on this mesh. Next is a discussion on how to find the values and derivatives of trilinear and triquadratic basis functions on the physical elements. Finally, we will discuss briefly the issues of quadrature and assembly.

The code that we have written thus far is for the Laplace and Stokes equations. We would like to ensure the reliability of our code for solving these equations. Therefore, tests against known solutions for the cylinder are performed and given here.

This paper is meant as a survey of the issues related to the topics mentioned above. As such, it is brief in all areas. In the cases where more detail on the theory is desired, the reader is encouraged to look into the references provided. For the reader who wants to move quickly through this paper and avoid details, see Sections 3 and 5.

2. Theoretical Background. We will begin by exploring the theory regarding the description of microflows. We first need to understand the Electric Double Layers (EDL). Next, a discussion of the governing equations in electrokinetic flows is offered. We then will see how these equations can be simplified under certain assumptions to yield a system which is more easily solved. Finally, we will summarize the statement of a problem which Yariv and Brenner [9] found an analytic solution to.

2.1. The Electric Double Layer (EDL). When a surface comes into contact with an electrolyte, it will typically acquire a surface charge [4]. Due to the presence of the electric field induced by the static surface charge, ions of the opposite charge are attracted to the surface and ions of the same charge are repelled.

> For example, when silica is in contact with an aqueous solution, its surface hydrolyzes to form silanol surface groups. These groups may be positively charged as Si-OH_2^+, neutral as Si-OH, or negatively charges as Si-O^-, depending on the pH value of the electrolyte solution. [3]

This results in what is called the **Electric Double Layer**, which gives rise to electrokinetic phenomena in the presence of an applied eletric field $\mathbf{E}$. In many cases, the length of the EDL is on the order of a few nanometers. As such, even in the case of microflows, the EDL does not occupy a significant portion of the flow region.

We may conceptually decompose the EDL into two distinct regions. The first, named the **Stern layer**, consists of those ions which have been "glued" to the wall due to their electric attraction. The second is the **Gouy-Chapman** diffuse layer. In this layer, the ions are capable of moving freely into the bulk fluid "and therefore are available to impart work on the fluid [4]." The plane between the Stern layer and Gouy-Chapman diffuse layer is called the **shear plane**. See Figure 2.1.

2.2. Governing Equations. At this point, we make the assumptions that the fluid which we consider has constant viscosity and constant permittivity. We will take the electric charge density ρ_e to be

$$\rho_e = F \sum_i z_i c_i \tag{2.1}$$

where F is Faraday's constant [3], the sum being over the N species of ion present. Each species i obeys a conservation law in the absence of chemical reactions [3]

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (-D_i \nabla c_i + c_i(\mathbf{u} + \mu_i \mathbf{E})) = 0 \tag{2.2}$$

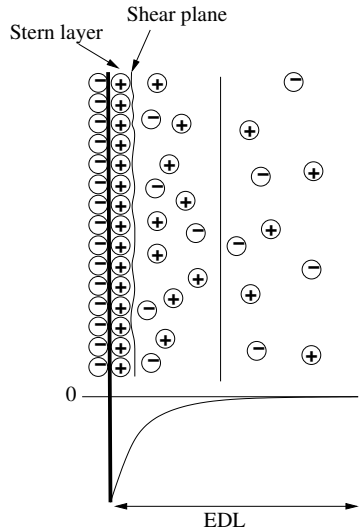


FIG. 2.1. *A schematic of the EDL and together with the electric potential ψ due to the ion distribution.*

where μ_i and D_i are the **electrophoretic mobility** and the diffusion coefficient, respectively, and $\mathbf{u}$ is the velocity of the fluid. Assuming an incompressible Newtonian fluid with constant viscosity ν, we have the following Navier-Stokes equations which govern the motion of the bulk fluid.

$$\rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \tag{2.3}$$

We take ρ_f to be the fluid density and $\mathbf{f}$ as the electrokinetic body force. Under the assumptions of incompressibility and constant electric permittivity, we have [3]

$$\mathbf{f} = \rho_e \mathbf{E}.$$

Due to the distribution of ions in the Gouy-Chapman diffuse layer, there is a net electric charge there. Taking ε as te electric permittivity, we can relate the local electric potential ψ to the electric charge density by

$$\nabla^2 \psi = -\frac{\rho_e}{\varepsilon}. \tag{2.4}$$

The bulk fluid far from the walls has no net electric charge.

Let us now make the further assumtion, for the moment, that we are considering the EDL for a flat plate. We then have the concentration of ion species i defined by the Boltzmann distribution

$$c_i = c_{\infty,i} \exp\left(-\frac{ze\psi}{kT} \right) \tag{2.5}$$

where "$c_{\infty,i}$ is the molar concentration of ion i in the bulk, z is the valance number of the ion, ϕ is the local potential, T is the temperature, e is the charge of an electron and k is Boltzmann's constant [4]." If we further assume that the electrolyte is symmetric

in that it has equal concentrations of monovalent ions, then on using Equations (2.5) and (2.1) in Equation (2.4), we have the Poisson-Boltzmann equation

$$\nabla^2\psi = \frac{2Fzc_\infty}{\varepsilon}\sinh\left(\frac{ze\psi}{kT}\right) \tag{2.6}$$

where c_∞ is the molar concentration of both ions in the bulk fluid [3] [4] [2].

Using a nondimensionalized form of the Poisson-Boltzmann equation, Karniadakis et al. [3] show that the electrokinetic potential decays rapidly to zero across the EDL from a maximum at the wall. See Figure 2.1.

2.3. Electroosmosis. We see that the flow in the Navier-Stokes equations (2.3) is driven by a body force which is proportional to an electric field $\mathbf{E}$. This external electric field, which induces the flow, can be represented as [3]

$$\mathbf{E} = -\nabla\phi. \tag{2.7}$$

Assuming constant electric conductivity and the laws of elastostatics, we have since $\nabla\cdot\mathbf{E} = 0$. Therefore,

$$\nabla^2\phi = 0 \tag{2.8}$$

Furthermore, if we assume the boundary is electrically insulating, then the boundary condition

$$\nabla\phi\cdot\mathbf{n} = 0 \tag{2.9}$$

holds. Once this external electric field is determined, we have the full set of equations to describe the electrokinetic flow.

2.4. Paring things down. In the case that we have "thin" EDL and the flow is steady and at low Reynold's numbers, we may make some assumptions which simplify things tremendously. Under these assumptions, we neglect the electrokinetic forces in the EDL. Taking these to be zero in the bulk fluid (since $\rho_e \approx 0$, ie - the fluid is neutral) and neglecting the gradients of velocity and time derivative in the Navier-Stokes equations, we have that the governing equations for the fluid are the Stokes equations. In this case, we have removed the body forces for driving the flow. As such, the flow is driven by the electrokinetic effects using the "slip" boundary condition

$$\mathbf{u} = \mu\mathbf{E} \tag{2.10}$$

μ being the electrophoretic mobility. Equation (2.10) is the Helmholtz-Smoluchowski slip condition [3]. Furthermore, we do not have to solve the Poisson-Boltzmann equation [3]. The problem is therefore simplified tremendously, resulting in the case where we have only to solve the following two equations

$$\begin{aligned} -\nabla^2\mathbf{u} + \nabla p &= \mathbf{0} \\ \nabla\cdot\mathbf{u} &= 0, + \text{ boundary conditions} \end{aligned} \tag{2.11}$$

and

$$\nabla^2\phi = 0, + \text{ boundary conditions.} \tag{2.12}$$

2.5. A problem statement from [8] [9]. In two papers [8] [9], Yariv and Brenner consider problems governed by the Equations (2.11) and (2.12). The geometry of the problems consists of the same structures, although the specific setup differs slightly between the two.

In [8], the authors consider the motion of an eccentrically positioned sphere moving in a cylindrical pore. The method of reflections is employed to obtain an open form analytical solution in the case that the sphere was eccentrically positioned in a cylinder of infinite length, yet still "far" from the cylinder wall. This represents a generalization of the result obtained by Keh and Anderson [7], who considered the case of a concentrically positioned sphere. In [9], the authors consider an eccentrically positioned sphere in an infinitely long cylindrical pore, this time under the assumption that the sphere fits closely with the cylinder.

The reader is referred to the papers for details regarding the calculations and for more specific information. Here we will present the geometry used in [9]. We will let the spherical particle have radius a and the cylinder have radius $(1+\varepsilon)a$. The cylinder will be filled with an electrolyte solution and a uniform electric field E_∞ applied parallel to the cylinder walls. Denote by $\mathcal{W}$ and $\mathcal{P}$ the surfaces of the cylinder and the sphere, respectively. Assume that they have uniform charge densities with zeta potentials $\zeta_{\mathcal{W}}$ and $\zeta_{\mathcal{P}}$. Further, let the vectors $\hat{\boldsymbol{n}}_{\mathcal{W}}$ and $\hat{\boldsymbol{n}}_{\mathcal{P}}$ be the unit normal vectors to their respective surfaces. Under the assumption of thin EDL, we have the following dimensionless forms of the relevant equations (see [9] for details). Owing to the use of an infinitely long cylindrical pore, far-field conditions are used for the "ends" of the cylinder.

To begin, the electric potential ϕ satisfies (see Equation (2.7))

$$\nabla^2\phi = 0 \tag{2.13}$$

with boundary conditions (see Equation (2.9))

$$\hat{\boldsymbol{n}}_{\mathcal{W}} \cdot \nabla\phi = 0 \text{ on } \mathcal{W} \tag{2.14}$$

$$\hat{\boldsymbol{n}}_{\mathcal{P}} \cdot \nabla\phi = 0 \text{ on } \mathcal{P} \tag{2.15}$$

$$\nabla\phi \to -\hat{\boldsymbol{z}} \text{ as } |z| \to \infty. \tag{2.16}$$

Equation (2.16) is the far-field condition and $\hat{\boldsymbol{z}}$ is a unit vector which points in the same direction as the applied vector field.

We consider next the Stokes equations governing the bulk fluid. These are

$$-\nabla^2\mathbf{u} + \nabla p = \mathbf{0} \tag{2.17}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.18}$$

with boundary conditions (see Equations (2.7) and (2.10))

$$\mathbf{u} = \gamma\nabla\phi \text{ on } \mathcal{W} \tag{2.19}$$

$$\mathbf{u} = \nabla\phi + \mathbf{U} + \boldsymbol{\Omega} \times \mathbf{r} \text{ on } \mathcal{P} \tag{2.20}$$

$$\mathbf{u} \to -\gamma\hat{\boldsymbol{z}} \text{ as } |z| \to \infty \tag{2.21}$$

where we take $\mathbf{U}$ to be the linear velocity of the particle, $\boldsymbol{\Omega}$ the angular velocity, $\mathbf{r}$ a position vector relative to the center of the sphere, and $\gamma = \frac{\zeta_{\mathcal{W}}}{\zeta_{\mathcal{P}}}$. The far-field condition ensures the velocity approaches an electroosmotic "plug flow" profile.

As one can see from the equations, we may first find a solution for the electrostatic potential ϕ. Using this, the electric field ($\mathbf{E} = \nabla\phi$) may be determined, thereby allowing us to solve the Stokes equations.

3. The Current Project. Keeping in mind the ultimate goal of solving the full set of equations describing the electrophoretic motion of a particle moving in a cylinder, the decision was made to, as a first step, find the solution to easier problems. As such, we chose as our starting point the problems presented in [8] and [9]. The choice of these was guided by Professor Dimiter Petsev of the University of New Mexico. The benefit of the choice of these problems is they provide the opportunity to produce numerical results for a solver involving a fewer number of equations than if the full set of equations was being considered. Furthermore, the coupling is much simpler. These results may be then be weighed against the theoretical results in order to test the performance of the solver. Once things are performing well at this level, the full set of equations may then be considered.

We note that in the case that we seek a numerical solution to the problems defined in [8] and [9], we must consider a cylinder which has *finite* length. As such, the far-field conditions in Equations (2.13)-(2.21) must be replaced by conditions at the end of the cylinders which we are considering. Also, the length of the cylinder as compared to the radius should be "long" enough such that for comparisons in the middle of the flow, we obtain results that are not largely affected by the effects of the inflow and outflow boundary conditions. We will assume for the time being that the cylinder (of length L) has one face in the xy-plane and the other in the plane $z = L$.

As an intermediate step, we wanted to consider the further simplification of the above problem where the spherical particle is removed from the domain. In this case, we are simply solving the following set of equations (note the replacement of the far-field conditions).

For the electric potential ϕ,

$$\nabla^2 \phi = 0 \tag{3.1}$$

with boundary conditions

$$\hat{\boldsymbol{n}}_{\mathcal{W}} \cdot \nabla \phi = 0 \text{ on } \mathcal{W} \tag{3.2}$$

$$\phi = -z \text{ on } z = 0 \text{ or } z = L. \tag{3.3}$$

We consider next the Stokes equations governing the bulk fluid. These are

$$-\nabla^2 \mathbf{u} + \nabla p = \mathbf{0} \tag{3.4}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{3.5}$$

with boundary conditions

$$\mathbf{u} = \gamma \nabla \phi \text{ on } \mathcal{W} \tag{3.6}$$

$$\mathbf{u} = -\gamma \nabla \phi \text{ on } z = 0 \text{ or } z = L. \tag{3.7}$$

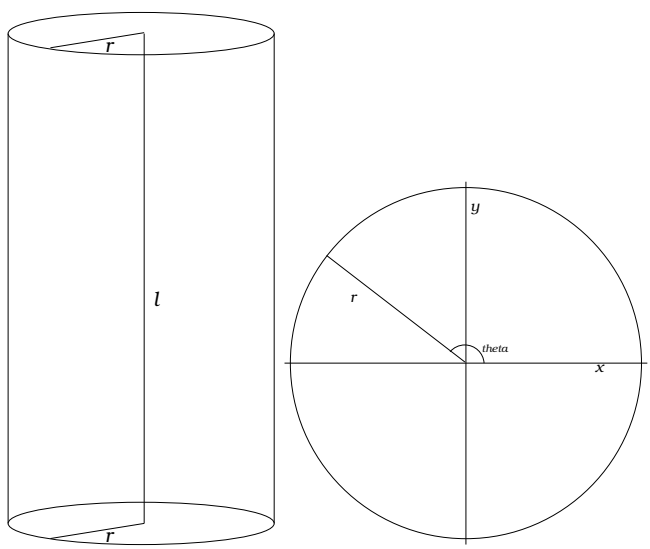
Testing of the code at this intermediate step will be carried out by comparing with exact solutions for slightly modified equations. More on this will discussed in the results section. After this section of code is running satisfactorily, the sphere will be added back into the problem (with its associated boundary conditions included in the equations). The goal for the inclusion of the sphere is to never have to remesh, but rather have the sphere placed in the flow, "moving" against the fixed mesh in the background.

4. Implementation. With the theory of the problems fully introduced and the problem which we will currently implement well-defined (Equations (3.1)-(3.7)), we now turn to the various pieces necessary to implement a code for solving the stated problem. The four main areas of interest here are the method used to generate a mesh, locating nodes on the mesh for both trilinear and triquadratic basis functions, how values of basis functions and their derivatives on a particular element were determined, considerations involved in evaluating integrals and assembling the stiffness matrix, and considerations involved in evaluating integrals and assembling the stiffness matrix.

Regarding the second point (and the first) listed above, the methods here are not optimal in that they don't produce a matrix of as narrow bandwidth as possible. Here, the goal was to get a matrix which could be inverted to obtain a solution and work on performance improvement at a later point. Qualitative descriptions of the algorigthms for determining the mesh and nodes are given.

4.1. Meshing. Assume that we begin with a cylindrical domain of length l and radius r. We would like to partition the domain into hexahedra and at the end have two output files `vertices.txt` and `elements.txt`. These two files contain the information relevant to a finite element program for locating vertices and their relationships to one another. We will see more specific information about these shortly.

4.1.1. Cylinder geometry. The geometrical setup is pictured below in Figures 4.1(a) and 4.1(b). For our purposes, we will assume that the cylinder's axis lies on



(a) The geometry of the cylidner.

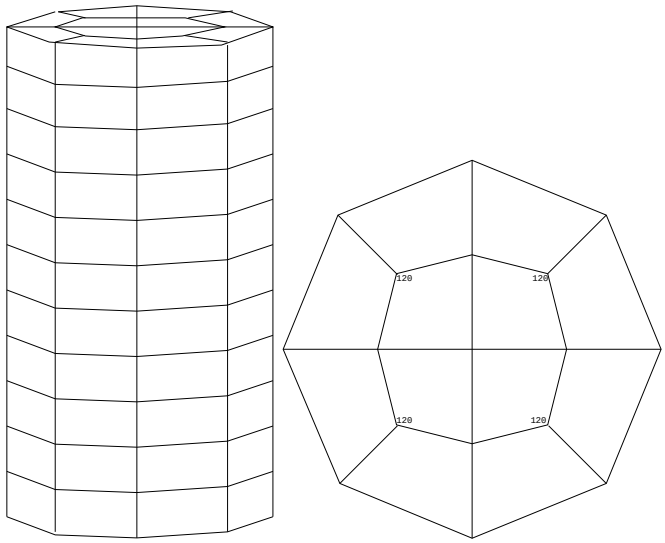
(b) The geometry of a circular section of the cylinder.

FIG. 4.1. *Geometries of the cylindrical domain which we will consider.*

the z-axis and the lower face of the cylinder lies in the xy-plane.

With the partitioning, we are not concerned at this moment about matching the circular shape of the cylinder shaft, but rather with having vertices of straight-edged hexahedra on the boundary being at a distance r from the axis of the cylinder. As such, our goal at this time is to produce a mesh that recovers the circular shape when taken with an infinite number of points at the boundary. In Figures 4.2(a) and 4.2(b)

is an example of the type of mesh we have in mind.



(a) The partitioning of the whole cylindrical domain.

(b) The partitioning of the cylinder as viewed from above.

FIG. 4.2. *An example of the type of partition for the cylindrical domain which we hope to reproduce.*

There are five parameters we identify that allow us to define our mesh.

- $\boldsymbol{r}$ - The radius of the cylinder.
- $\boldsymbol{l}$ - The length of the cylinder.
- **cpn** - The number of vertices on the circular boundary. (Short for 'circle points number').
- **csn** - The number of 'slabs' of elements extending in the radial direction when looking at a face of the cylinder. (Short for 'circle-slabs number'.)
- **zpn** - The number of 'slabs' of elements when looking at the cylinder with the z-axis perpendicular to the line of sight. (Short for 'z-slabs number'.)

For example, Figures 4.2(a) and 4.2(b) have $cpn = 8$, $csn = 2$, $zsn = 11$.

One thing to note about the mesh is that we have moved the vertices on edges which do not intersect (shown as bold lines in Figure 4.2(b)) at the axis of the cylinder so that all angles are 120 degrees. This is done so that if we increase *cpn* toward infinity, we don't get elements inside with angles which approach 180 degrees.

4.1.2. Preliminaries. To begin, we discuss the meaning of the contents of the ASC-II `elements.txt` file. First, some definitions are in order. Imagine that you have gone through and numbered each of the vertices exactly once in a mesh which you have drawn. We'll call these the **global vertex numbers** since they are members of a global numbering system for the mesh. If we look at each cell individually, we could number the vertices of this cell, going counterclockwise around one face, then moving to the next and going counterclockwise once again (the starting place being the same as for the first place, just horizontally displaced). We will call the numbers which have been assigned in this way the **local vertex numbers**. Each local numbering will have exactly the numbers $1 - 8$ since we have partitioned into hexahedra. The `elements.txt` file describes the cells according to the global vertex

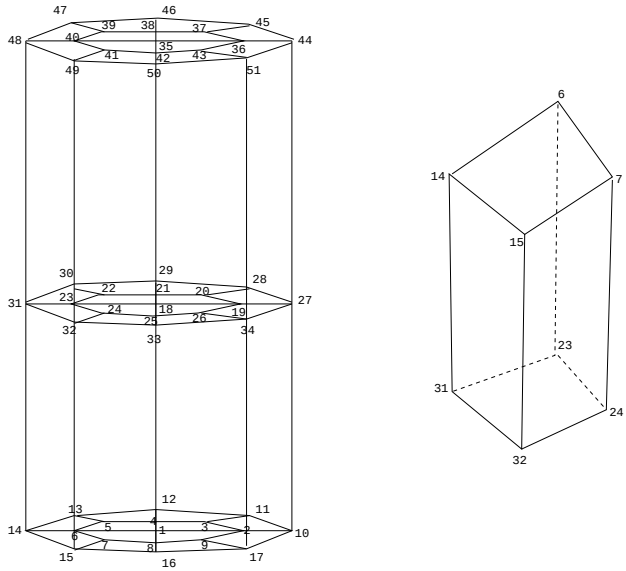


FIG. 4.3. *Global numbers are assigned to the vertices. A representative element is shown to the side.*

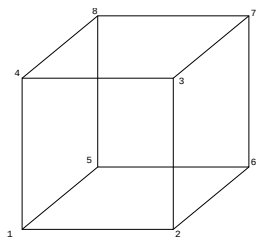


FIG. 4.4. *An example of a cell with local numbers assigned.*

number of the vertices which make it up. There are exactly as many rows as there are cells, where each row contains exactly eight numbers. These eight numbers appear in the order which corresponds to the local numbers for the element, where the entry corresponds to the global vertex number. In Table 4.1 is an example corresponding to the numbering shown in Figures 4.3 and 4.4.

⋮		⋮			⋮		⋮
6	14	15	7	23	31	32	24
⋮		⋮			⋮		⋮

TABLE 4.1
An example of the the representative cell in Figure 4.3 using the numbering as described above.

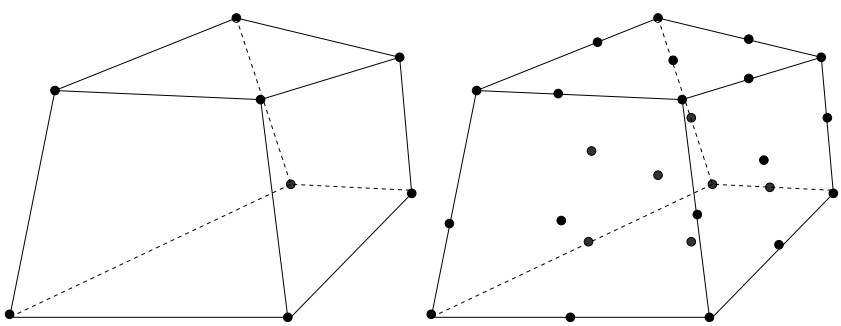
4.1.3. Algorithm. Here we see the method by which the mesh was determined and the files `vertices.txt` and `elements.txt` were created. We first describe how we determined the (x, y, z) coordinates of the vertices and how they are organized in the `vertices.txt` file. Then we will describe how `elements.txt` was created.

We will use an algorithm that assumes a global vertex numbering similar to that shown in Figure 4.3 for arbitrary *cpn*, *csn*, and *zsn*. Namely, we start at the face in the xy-plane, assigning the coordinates of the vertex in the center to the first row of an array (say 'vertices'), then working outwards, going counterclockwise, until all vertices in the plane have been given coordinates (at their respective location in the 'vertices' array). Then we move up to the next level at which we must describe the vertices. This process continues until we have assigned all vertices their coordinates. The end result is an array with (x, y, z) coordinates in the rows, where the location of the row indicates which global vertex we are looking at. The contents of this array make up the values contained in the `vertices.txt` file.

The creation of the `elements.txt` file needs no input from the `vertices.txt` file. We just need to make sure that the global vertex numbers are consistent with eachother so that when we use the information later, we don't end up using coordinates that don't have anything to do with a vertex we think we are looking at. That said, we need only to examine the number scheme to determine if there is any pattern. To this end, we may observe that there is, as may be readily verified. The exploitation of this is self-evident in the code, so here we will only discuss the organization of the code.

We must first decide how we will number the cells. In current implementation, we begin with a cell having a face in the xy-plane and one of its vertices being the center of that face. This will be cell 1. The next cell is the one immediately next to it in the z-direction. This is continued until there are no more cells in the vertical direction. We then return to the xy-plane and number a cell neighboring cell 1 and work our way vertically.

4.2. Locating nodes. We turn now to the method by which global numbering of the nodes (which relates to the type of interpolation we want to use) is performed. The code needs this information to create basis functions and their derivatives in order to evaluate integrals involving these. As indicated above, we have implemented both 8-node trilinear and 27-node triquadratic reference elements (see Figures 4.5(a) and 4.5(b)). A call is made to a function which determines where the nodes are located



(a) A general 8-node element in the physical domain.

(b) A general 27-node element in the physical domain.

FIG. 4.5. *Example of elements which have 8 nodes or 27 nodes.*

and what cells they belong to. Within the code, the data arrays IEN and *nodes* carry this information. The structure of the data arrays created is similar to the structure of information in the `vertices.txt` and `elements.txt` files. In fact, in the case of trilinear basis functions the nodes and vertices are exactly the same and we take the

same local numbering scheme. Therefore, the necessary information is exactly that which is stored in the `vertices.txt` and `elements.txt` files. The case where we will use triquadratic interpolation is not so easy.

A few definitions of some variables is in order here. We will take **nvt** to be the total number of vertices in the mesh and **nnd** to be the total number of nodes. In the case of triquadratic basis functions on the reference element, we will begin by taking all vertices of the mesh as nodes, and then add nodes in the middle of edges, faces, and hexahedra as necessary. The nvt vertices from the mesh are taken as the first nvt nodes. Then we start in the xy-plane and work counterclockwise and outwards, assigning numbers to the nodes that haven't yet been accounted for. Once this is done, we move up and repeat the process for the next interface of elements. Finally, we start at the lowest elements and assign all nodes in the middle of the element (by the z-axis) and then work our way up. All of the nnd nodes now have a global number. We will take the local numbering scheme in this case to be as shown in Figure 4.5(b). Note that the order of the nvt nodes that were assigned first has been changed to reflect this different local numbering.

4.3. Elements. We now explore how to describe the functions on a physical element by means of a reference element. The discussion here is general for all Lagrange type interpolations in 3D in which we associate one degree of freedom with each node. Here, we will also assume a general domain (for some arbitrary problem) which has been partitioned into hexahedral elements. We shall let this partitioning be denoted by $\mathcal{T}$ and the elements denoted by K.

To begin, we give a few definitions

- **reference element** - The element on the cube $[-1,1]^3$ on which we may define trilinear, triquadratic, etc. basis functions and the location of nodes necessary to define these functions. The coordinates in this domain will be denoted $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$.
- **physical element** - An element that is defined on a cell which is a member of the mesh of the domain on which we are solving an equation. (These are the members K of the partition $\mathcal{T}$.) In general we do not know the basis functions. The coordinates in this domain will be denoted $\mathbf{x} = (x_1, x_2, x_3)$.

In creating the stiffness matrix and load vector associated with a discretization, it is often necessary to evaluate integrals on physical elements such as:

$$\int_K \nabla N_a \cdot \nabla N_b dx \text{ or } \int_K N_a \mathbf{b} \cdot \nabla N_b dx$$

However, as indicated in the definition, we do not in general know the basis functions of a physical element. As such, we need to work out some relationships to the functions on the reference element in order to evaluate integrals of the type shown above.

4.3.1. Reference element basis functions. Let the interpolation we use be denoted by i. The number of nodes for this element is $nnd_{el} = (i+1)^3$. As such, we have nnd_{el} basis functions for a typical Lagrangian element. The nnd_{el} basis functions for the reference element will be defined as the product of ith-order polynomials in each of the coordinate directions. Let $\hat{N}_a(\boldsymbol{\xi})$, $a \in \{1, \ldots, nnd_{el}\}$ be basis functions in the reference element. Letting $\boldsymbol{\xi}_b$ be a node for the reference element, $b \in \{1, \ldots, nnd_{el}\}$, we enforce the property $\hat{N}_a(\boldsymbol{\xi}_b) = \delta_{ab}$.

4.3.2. Reference element to physical element mapping. We define a mapping which takes points in the reference element and maps them to points in the

physical element. We will denote this mapping by $\mathcal{F}$ and its inverse by $\mathcal{G}$. Since we have physical elements defined on cells with straight edges, the mapping $\mathcal{F}$ may be defined as [6]

$$\mathcal{F} = \sum_{a=1}^{8} \hat{N}_a(\boldsymbol{\xi})\mathbf{x}_a \tag{4.1}$$

where the $\hat{N}_a$ are the eight trilinear basis functions defined on the reference element.

As was stated, we will take $\mathcal{G}$ to be the inverse of $\mathcal{F}$. Therefore, the following relations hold

$$(\mathcal{F} \circ \mathcal{G})(\mathbf{x}) = \mathbf{x} \text{ and } (\mathcal{G} \circ \mathcal{F})(\boldsymbol{\xi}) = \boldsymbol{\xi}. \tag{4.2}$$

In general, we don't know the form of the inverse mapping $\mathcal{G}$, but it turns out not to be important. We simply need to know relationships involving it. These relationships will be given below.

4.3.3. Properties of the mappings. To begin, we define the following Jacobian matrices [1]

$$J_{\mathcal{F}} = \begin{pmatrix} \frac{\partial \mathcal{F}_1}{\partial \xi_1} & \frac{\partial \mathcal{F}_1}{\partial \xi_2} & \frac{\partial \mathcal{F}_1}{\partial \xi_3} \\ \frac{\partial \mathcal{F}_2}{\partial \xi_1} & \frac{\partial \mathcal{F}_2}{\partial \xi_2} & \frac{\partial \mathcal{F}_2}{\partial \xi_3} \\ \frac{\partial \mathcal{F}_3}{\partial \xi_1} & \frac{\partial \mathcal{F}_3}{\partial \xi_2} & \frac{\partial \mathcal{F}_3}{\partial \xi_3} \end{pmatrix} \text{ and } J_{\mathcal{G}} = \begin{pmatrix} \frac{\partial \mathcal{G}_1}{\partial x_1} & \frac{\partial \mathcal{G}_1}{\partial x_2} & \frac{\partial \mathcal{G}_1}{\partial x_3} \\ \frac{\partial \mathcal{G}_2}{\partial x_1} & \frac{\partial \mathcal{G}_2}{\partial x_2} & \frac{\partial \mathcal{G}_2}{\partial x_3} \\ \frac{\partial \mathcal{G}_3}{\partial x_1} & \frac{\partial \mathcal{G}_3}{\partial x_2} & \frac{\partial \mathcal{G}_3}{\partial x_3} \end{pmatrix}.$$

If we denote the rows of $J_{\mathcal{G}}$ by $\nabla \mathcal{G}_j$ and the columns of $J_{\mathcal{F}}$ by V_i and write their determinants as $|J_{\mathcal{F}}|$ and $|J_{\mathcal{G}}|$, it is possible to derive the following relationships [1]

$$\nabla \mathcal{G}_i = \frac{1}{|J_{\mathcal{F}}|}\left(V_j \times V_k\right) \text{ and } V_i = |J_{\mathcal{F}}|\left(\nabla \mathcal{G}_j \times \nabla \mathcal{G}_k\right). \tag{4.3}$$

4.3.4. Evaluation of basis functions on physical elements. We may now define how basis functions for physical elements and their derivatives are evaluated. Let $N_a(\mathbf{x})$ be basis functions of the physical element. We will give values to the physical basis functions via the following relationship

$$N_a(\mathbf{x}) = (\hat{N}_a \circ \mathcal{G})(\mathbf{x}). \tag{4.4}$$

It follows from this that

$$\nabla N_a(\mathbf{x}) = \nabla(\hat{N}_a \circ \mathcal{G})(\mathbf{x}). \tag{4.5}$$

Next, consider that it is typical to deal with integrands involving functions such as $N_a(\mathbf{x})$ or $\nabla N_a(\mathbf{x})$. Since we don't actually know the form of $N_a(\mathbf{x})$, we have to make use of the relationships shown in Equations (4.4) and (4.5). For instance, because of Equations (4.2) and (4.4)

$$(N_a \circ \mathcal{F})(\boldsymbol{\xi}) = \hat{N}_a(\boldsymbol{\xi}). \tag{4.6}$$

Furthermore, we may write, using Equation (4.5)

$$\nabla N_a(\mathbf{x}) = \nabla(\hat{N}_a \circ \mathcal{G})(\mathbf{x}) = \sum_{i=1}^{3}(\hat{N}_{a,i} \circ \mathcal{G})(\mathbf{x})\nabla \mathcal{G}_i(\mathbf{x}). \tag{4.7}$$

We may therefore substitute (4.3) into (4.7) and use Equation (4.2) to write the following useful equality which allows us to evaluate integrals involving functions of the type $\nabla N_a(\mathbf{x})$

$$(\nabla N_a \circ \mathcal{F})(\boldsymbol{\xi}) = \sum_{(i,j,k)+} \hat{N}_{a,i}(\boldsymbol{\xi})(V_j(\boldsymbol{\xi}) \times V_k(\boldsymbol{\xi}))\frac{1}{|J_{\mathcal{F}}|} \tag{4.8}$$

where the sum over $(i,j,k)+$ is taken over cyclic permutations of i,j,k through the numbers $1,2,3$. For an explicit form of Equation (4.8), see [6].

4.4. Assembly. We now turn to a few aspects regarding the actual assembly of the stiffness matrix and load vector. We will address numerical quadrature and the use of data arrays to create the assembly of the matrices.

4.4.1. Numerical quadrature. In general, numerical quadrature rules have the following form

$$\int_K f(\mathbf{x})dx \approx \sum_{i=1}^{n} w_i f(\mathbf{x}_i) \tag{4.9}$$

where the $\mathbf{x}_i$ are called **quadrature points** and w_i are called **weights**. It is well-known that if function f meets certain criteria which depend on the way in which the weights and quadrature points are chosen, then we may replace $\approx$ by $=$ in Equation (4.9). See [6] for a discussion of this.

Though for the type of polynomials which we will evaluate in the current code, 8-point Gaussian quadrature should be enough to evaluate them exactly, the presence of $\frac{1}{|J_{\mathcal{F}}|}$ in the integrals which we evaluate seems to throw a wrench in the works when trying to evaluate integrals with triquadratic basis functions. We suspect that this is because the term $\frac{1}{|J_{\mathcal{F}}|}$ is not constant and not a polynomial. Whatever the case, when attempting to use 8-point Gaussian quadrature, the stiffness matrices which result are singular. When one switches to 27-point Gaussian quadrature, the difficulty is alleviated and the stiffness matrix is invertible. We have been able to track down a reference [5] which seems to shed some light on why this works. Though we have not yet worked through it, there is a section which indicates that it should be no surprise that the evaluation of an integral with triquadratic basis functions using 8-point quadrature yields a singular matrix. The document suggests the use of 27-point quadrature in this case. It would be interesting to explore this a bit further and determine if there is any other type of quadrature rule that is cheaper to implement, yet produces invertible matrices.

4.4.2. Data arrays. The data arrays which we use in the assembly of the stiffness matrix and load vector are those suggsted by Hughes. They are the IEN, ID, and LM arrays. See [6] for detailed information on this.

5. Results. In order to test the code at the intermediate step previously indicated, we must formulate problems with known solutions to which we can compare our results. First, we will consider the problem involving the electrostatic potential. Then we will consider a problem

6. Continued Work. The next stage of work will be to couple the Stokes equations with the Laplace equation in the case that the sphere is present. The plan is to do this in a framework where no meshing relative to the sphere is done. In that

sense, we hope to introduce the sphere on top of the mesh already implemented in the work we have completed up to the present. From this point, we will be able to make comparisons of our solution to the solution obtained in [9] via some values computed from the solution. From here we hope to animate the motion of the sphere. We still need to explore methods for including the sphere and enforcing the boundary conditions it presents without remeshing around it.

In the stage following this, we would use our existing solvers together with solvers for the equations we have negelected so far in order to solve the full set of equations describing the electrokinetic flow. In this sense, we should at this point be able to simulate the motion of the particle in a tube assumed to be on the order of nanoscales. As a part of this stage, it will be important to investigate the validity of the continuum models at the nanoscale.

7. Conclusions. In this paper, we have introduced the theory describing electrokinetic flows. Employing some assumptions, we have seen how the system of equations can be simplified to a form that admits analytical solutions for certain geometries. Our hope is to implement a code to produce numerical results that we would be able to compare to one of these analytical solutions. Yariv and Brenner [8] [9] have produced solutions for the case that the geometry is a sphere inside a cylinder. As a first step toward a comparison with this result, we have developed a solver for the coupled (through boundary conditions) Laplace-Stokes system in the case that the sphere is not present. In order to test the Laplace and Stokes solvers we have implemented in Matlab, we have shown comparisons to known results. Finally, we have outlined what the next stages of work will consist of.

REFERENCES

[1] Pavel Bochev and Allen C. Robinson, *Exact sequences of finite elements on hexahedral and quadrilateral lattices.* Sandia National Laboratories.

[2] S. Devasenathipathy and J.G. Santiago, *Electrokinetic flow diagnostics*, in Micro- and Nano-Scale Diagnostic Techniques, K.S. Breuer, ed., Springer-Verlag, 2003.

[3] George Karniadakis et al., *Microflows and Nanoflows: Fundamentals and Simulation*, Springer, 2005.

[4] Kendra V. Sharp et al., *Liquid flows in microchannels*, in The MEMS Handbook, Mohamed Gal-El-Hak, ed., CRC Press, 2002.

[5] Carlos Felippa, *N/a.* Notes for a class taught by Professor Felippa, website http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/.

[6] Thomas J.R. Hughes, *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*, Dover, 2000.

[7] H.J. Keh and J.L. Anderson, *Boundary effects on electrophoretic motion of colloidal spheres*, J. Fluid Mech., 153 (1985), pp. 417–439.

[8] Ehud Yariv and Howard Brenner, *The electrophoretic mobility of an eccentrically positioned spherical particle in a cylindrical pore*, Phys. Fluids, 14 (2002), pp. 3354–3357.

[9] ———, *The electrophoretic mobility of a closely fitting sphere in a cylindrical pore*, SIAM J. Appl. Math., 64 (2003), pp. 423–441.

EFFECTS OF INTERFACIAL INTERACTIONS ON ADHESION OF NANOSCALE GOLD FILMS ON SILICON SUBSTRATES

M.S. KENNEDY* AND N.R. MOODY†

Abstract. The reliability of thin films used in nanostructured devices is dependent on the interfacial fracture energies between the film and substrate. This interfacial energy imposes limitations on stresses in the device since when the interfacial fracture energy is exceeded, delamination occurs. This study examined the effects of chemistry changes that occurred during processing and subsequent service on film adhesion. In particular, we examined the impact of changes in chemical bonding along an interface on the interfacial fracture energy. During this project, experimental measurements of adhesion were coupled with analytical solutions to determine interfacial fracture toughness of Au/SiO_2. This film system was picked due to its utilization in Sandias MEMS devices, such as MEMS mirrors, and also microelectronics systems.

1. Introduction. Many nanostructured materials are currently integrated into devices utilized by Sandia and its consumers. These materials are also part of emerging nanotechnologies with applications in medicine, security, and defense. These nanostructured materials include metals, polymers and ceramics and are often fabricated into thin film structures. Fracture properties are of high importance in small-scale systems, and currently few experimental testing methods are available to measure deformation and fracture phenomena in these systems. Some of the more recent techniques being utilized include spontaneous blisters, stressed overlayers [2, 6–9, 15], nanoindentation induced delamination [2], scratch testing [10,12,13,15] and four point bending [1, 3, 4]. These methods induce delamination at the interface and from which the adhesion energy or the crack growth rate can be determined. Idealized interfaces have uniform dimension, chemistries, and depth, but are typically seen only after the initial stages of fabrication. In devices where processing includes high temperatures or large strains, the interfaces may undergo transformation due to diffusion. In addition, interfaces have been known to change during device service. The effects of these changes in composition on the film functionality, mechanical response and adhesion, are currently not known.

One film system that is currently employed in MEMS mirrors and nanoscale interconnects is the Au/SiO_2 system. Recent work on thin Au films (100-200 nm) has shown that the interface of this system is not stable during processing and service. When these films were deposited with higher thicknesses, this change in composition was not a significant concern because the diffusion rate limited the composition changes to small regimes. For thinner films, the interactions lead to a complete film composition change. This study looks at the property changes of the interfacial adhesion and mechanical response of Au/SiO_2 with both experimentation and finite element modeling. The interface compositions were studied using SAM and the surface topography, such as roughness, were measured using AFM. The films moduli were then measured using nanoindentation and adhesion energies by stressed overlayers and four point bending. In addition, a finite element based indentation model was used to define film and substrate contributions on elastic and plastic properties. By looking at as deposited, annealed, and changes with long service times, this study aims to bring clarity to aging kinetics of this interface.

2. Background.

*Washington State University
†Sandia National Laboratories

2.1. Adhesion and Interfacial Fracture Energy. To understand the importance of interfacial fracture energy of a solid-solid interface, a working definition of adhesion needs to be specified. Adhesion is the energy per unit area required to separate a film from its substrate along the dividing interface. This energy is delineated into two parts: the true work of adhesion and the inelastic contributions that occur during the separation process [5, 14].

The true work of adhesion, Γ_A, is the thermodynamic work required to create two new surfaces at the expense of the interface and is an intrinsic property of a given system property that depends on the type of chemical bonding between the film and substrate. Γ_A is calculated from

$$\Gamma_A = \gamma_f + \gamma_s - \gamma_{fs} \tag{2.1}$$

where γ_f is the surface energy of the film, γ_s is the surface energy of the substrate and γ_{fs} is interfacial energy of the two materials in contact. This true work of adhesion can only be measured directly for liquid-solid interfaces where the contact angle between the surface and substrate can be measured. The work of adhesion can be expressed by Young-Dupr as

$$\Gamma_A = \gamma_f \left(1 + \cos\Theta\right) \tag{2.2}$$

where Θ is the angle between the film and substrate as shown in Figure 2.1.

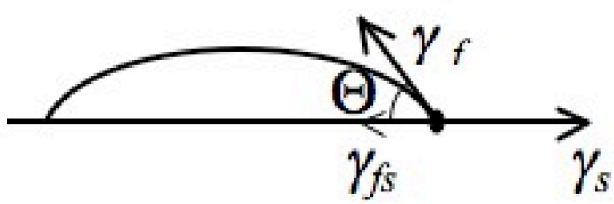


FIG. 2.1. *Contact angle measurement to determine the adhesion between a liquid film and solid substrate. This angle can be used with the Young-Dupr equation to determine the adhesion energy.*

In most practical cases of mechanical de-adhesion, there is additional inelastic damage, including plasticity and micro-cracking occurring in the regions of the substrate and film near the interface. This additional inelastic contribution is taken into account as follows

$$\Gamma_P = \Gamma_A + \Gamma_{inelastic} \tag{2.3}$$

where Γ_P is the total fracture energy, Γ_A is the true adhesion energy and $\Gamma_{inelastic}$ is the energy per unit area required for the inelastic deformation. Although it is impossible to completely limit inelastic contributions, careful consideration of test methods will approach the calculated true work of adhesion.

2.2. LEFM models and current experimental techniques. Recently developed quantitative methods are based on linear elastic fracture mechanics, LEFM. These quantitative methods compare the applied strain energy release rate, G, and the resistance to crack propagation (the interfacial fracture toughness, Γ). Delamination will only occur when the strain energy release rate is greater than the interfacial

fracture toughness. Measurement of Γ is achieved by increasing the strain energy release rate to or above a critical point where the crack starts to propagate. Then the critical rate for fracture, G, is often assessed when the delaminating crack arrests.

The film delamination is modeled as a bimaterial crack and the adhesion energy is calculated as the strain energy release per unit increase in delamination area. Mechanics based models [1, 11] made it possible to calculate the adhesion energy using the buckled film morphologies after delamination. This method, however, depended on the films being compressively stressed and needing only the energy stored within the film to overcome the interfacial adhesion energy.

As previously stated, four quantitative techniques used to measure adhesion include stressed overlayers, indentation induced delamination, scratch testing and four point bending. Of these four test techniques, our study has utilized two: stressed overlayers and indentation induced delamination. A third method, four point bending, is being developed. Below is a brief description of these techniques.

The measurement of spontaneous buckles to determine adhesion energy has been summarized by Hutchinson and Suo [5]. This type of measurement is only applicable when a compressive film has enough residual stress in order to over come the interfacial fracture toughness. Various models have been proposed to determine the energy from a variety of buckle shapes including Euler buckles (long, straight), telephone cord (wavy), and circular blisters. In models for each of the morphologies, the process is similar and so only the steps for an Euler buckle will be described here. The buckling stress, the amount of stress needed to delaminate the film, must be less than or equal to the residual stress of the film. The buckling stress of a straight sided blister, σ_b, can be described by

$$\sigma_b = \frac{\pi^2 E}{12\left(1-\nu^2\right)}\left(\frac{h}{b}\right)^2 \tag{2.4}$$

where h is the thickness of the film, b is the radius of the induced blister, E is the elastic modulus of the film and ν is Poissons ratio for the film. Since film delamination has already occurred, the driving stress, σ_d, is known to be greater than, and proportional to, the buckling stress. This driving stress is estimated as

$$\sigma_d = \sigma_b\left[\frac{3}{4}\left(\frac{\delta}{h}\right)^2 + 1\right] \tag{2.5}$$

where δ is the height of the film delamination and h is the thickness of the film. The mixed mode interfacial fracture energy, $\Gamma(\Psi)$, is

$$\Gamma(\Psi) = \frac{\left(1-\nu^2\right)h}{2E}\left(\sigma_b - \sigma_d\right)\left(\sigma_b + 3\sigma_d\right) \tag{2.6}$$

where all the constants are the same as described in equations (2.4) and (2.5). Conversions to mode I adhesion values are obtained by first calculating the phase angle of loading. This loading angle is calculated by

$$\Psi = \tan^{-1}\left(\frac{\sqrt{3}\xi\sin\omega + 4\cos\omega}{\sqrt{3}\xi\cos\omega - 4\sin\omega}\right) \tag{2.7}$$

where ω is 51.2° and

$$\xi = \left[\frac{1}{c_1}\left(\frac{\sigma_b}{\sigma_d} - 1\right)\right]^{1/2}. \tag{2.8}$$

c_1 is a constant dependant on Poissons ratio of the film and is given as

$$c_1 = 0.2473\,(1+\nu) + 0.2231\left(1-\nu^2\right) \tag{2.9}$$

The mode I adhesion energy is simply a function of the mixed mode value, the phase angle,

$$\Gamma_I = \frac{\Gamma\,(\Psi)}{\left[1+\tan^2\left((1-\lambda)\,\Psi\right)\right]} \tag{2.10}$$

where λ is often assumed to be a constant, 0.3.

Films may fail under a variety of stresses ranging from all normal stresses to a combination of shear and normal stresses to pure shear stresses. Each test technique measures the adhesion energy corresponding to a different ratio of shear to normal stresses and is described by the phase angle of loading or mode mixity, Ψ. This ratio of stresses influences the magnitude of energy dissipation needed for fracture. When failure occurs due to mode I, opening mode loading, the interfacial fracture toughness is equal to the thermodynamic work of adhesion. Since the work of adhesion involves only normal forces, adhesion values are often reported as both as a function of the experimental mode mixity and mode I value. An estimate of the mode I value can be made once the phase angle of loading is determined. The Ψ is a ratio of the mode II and mode I stress intensity as shown by

$$\Psi = \tan^{-1}\left(\frac{K_{II}}{K_I}\right) \tag{2.11}$$

where K_{II} and K_I are the stress intensity factors for mode II and mode I respectively. The relationship between the phase angle and fracture energy is shown in figure 2.2.

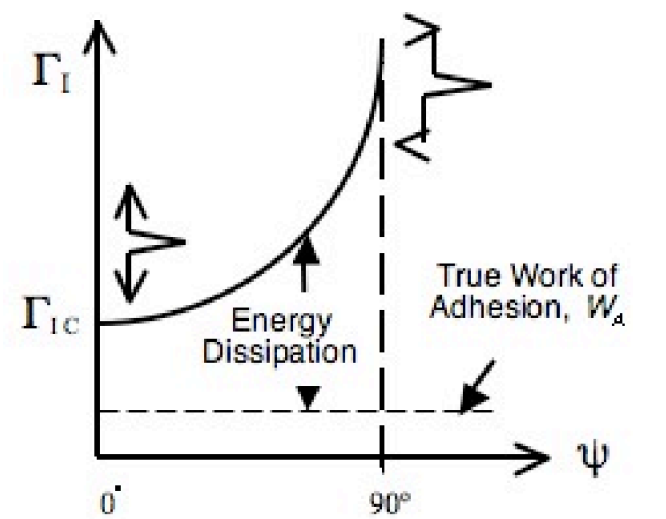


Fig. 2.2. *This shows the relationship between the mode mixity and the interfacial adhesion energy.*

The simplest method to apply energy if the system has not delaminated is by the use of stressed overlayers where an additional film with a uniform compressive stress is applied to an underlying ductile layer to add energy. This method enables buckle

morphologies to be used to determine the adhesion energy for films under a tensile stress. Like spontaneous buckles, it is the ratio of the driving stress to the buckling stress. The buckling stress of a multilayer film was first modeled by Kriese et al [8]. This group showed that stress for delamination, σ_b, was proportional to the moment of Inertia, I_T, by

$$\sigma_b = \frac{\mu^2}{ahb^2}\left[\frac{E}{(1-\nu^2)}\right] I_T \tag{2.12}$$

where h is the thickness of the film of interest and the overlayer film, b is the radius of the induced blister, E is the elastic modulus of the film, ν is Poissons ratio for the film, I_T is the moment of inertia and a is a geometric constant. The geometric constant drops out when evaluated with I_T.

With indentation-induced delamination, a spherical-tipped conical indenter is driven perpendicularly into the film to add mechanical energy. When modeling the indentation events, the systems are simplified. It is assumed that the interfacial crack is directly under the indenter tip. As the film is indented, there is radial expansion of the film and the film is buckled either during or after the indentation test. The total strain energy to create a blister has been modeled by assuming the fracture is a clamped circular plate, where the indent is kept within the film. Like the stressed overlayer method, the fracture energy, $\Gamma(\Psi)$, takes into account the critical buckling stress, σ_b, the driving stress, σ_d, and the indentation stress, σ_I. The indentation stress can be calculated with the following equation:

$$\sigma_I = \frac{V_I E}{2\pi h b^2 (1-\nu)}, \tag{2.13}$$

where V_I is the volume of the residual indentation. In the case of the conical indenter, the volume can be estimated to be more similar to a semicircle than a cone. The interfacial fracture energy, $\Gamma(\Psi)$, is then given by

$$\Gamma(\Psi) = \frac{h\sigma_I^2\left(1-\nu^2\right)}{2E} + (1-\alpha)\frac{h(1-\nu)}{E}\left\{\sigma_d^2 + (\sigma_I - \sigma_b)^2\right\}, \tag{2.14}$$

where

$$\alpha = 1 - \frac{1}{1+0.902(1-\nu)}. \tag{2.15}$$

Another way of applying energy is through scratch testing; a method that applies both lateral as well as the normal forces with indentation. The test consists of drawing a stylus of an indenter with a known radius of curvature over a film or coating under increasing vertical loads. Resultant scratches are then observed under an optical or scanning electron microscope to measure fracture events corresponding to loading and measure the area of delamination.

Another LEFM based test technique for interfacial adhesion energy is the four point bend test. In this test, the interface of interest is bonded between two elastic substrates. The sample is loaded as shown in Figure 2.3 to propagate the crack along the weakest interface. Unlike spontaneous, stressed overlayer or indentation induced delamination, the mode mixity of the loading is constant, 45°. This is an attractive technique since the loading mode mixity represents an almost equal amount of shear to normal stresses. It also is not dependent on an unknown precrack length

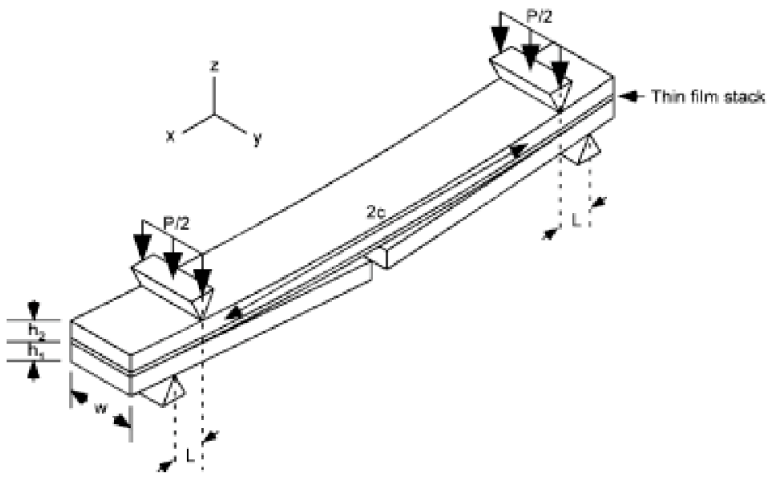


FIG. 2.3. *Four point bend specimen with a crack length of 2c [4].*

as observed in spontaneous buckling, indentation induced buckling or scratch testing. During testing, the load is measured and the displacement rate is held constant. By studying the plateau in load versus displacement, the critical strain energy release rate can be determined. The initial load-displacement curve shows the composite loading elastically, storing strain energy in the elastic substrates. At a critical load, the stain energy release rate exceeds the crack resistance of the weakest interface and there is a load plateau. A typical load-displacement graph is shown in Figure 2.4.

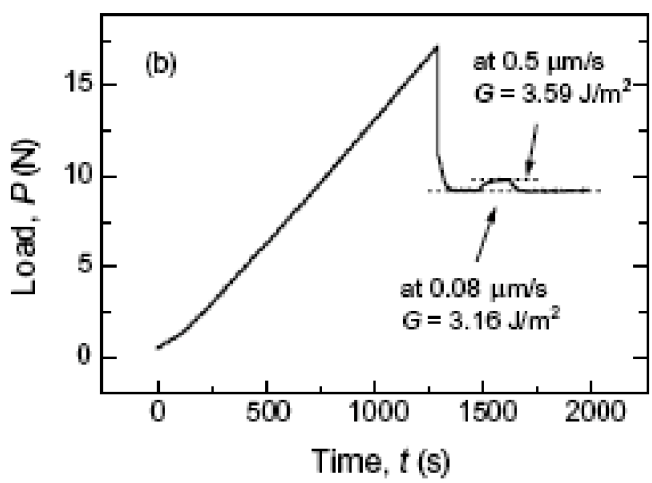


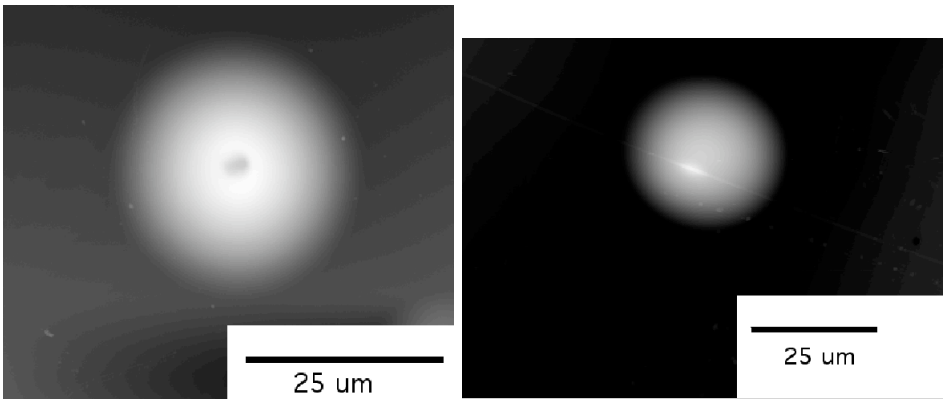
FIG. 2.4. *A load displacement curve showing typical trends during a four point bend test [3].*

The interface fracture energy is a function of the critical value of the applied strain energy release rate (G) as follows:

$$G = \frac{21}{4} \frac{\left(1 - \nu^2\right) M^2}{E b^2 h^3} \tag{2.16}$$

where the bending moment $M = Pl/2$, P is the load, l is the spacing between the inner and the outer loading lines, b is the beam width, h is the half thickness, and E and ν are the elastic modulus and Poisson's ratio of the bulk substrate respectively.

3. Results. The adhesion energy of the as deposited Au/SiO_2 was determined by both spontaneous and indentation induced delamination. A 250 nm compressive W layer was deposited onto the 150 nm Au / 10 nm SiO_2. Figures 3.1(a) and 3.1(b) are of a spontaneous buckle and indentation induced buckle, respectively. These AFM



(a) Spontaneous blister (b) Indentation induced blister

FIG. 3.1. *AFM images of a spontaneous blister, and an indentation induced blister in the W/Au/SiO$_2$ system.*

images, and the profiles shown in Figure 3.2, showed that these blisters had no radial cracking or pile-up. Calculations of the fracture energies were followed from work

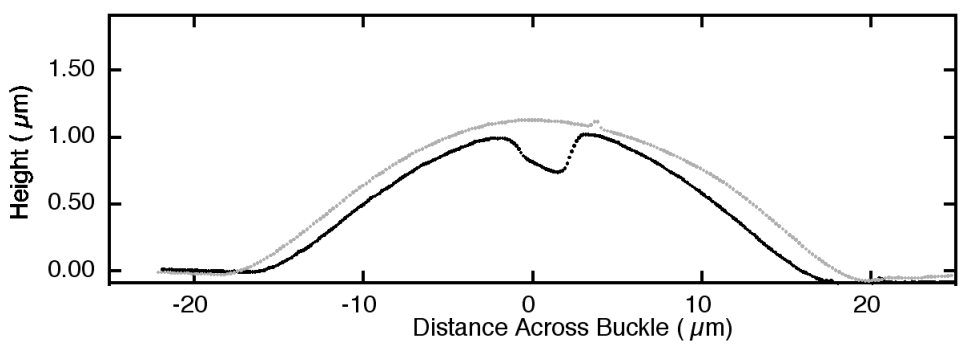
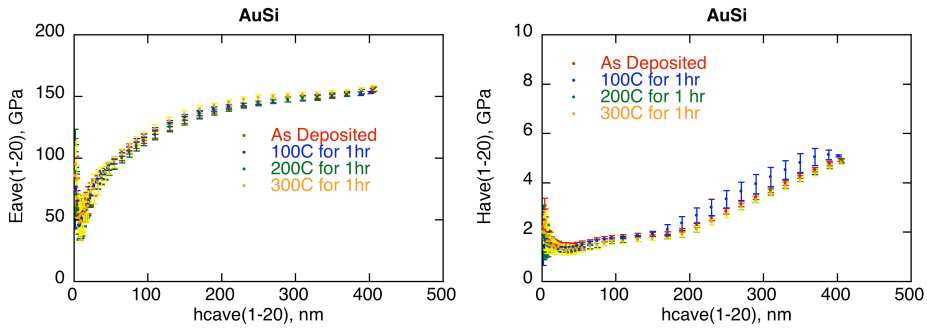


FIG. 3.2. *Indentation profiles of the spontaneous and indentation blisters shown in figures 3.1(a) and 3.1(b).*

done by Kriese with one modification. The residual stresses in the films were not measured but were considered equivalent to the calculated driving stress from the spontaneous buckles [1]. The calculated interfacial adhesion energies, 0.10 J/m^2 $\pm$ 0.03 for the spontaneous buckles and 0.09 J/m^2 $\pm$ 0.05 for the indentation induced blisters, were practically identical confirming that maintaining close approximations of model limitations makes calculations more exact.

To study aging, the Au films were placed into a furnace and heated at 100 and 300°C for 1 hr. The mechanical properties of these films were then determined by the continuous stiffness method, using a frequency of 45 Hz and amplitude of 2 nm. Figures 3.3(a) and 3.3(b) show modulus and hardness as a function of depth for all the aged samples. This data indicates that the properties do not significantly change



(a) Elastic modulus (b) Hardness

FIG. 3.3. *Elastic modulus and hardness as functions of depth for as deposited and aged samples of Au/SiO_2.*

during the aging process and the initial values can be used in all fracture toughness calculations.

Au/SiO_2 samples aged for 1 hr at 100°C showed little visible change in film morphology, figure 3.4. Auger scans of the interface, figure 3.5, also showed little

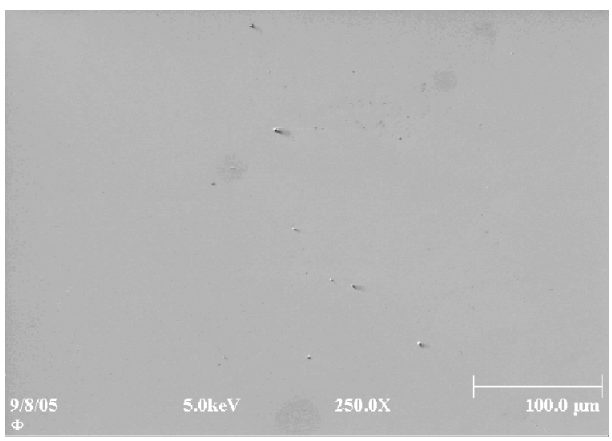


FIG. 3.4. *An SEM image of the surface of the Au after accelerated aging. This sample was heated in air at 100° C for 1 hr.*

diffusion of the layers. However, the interfacial fracture energies were significantly larger than the as deposited, 1.2 J/m^2 compared to .1 J/m^2. Figures 3.6(a) and 3.6(b) show the delamination buckles and some of the measured adhesion values.

Au/SiO_2 samples aged for 1 hr at 300°C showed a roughed surface, figure 3.7. The measured RMS values went from 2nm for an as deposited film to 82 nm for an aged film. Auger scans of the interface, Figure 3.8, also showed significant diffusion between the substrate and corresponding film layers. These interfacial fracture energies were

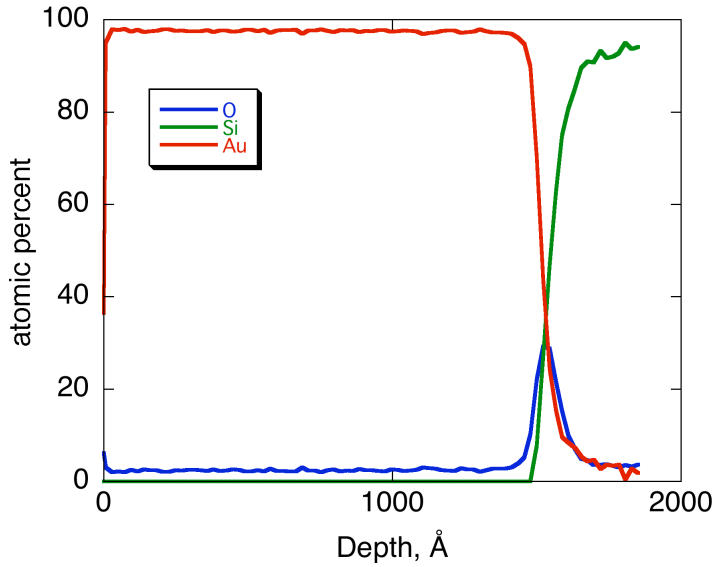
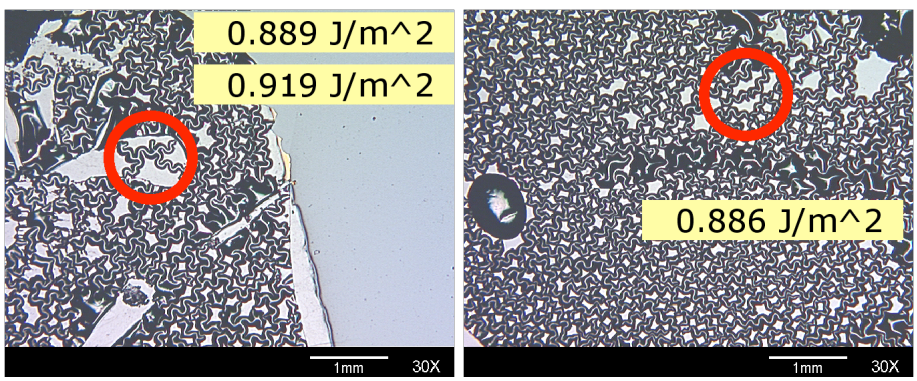


FIG. 3.5. *An Auger scan of the Au interface after heat-treating at 100° C for 1 hr.*



(a) (b)

FIG. 3.6. *These optical images are of the accelerated aging of the Au/SiO_2 interface. This interface was heated in air at 100° C for 1 hr and then a compressive W layer was deposited on top. The interfacial energies are shown next to their respective regions.*

significantly larger than the as deposited and those films aged at 100°C. Energies emerging from these delamination buckles, figures 3.9(a) and 3.9(b), were as high as 9.9 J/m^2.

In addition to measuring accelerated aging, we also measured the adhesion energy of a 200 nm Au film on SiO_2 that was originally deposited in 1998 by Alex Volinsky. On this Au sample, we deposited 220 nm of compressive W. Spontaneous blisters formed near the center of the sample, as shown in figure 3.10. From these delaminations, we measured the mode I interfacial fracture energy to range between 1.2-1.9 J/m^2. This was higher than the originally adhesion energy, which was determined to be .6 J/m^2 in 1998.

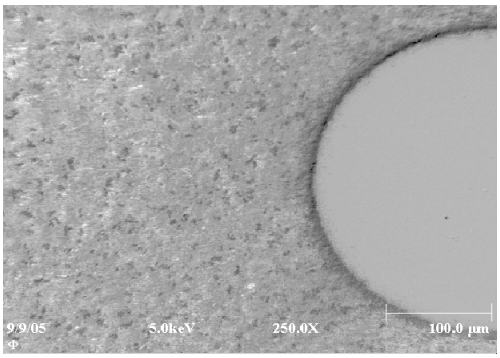


FIG. 3.7. *A SEM image of the surface of the Au after accelerated aging. This sample was heated in air at 300° C for 1 hr. A rough silicide has formed on the surface.*

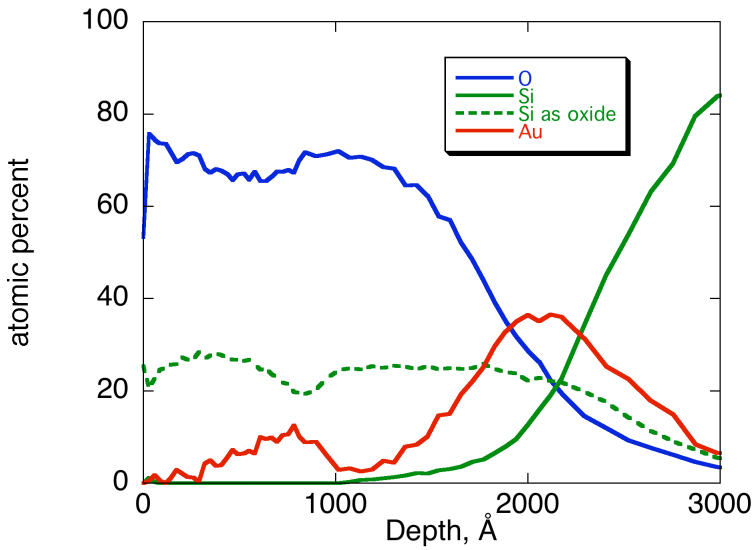


FIG. 3.8. *An Auger scan of the Au interface after heat-treating at 300° C for 1 hr. This shows that significant movement of the Au and Si and the complete alteration of the films chemistry and interface boundary.*

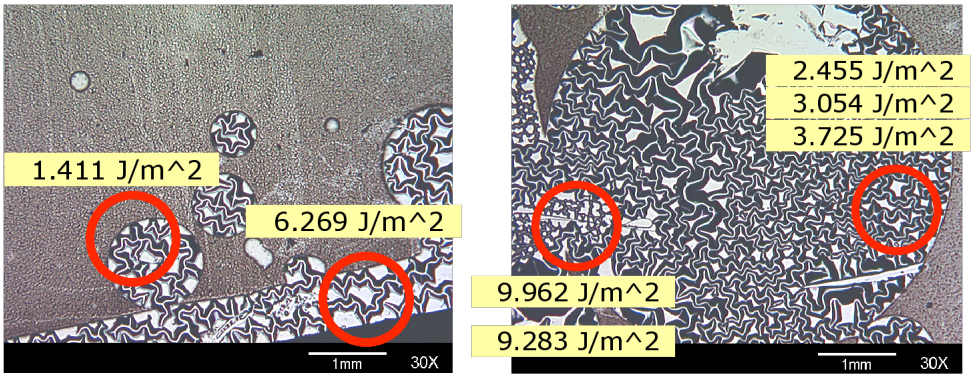
4. Conclusion. Systematic aging showed that limited diffusion increased the adhesion energy of Au/SiO_2 interfaces. That is, when diffusion of only Au and Si is allowed along the interface, the interface will strengthen over time. The addition of water or other contaminates along the interface will weaken the interface over time.

REFERENCES

[1] N. BARBOSA, R.S. RIDLEY, C.H. STRATE, R.S. DWYER, T. GREBS, AND R.P. VINCI, *Metal adhesion to less than or equal 100¿ si substrates with varying surface conditions.*, vol. 695, Materials Research Society, 2002, pp. L7.9.1–5.

[2] M.J. CORDILL, D.F. BAHR, N.R. MOODY, AND W.W. GERBERICH, *Recent developments in thin film adhesion measurement.*, IEEE Trans. Device and Mater. Rel., 42 (2004), pp. 163–168.

[3] R.H. DAUSKARDT, M. LANE, Q. MA, AND N. KRISHNA, *Adhesion and debonding of multi-layer thin film structures.*, Eng. Frac. Mech., 61 (1998), pp. 141–162.



(a) Adhesion energies on the edge of the wafer and image

(b) Adhesion energies for the central regions

FIG. 3.9. *These optical images are of the accelerated aging of the Au/SiO_2 interface. This interface was heated in air at 300° C for 1 hr and then a compressive W layer was deposited on top. The interfacial energies are shown next to their respective regions.*

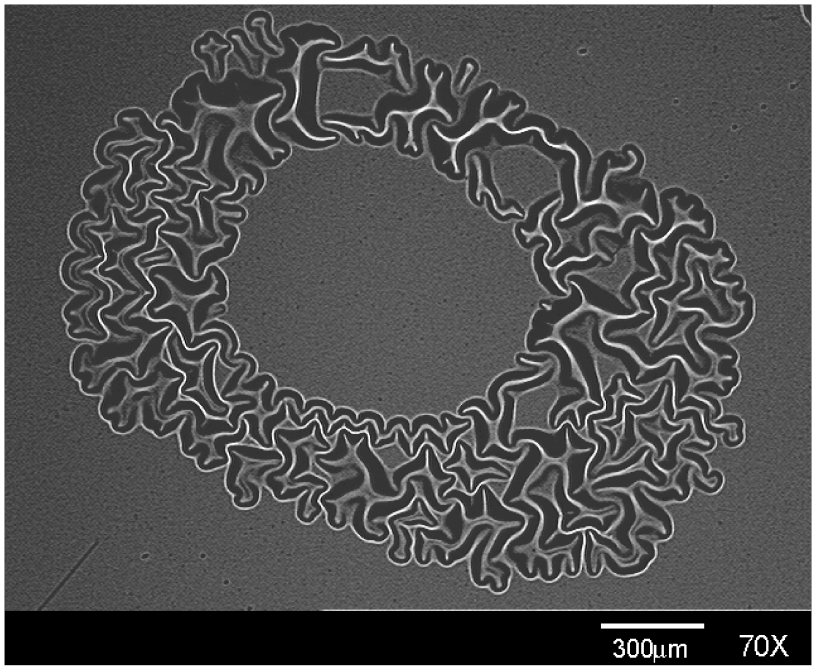


FIG. 3.10. *Spontaneous blisters formed from fracture of the aged Au/SiO_2 interface after 220 nm of compressive W were deposited. The measured adhesion energy was significantly higher than the as deposited interface.*

[4] M.P. Hughey, D.J. Morris, R.F. Cook, S.P. Bozeman, B.L. Kelly, A.L.N. Chakravarty, D.P. Harkens, and L.C. Sterns, *Four-point bend adhesion measurements of copper and permalloy systems.*, Eng. Frac. Mech., 71 (2004), pp. 245–261.

[5] J.W. Hutchinson and Z. Suo, *Mixed mode cracking in layer materials.*, Adv. Appl. Mech., 29 (1992), pp. 63–191.

[6] U. Kang, T. Lee, and Y.-H. Kim, *Pt-ti thin film adhesion on sinx/si substrates.*, Jpn. J. Appl. Phys., 38 (1999), pp. 4147–4151.

[7] M.D. Kriese, D.A. Boismier, N.R. Moody, and W.W. Gerberich, *Nanomechanical fracture testing of thin films.*, Eng. Frac. Mech., 61 (1998), pp. 1–20.

[8] M.D. Kriese, N.R. Moody, and W.W. Gerberich, *Effects of annealing and interlayers on the adhesion energy of copper thin films to sio_2/si substrates.*, Acta Mater., 46 (1998), pp. 6623–6630.

[9] A. Lee, C.S. Littenken, R.H. Dauskardt, and W.D. Nix, *Comparison of the telephone cord delamination method for measuring interfacial adhesion with the four-point bending method.*, Acta Mater., 53 (2005), pp. 609–616.

[10] N.R. Moody, D. Medlin, D. Boehme, and D.P. Norwood, *Film thickness effects on the fracture of tantalum nitride on aluminum nitride thin film systems.*, Eng. Frac. Mech., 61 (1998), pp. 107–118.

[11] M.W. Moon, H.M. Jensen, J.W. Hutchinson, K.H. Oh, and A.G. Evans, *The characterization of telephone cord buckling of compresses thin films on substrates.*, J. Mech. Phys. Solids, 50 (2002), pp. 2355–2377.

[12] H. Ollendorf and D. Schneider, *A comparative study of adhesion test methods for hard coatings.*, Surf. & Coatings Tech., 113 (1999), pp. 86–102.

[13] M. Toparli and S. Sasaki, *Evaluation of the adhesion of tin films using nanoindentation and scratch testing.*, Philos. Mag. A, 82 (2002), pp. 2191–2197.

[14] A.A. Volinsky, D.F. Bahr, M.D. Kriese, N.R. Moody, and W.W. Gerberich, vol. 8.13, Elsevier Pergamon, Amsterdam, 1st ed., 2003, pp. 453–493.

[15] A.A. Volinsky, N.R. Moody, and W.W. Gerberich, *Interfacial toughness measurements for thin films on substrates.*, Acta Mater., 50 (2002), pp. 441–466.

THE IMMERSED FINITE ELEMENT METHOD FOR INCOMPRESSIBLE FLOW COMPUTATIONS

A. M. KOPACZ*, T. D. NGUYEN†, AND G. J. WAGNER‡

Abstract. The immersed finite element method (IFEM) is utilized to solve complex fluid and deformable structure interaction problems. In IFEM, a Lagrangian solid mesh moves on top of a background Eularian fluid mesh which spans over the entire domain. This technique evicts the need of mesh update thus saving computation time. Both the fluid and the solid domains are modelled with the finite element method. The continuity between the fluid and solid sub-domains are enforced via the interpolation of the velocities and the distribution of forces employing the reproducing kernel particle method (RKPM) delta function. The use of such kernel functions permits non-uniform fluid spatial meshes with arbitrary geometries and boundary conditions. The goal of this work is to implement the immersed finite element method within TAHOE's infrastructure. Much of the effort this summer has been put forth implementing the stabilized formulation of the Navier-Stokes equations solver. The stabilized formulations of our main focus is the streamline-upwind/Petrov-Galerkin (SUPG) and the pressure-stabilizing/Petrov-Galerkin (PSPG) method. These formulations are applied to the nonlinear Navier-Stokes equations for incompressible flows. Currently, my collaborators and I are verifying the correctness of the implemented fluid algorithm. Future work necessary to fulfill the objective will also be discussed.

1. Immersed Finite Element Method.

1.1. Introduction. The Immersed Finite Element Method (IFEM) was developed to treat fluid-structure interaction problem more effectively and more efficiently. The following formulation is taken from Zhang et al. [7].

1.2. Equations of Motion. In this section, we start by introducing the equations of motion for a continuum that contains both the fluid and solid. The governing equations of both domains are then extracted from the equations of motion along with a reproducing kernel particle method (RKPM) delta function for enforcing the continuity between the fluid and solid sub-domains.

In a continuum, the inertial force of a particle is balanced with the derivative of the Cauchy stress σ_{ij} and the external force f_i^{ext} exerted on the continuum. In Einstein notation we have:

$$\rho \frac{dv_i}{dt} = \sigma_{ij,j} + f_i^{ext}. \tag{1.1}$$

If the solid density ρ^s is different from the fluid density ρ^f, we can divide the inertial force into two components within the solid domain Ω^s:

$$\rho \frac{dv_i}{dt} = \begin{cases} \rho^f \frac{dv_i}{dt}, & \mathbf{x} \in \mathbf{\Omega}/\mathbf{\Omega^s} \\ \rho^f \frac{dv_i}{dt} + (\rho^s - \rho^f)\frac{dv_i}{dt}, & \mathbf{x} \in \mathbf{\Omega^s}. \end{cases} \tag{1.2}$$

Moreover, the external force f_i^{ext} is illustrated as the gravitational force and since the computational fluid domain is the entire domain Ω, we ignore the hydrostatic pressure and obtain:

$$f_i^{ext} = \begin{cases} 0, & \mathbf{x} \in \mathbf{\Omega}/\mathbf{\Omega^s}, \\ (\rho^s - \rho^f) g_i, & \mathbf{x} \in \mathbf{\Omega^s}. \end{cases} \tag{1.3}$$

*Northwestern University, a-kopacz@northwestern.edu
†Sandia National Laboratories, tdnguye@sandia.gov
‡Sandia National Laboratories, gjwagne@sandia.gov

Furthermore, the derivative of the Cauchy stress can also be decomposed as:

$$\sigma_{ij,j} = \begin{cases} \sigma^f_{ij,j}, & \mathbf{x} \in \mathbf{\Omega}/\mathbf{\Omega^s}, \\ \sigma^f_{ij,j} + \sigma^s_{ij,j} - \sigma^f_{ij,j}, & \mathbf{x} \in \mathbf{\Omega^s}. \end{cases} \tag{1.4}$$

Note in former equations, the fluid domain Ω^f is represented as the entire domain Ω minus the solid domain Ω^s, namely, Ω/Ω^s. The fluid-structure interaction force(FSI) within the domain Ω^s as $f_i^{FSI,s}$:

$$f_i^{FSI,s} = -(\rho^s - \rho^f)\frac{dv_i}{dt} + \sigma^s_{ij,j} - \sigma^f_{ij,j} + (\rho^s - \rho^f)g_i, \quad \mathbf{x} \in \mathbf{\Omega^s}. \tag{1.5}$$

RKPM delta function is used to distribute the interaction force from the solid domain onto the computational fluid domain. Hence, the governing equation for the fluid can be derived by combining the fluid terms and the interaction force as:

$$\rho^f \frac{dv_i}{dt} = \sigma^f_{ij,j} + f_i^{FSI}, \quad \mathbf{x} \in \mathbf{\Omega}. \tag{1.6}$$

Since we consider the entire domain Ω to be incompressible, we only need to apply the incompressibility constraint once in the entire domain Ω:

$$v_{i,i} = 0, \quad \mathbf{x} \in \mathbf{\Omega}. \tag{1.7}$$

The coupling of the fluid and solid velocity fields is also accomplished through the RKPM delta function.

2. Fluid Dynamics.

2.1. Introduction. In this section, we discuss in detail the 3D finite element fluid solver used in the IFEM formulation that has been implemented within TAHOE's infrastructure. The derivations include the strong form, namely, the Navier-Stokes equations for the incompressible Newtonian fluid with constant density and viscosity, the streamline-upwind/Petrov-Galerkin (SUPG) and the pressure-stabilizing/Petrov-Galerkin (PSPG) weak form and the corresponding discretization. Furthermore, the mixed integration scheme implemented in TAHOE is also detailed.

2.2. Navier-Stokes Strong Form. Consider an incompressible Newtonian fluid satisfying Navier-Stokes equations. The conservation of mass and the conservation of momentum in the domain Ω bounded by Γ is given as:

$$v_{i,i} = 0 \tag{2.1}$$

$$\rho^f(\dot{v}_i + v_j v_{i,j}) = \sigma^f_{ij,j} + f_i^{FSI} \tag{2.2}$$

where v is the velocity, ρ^f is the density, σ^f_{ij} is the Cauchy stress tensor, and f_i^{FSI} denotes the fluid-structure interaction force. The Cauchy stress tensor is comprised of a pressure and viscous term defined as:

$$\sigma^f_{ij} = -p\delta_{ij} + 2\mu\tau_{ij} \tag{2.3}$$

where μ is the dynamic viscosity, p is the pressure, and the isotropic viscous term τ_{ij} is defined as:

$$\tau_{ij} = \frac{1}{2}(v_{i,j} + v_{j,i}) \tag{2.4}$$

The fluid boundary can be stipulated in the following manner:

$$v_i = g_i \ \texttt{on} \ \Gamma_{g_i} \tag{2.5}$$

$$\sigma^f_{ij} n_j = h_i \ \texttt{on} \ \Gamma_{h_i} \tag{2.6}$$

where both g_i and h_i are given functions describing the boundary. Γ_{g_i} is the part of the boundary at which the velocity is assumed to be specified and Γ_{h_i} is the natural boundary associated with the conditions on the stress components. The fluid boundary Γ is partitioned such that the Dirichlet Γ_{g_i} and Neumann Γ_{h_i} boundaries are complementary subsets of Γ, they span the entire domain but do not intersect:

$$\Gamma_{g_i} \bigcup \Gamma_{h_i} = \Gamma \tag{2.7}$$

$$\Gamma_{g_i} \bigcap \Gamma_{h_i} = 0 \tag{2.8}$$

2.3. Petrov-Galerkin Weak Form. The first step in the development of the weak form, consists of taking the product of the test functions δv_i and δp with the Navier-Stokes equations and integrating over the domain Ω [2]. This approach will lead to potential numerical instabilities associated with the standard Galerkin formulation of the problem. These numerical instability are also apparent in other standard discretization techniques such as finite difference and finite volume methods [3]. Such numerical oscillations in the velocity field are due to the presence of the advection term which become more apparent for advection-dominated flows, i.e. high Reynolds number, and flows with sharp layers in the solution. The other possible source of instability occurs when an inappropriate combination of interpolation functions is used for the velocity and pressure fields leading to oscillations mostly in the pressure field [4].

To reduce or eliminate the numerical oscillations in both the velocity and pressure fields at the boundaries. we employ the SUPG stabilization method on the velocity field and the PSPG method on the pressure field. The algorithm follows the stabilized equal-order finite element formulation developed in Ref. [3, 4].

The stabilization is achieved by adding two extra terms to the standard Galerkin formulation of the problem. The first term is the SUPG, which prevents oscillations caused by the advection term without introducing excessive numerical dissipation. The second term is the PSPG, which dampens oscillations in the pressure field, resulting in a more accurate numerical result.

Within the framework of SUPG, the modified test functions $\delta\tilde{v}_i$ and $\delta\tilde{p}$ are employed along with the stabilization parameters τ^m and τ^c:

$$\delta\tilde{v}_i = \overbrace{\delta v_i}^{\texttt{Galerkin}} + \overbrace{\tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}}^{\texttt{discontinuous}} \tag{2.9}$$

$$\delta\tilde{p} = \underbrace{\delta p}_{\texttt{Galerkin}} + \underbrace{\tau^c \delta v_{i,i}}_{\texttt{discontinuous}} \tag{2.10}$$

The selection of these stabilization parameters τ^m and τ^c has attracted a significant amount of attention and research [5, 6]. These parameters involve a measure of the local length scale, also referred to as the element length, and other parameters such as the local Reynolds and Courant numbers. Following the work of Tezduyar et.

al [5], the stabilization parameters are defined as:

$$\tau = \tau^m = \tau^c \tag{2.11}$$

$$\tau = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|v_i\|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-\frac{1}{2}} \tag{2.12}$$

where ν is the kinematic viscosity and h is the element length given as:

$$h = 2\|v_i\| \left(\sum_{A=1}^{n_{en}} |v_i N_{A,i}| \right)^{-1} \tag{2.13}$$

For the fluid domain, the weak form of the momentum equation is obtained by multiplying the velocity test function $\delta\tilde{v}_i$ and integrating over Ω:

$$0 = \int_\Omega \delta\tilde{v}_i \left[\rho^f (\dot{v}_i + v_j v_{i,j}) - \sigma^f_{ij,j} - f_i^{FSI} \right] d\Omega \tag{2.14}$$

Likewise, the weak form of the continuity equation is obtained by multiplying the pressure test function $\delta\tilde{p}$ and integrating over Ω:

$$0 = \int_\Omega \delta\tilde{p} v_{i,i} d\Omega \tag{2.15}$$

Following with a simple substitution:

$$0 = \int_\Omega (\delta p + \tau^c \delta v_{i,i}) \, v_{j,j} d\Omega \tag{2.16}$$

$$0 = \int_\Omega (\delta v_i + \tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}) \left[\rho^f (\dot{v}_i + v_j v_{i,j}) - \sigma^f_{ij,j} - f_i^{FSI} \right] d\Omega \tag{2.17}$$

Rewriting the momentum equation:

$$\begin{aligned} 0 = &\int_\Omega (\delta v_i + \tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}) \left[\rho^f (\dot{v}_i + v_j v_{i,j}) - f_i^{FSI} \right] d\Omega \\ &- \int_\Omega \delta v_i \sigma^f_{ij,j} d\Omega - \int_{\Omega^e} (\tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}) \, \sigma^f_{ij,j} d\Omega^e \end{aligned} \tag{2.18}$$

Employing integration by parts, we obtain:

$$\int_\Omega \delta v_i \sigma^f_{ij,j} d\Omega = \int_\Omega \left(\delta v_i \sigma^f_{ij} \right)_{,j} d\Omega - \int_\Omega \delta v_{i,j} \sigma^f_{ij} d\Omega \tag{2.19}$$

Employing divergence theorem, we obtain:

$$\int_\Omega \left(\delta v_i \sigma^f_{ij} \right)_{,j} d\Omega = \int_{\Gamma_{h_i}} \delta v_i h_i d\Gamma_{h_i} + \int_{\Gamma_{g_i}} \delta v_i \sigma^f_{ij} n_j d\Gamma_{g_i} \tag{2.20}$$

The weak form of the momentum equation is then written as:

$$\begin{aligned} 0 = &\int_\Omega (\delta v_i + \tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}) \left[\rho^f (\dot{v}_i + v_j v_{i,j}) - f_i^{FSI} \right] d\Omega \\ &+ \int_\Omega \delta v_{i,j} \sigma^f_{ij} d\Omega - \int_{\Gamma_{h_i}} \delta v_i h_i d\Gamma_{h_i} - \int_{\Gamma_{g_i}} \delta v_i \sigma^f_{ij} n_j d\Gamma_{g_i} \\ &- \int_{\Omega^e} (\tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}) \, \sigma^f_{ij,j} d\Omega^e \end{aligned} \tag{2.21}$$

The final weak form that is computed is the combination of the momentum and the continuity equation, which yields:

$$
\begin{aligned}
0 = & \int_{\Omega} \left(\delta v_i + \tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}\right) \left[\rho^f \left(\dot{v}_i + v_j v_{i,j}\right) - f_i^{FSI}\right] d\Omega \\
& + \int_{\Omega} \delta v_{i,j} \sigma_{ij}^f d\Omega - \int_{\Gamma_{h_i}} \delta v_i h_i d\Gamma_{h_i} - \int_{\Gamma_{g_i}} \delta v_i \sigma_{ij}^f n_j d\Gamma_{g_i} \\
& - \int_{\Omega^e} \left(\tau^m v_k \delta v_{i,k} + \tau^c \delta p_{,i}\right) \sigma_{ij,j}^f d\Omega^e + \int_{\Omega} \left(\delta p + \tau^c \delta v_{i,i}\right) v_{j,j} d\Omega
\end{aligned} \tag{2.22}
$$

2.4. Finite Element Discretization. The discretization process using finite elements follows the procedures in Ref. [1]. The velocity and pressure test functions, δv_i and δp, are interpolated in the following manner:

$$\delta v_i^h = \sum_A N_A c_{iA} \tag{2.23}$$

$$\delta p^h = \sum_A N_A q_A \tag{2.24}$$

The following is then obtained:

$$
\begin{aligned}
0 = & \sum_A \int_{\Omega} N_A c_{iA} \left[\rho^f \left(\dot{v}_i^h + v_j^h v_{i,j}^h\right) - f_i^{FSI,h}\right] d\Omega \\
& + \sum_A \int_{\Omega} \tau^m v_k^h N_{A,k} c_{iA} \left[\rho^f \left(\dot{v}_i^h + v_j^h v_{i,j}^h\right) - f_i^{FSI,h}\right] d\Omega \\
& + \sum_A \int_{\Omega} \tau^c N_{A,i} q_A \left[\rho^f \left(\dot{v}_i^h + v_j^h v_{i,j}^h\right) - f_i^{FSI,h}\right] d\Omega \\
& + \sum_A \int_{\Omega} N_{A,j} c_{iA} \sigma_{ij}^{f,h} d\Omega - \sum_A \int_{\Gamma_{h_i}} N_A c_{iA} h_i^h d\Gamma_{h_i} \\
& - \sum_A \int_{\Gamma_{g_i}} N_A c_{iA} \sigma_{ij}^{f,h} n_i^h d\Gamma_{g_i} - \sum_A \int_{\Omega^e} \tau^m v_k^h N_{A,k} c_{iA} \sigma_{ij,j}^{f,h} d\Omega^e \\
& - \sum_A \int_{\Omega^e} \tau^c N_{A,i} q_A \sigma_{ij,j}^{f,h} d\Omega^e + \sum_A \int_{\Omega} N_A q_A v_{j,j}^h d\Omega \\
& + \sum_A \int_{\Omega} \tau^c N_{A,i} c_{iA} v_{j,j}^h d\Omega
\end{aligned} \tag{2.25}
$$

Let $\sigma_{ij,j}^f$ be approximated by:

$$\sigma_{ij,j}^f = -p_{,j} \delta_{ij} \tag{2.26}$$

Combining test functions c_{iA} and q_A, we have four sets of equations with respect to four unknowns at each node A:

$$c_{iA}^T r_{iA}^v = 0 \tag{2.27}$$

$$q_A^T r_A^p = 0 \tag{2.28}$$

where the residual vectors are defined as:

$$
\begin{aligned}
r^v_{iA} = &\int_\Omega N_A \rho^f \dot{v}^h_i d\Omega + \int_\Omega N_A \rho^f v^h_j v^h_{i,j} d\Omega - \int_\Omega N_A f^{FSI,h}_i d\Omega \\
&+ \int_\Omega \tau^m v^h_k N_{A,k} \rho^f \dot{v}^h_i d\Omega + \int_\Omega \tau^m v^h_k N_{A,k} \rho^f v^h_j v^h_{i,j} d\Omega - \int_\Omega \tau^m v^h_k N_{A,k} f^{FSI,h}_i d\Omega \\
&+ \int_\Omega N_{A,j} \sigma^{f,h}_{ij} d\Omega - \int_{\Gamma_{h_i}} N_A h^h_i d\Gamma_{h_i} - \int_{\Gamma_{g_i}} N_A \sigma^{f,h}_{ij} n^h_i d\Gamma_{g_i} \\
&+ \int_{\Omega^e} \tau^m v^h_k N_{A,k} p^h_{,i} d\Omega^e + \int_\Omega \tau^c N_{A,i} v^h_{j,j} d\Omega
\end{aligned} \tag{2.29}
$$

$$
\begin{aligned}
r^p_A = &\int_\Omega \tau^c N_{A,i} \rho^f \dot{v}^h_i d\Omega + \int_\Omega \tau^c N_{A,i} \rho^f v^h_j v^h_{i,j} d\Omega - \int_\Omega \tau^c N_{A,i} f^{FSI,h}_i d\Omega \\
&+ \int_{\Omega^e} \tau^c N_{A,i} p^h_{,i} d\Omega^e + \int_\Omega N_A v^h_{j,j} d\Omega
\end{aligned} \tag{2.30}
$$

The following equation is to be solved:

$$
\underbrace{\begin{bmatrix} r^{v,v}_{iA} & r^{v,p}_{iA} \\ r^{p,v}_A & r^{p,p}_A \end{bmatrix}}_{LSH} \begin{Bmatrix} \delta v_i \\ \delta p \end{Bmatrix} = -\underbrace{\begin{bmatrix} r^v_{iA} \\ r^p_A \end{bmatrix}}_{RHS} \tag{2.31}
$$

Now let us descritize and define LHS as:

$$
LHS = m_{iAjB} \begin{Bmatrix} \Delta \dot{d}_{jB} \\ \Delta \dot{q}_B \end{Bmatrix} + k_{iAjB} \begin{Bmatrix} \Delta d_{jB} \\ \Delta q_B \end{Bmatrix} \tag{2.32}
$$

where $i, j = 1, 2, 3$. The mass matrix m_{iAjB} and the stiffness matrix k_{iAjB} are defined as:

$$
r^{v,v}_{iA} = \int_\Omega N_A \rho^f \sum_B N_B \dot{d}_{iB} d\Omega + \int_\Omega \tau^m v^h_k N_{A,k} \rho^f \sum_B N_B \dot{d}_{iB} d\Omega \tag{2.33}
$$

$$
r^{v,p}_{iA} = 0 \tag{2.34}
$$

$$
\begin{aligned}
\Delta r^{v,v}_{iA} &= \int_\Omega N_A \rho^f \sum_B N_B \Delta \dot{d}_{iB} d\Omega + \int_\Omega \tau^m v^h_k N_{A,k} \rho^f \sum_B N_B \Delta \dot{d}_{iB} d\Omega \\
&= \sum_{jB} \underbrace{\int_\Omega \rho^f N_A N_B \delta_{ij} + \rho^f \tau^m v^h_k N_{A,k} N_B \delta_{ij} d\Omega}_{m_{iAjB}} \Delta \dot{d}_{jB}
\end{aligned} \tag{2.35}
$$

$$
r^{p,v}_A = \int_\Omega \tau^c N_{A,i} \rho^f \sum_B N_B \dot{d}_{iB} d\Omega = \int_\Omega \tau^c N_{A,l} \rho^f \sum_B N_B \dot{d}_{lB} d\Omega \tag{2.36}
$$

$$
r^{p,p}_A = 0 \tag{2.37}
$$

$$
\begin{aligned}
\Delta r^{p,v}_A &= \int_\Omega \tau^c N_{A,l} \rho^f \sum_B N_B \Delta \dot{d}_{lB} d\Omega \\
&= \sum_{jB} \underbrace{\int_\Omega \rho^f \tau^c N_{A,l} N_B \delta_{jl} d\Omega}_{m_{4AjB}} \Delta \dot{d}_{jB}
\end{aligned} \tag{2.38}
$$

$$
\begin{aligned}
r_{iA}^{v,v} = & \int_\Omega N_A \rho^f \sum_B N_B d_{jB} v_{i,j}^h d\Omega + \int_\Omega N_A \rho^f v_j^h \sum_B N_{B,j} d_{iB} d\Omega \\
& + \int_\Omega \tau^m v_k^h N_{A,k} \rho^f \sum_B N_B d_{jB} v_{i,j}^h d\Omega + \int_\Omega \tau^m v_k^h N_{A,k} \rho^f v_j^h \sum_B N_{B,j} d_{iB} d\Omega \\
& + \int_\Omega N_{A,j} \mu \sum_B \left(N_{B,j} d_{iB} + N_{B,i} d_{jB} \right) d\Omega + \int_\Omega \tau^c N_{A,i} \sum_B N_{B,j} d_{jB} d\Omega
\end{aligned} \tag{2.39}
$$

$$
r_{iA}^{v,p} = - \int_\Omega N_{A,i} \sum_B N_B q_B d\Omega + \int_\Omega \tau^m v_k^h N_{A,k} \sum_B N_{B,i} q_B d\Omega \tag{2.40}
$$

$$
\begin{aligned}
\Delta r_{iA}^{v,v} = & \int_\Omega N_A \rho^f \sum_B N_B \Delta d_{jB} v_{i,j}^h d\Omega + \int_\Omega N_A \rho^f v_j^h \sum_B N_{B,j} \Delta d_{iB} d\Omega \\
& + \int_\Omega \tau^m v_k^h N_{A,k} \rho^f \sum_B N_B \Delta d_{jB} v_{i,j}^h d\Omega + \int_\Omega \tau^m v_k^h N_{A,k} \rho^f v_j^h \sum_B N_{B,j} \Delta d_{iB} d\Omega \\
& + \int_\Omega N_{A,j} \mu \sum_B \left(N_{B,j} \Delta d_{iB} + N_{B,i} d_{jB} \right) d\Omega + \int_\Omega \tau^c N_{A,i} \sum_B N_{B,j} \Delta d_{jB} d\Omega \\
= & \sum_{jB} \underbrace{\int_\Omega N_A \rho^f N_B v_{i,j}^h + N_A \rho^f v_l^h N_{B,l} \delta_{ij} d\Omega}_{k_{iAjB}} \Delta d_{jB} \\
& + \sum_{jB} \underbrace{\int_\Omega \tau^m v_k^h N_{A,k} \rho^f N_B v_{i,j}^h + \tau^m v_k^h N_{A,k} \rho^f v_l^h N_{B,l} \delta_{ij} d\Omega}_{k_{iAjB}} \Delta d_{jB} \\
& + \sum_{jB} \underbrace{\int_\Omega N_{A,l} \mu N_{B,l} \delta_{ij} + N_{A,j} \mu N_{B,i} + \tau^c N_{A,i} N_{B,j} d\Omega}_{k_{iAjB}} \Delta d_{jB}
\end{aligned} \tag{2.41}
$$

$$
\begin{aligned}
\Delta r_{iA}^{v,p} = & - \int_\Omega N_{A,i} \sum_B N_B \Delta q_B d\Omega + \int_\Omega \tau^m v_k^h N_{A,k} \sum_B N_{B,i} \Delta q_B d\Omega \\
= & \sum_{jB} \underbrace{\int_\Omega -N_{A,i} N_B + \tau^m v_k^h N_{A,k} N_{B,i} d\Omega}_{k_{iA4B}} \Delta q_B
\end{aligned} \tag{2.42}
$$

$$
\begin{aligned}
r_A^{p,v} = & \int_\Omega \tau^c N_{A,i} \rho^f \sum_B N_B d_{jB} v_{i,j}^h d\Omega + \int_\Omega \tau^c N_{A,i} \rho^f v_j^h \sum_B N_{B,j} d_{iB} d\Omega \\
& + \int_\Omega N_A \sum_B N_{B,j} d_{jB} d\Omega
\end{aligned} \tag{2.43}
$$

$$
r_A^{p,p} = \int_\Omega \tau^c N_{A,i} \sum_B N_{B,i} q_B d\Omega \tag{2.44}
$$

$$\begin{aligned}\Delta r_A^{p,v} &= \int_\Omega \tau^c N_{A,i}\rho^f \sum_B N_B \Delta d_{jB} v_{i,j}^h d\Omega + \int_\Omega \tau^c N_{A,i}\rho^f v_j^h \sum_B N_{B,j}\Delta d_{iB} d\Omega \\ &\quad + \int_\Omega N_A \sum_B N_{B,j}\Delta d_{jB} d\Omega \\ &= \sum_{jB} \underbrace{\int_\Omega \tau^c N_{A,i}\rho^f N_B v_{i,j}^h + \tau^c N_{A,j}\rho^f v_l^h N_{B,l} + N_A N_{B,j} d\Omega}_{k_{4AjB}} \Delta d_{jB} \qquad (2.45)\end{aligned}$$

$$\begin{aligned}\Delta r_A^{p,p} &= \int_\Omega \tau^c N_{A,i} \sum_B N_{B,i}\Delta q_B d\Omega \\ &= \sum_{jB} \underbrace{\int_\Omega \tau^c N_{A,i} N_{B,i} d\Omega}_{k_{4A4B}} \Delta q_B \qquad (2.46)\end{aligned}$$

2.5. Mixed Integrator Scheme. In this section the mixed integration scheme implemented in TAHOE will be examined. Consider the following non-linear Navier Stokes equations in matrix form:

$$\begin{bmatrix} M_{vv} & M_{vp} \\ M_{pv} & \varepsilon M_{pp} \end{bmatrix} \begin{Bmatrix} \dot{V}_{n+1} \\ \dot{P}_{n+1} \end{Bmatrix} + \begin{Bmatrix} N_v(V_{n+1}, P_{n+1}) \\ N_p(V_{n+1}, P_{n+1}) \end{Bmatrix} = \begin{Bmatrix} F_{n+1} \\ G_{n+1} \end{Bmatrix} \qquad (2.47)$$

The mixed integration encompasses the trapezoidal rule, also referred to as the Crank-Nicolson method, on the fluid velocity and backward Euler on the pressure [8]. These computational algorithms are well-known members of the trapezoidal family [2]. The mixed integration can be explicitly written as:

$$V_{n+1} = V_n + \Delta t \dot{V}_{n+\frac{1}{2}} \qquad (2.48)$$

$$P_{n+1} = P_n + \Delta t \dot{P}_{n+1} \qquad (2.49)$$

The predictor can be defined as:

$$\tilde{V}_{n+1} = V_n + \frac{1}{2}\Delta t \dot{V}_n \qquad (2.50)$$

$$\tilde{P}_{n+1} = P_n \qquad (2.51)$$

Starting with the first iteration, initialize:

$$i = 0 \qquad (2.52)$$

$$\dot{V}_{n+1}^{(i)} = 0 \qquad (2.53)$$

$$\dot{P}_{n+1}^{(i)} = 0 \qquad (2.54)$$

$$V_{n+1}^{(i)} = \tilde{V}_{n+1} \qquad (2.55)$$

$$P_{n+1}^{(i)} = \tilde{P}_{n+1} = P_n \qquad (2.56)$$

Following the implementation of the V-from, given by Hughes [2], the following equations are solved:

$$[M^\star] \begin{Bmatrix} \Delta\dot{V} \\ \Delta\dot{P} \end{Bmatrix} = \begin{Bmatrix} R_{v,n+1}^{(i)} \\ R_{p,n+1}^{(i)} \end{Bmatrix} \qquad (2.57)$$

where:

$$\left\{ \begin{array}{c} R^{(i)}_{v,n+1} \\ R^{(i)}_{p,n+1} \end{array} \right\} = \left\{ \begin{array}{c} F_{n+1} \\ G_{n+1} \end{array} \right\} - \left\{ \begin{array}{c} N_v(V^{(i)}_{n+1}, P^{(i)}_{n+1}) \\ N_p(V^{(i)}_{n+1}, P^{(i)}_{n+1}) \end{array} \right\} - \left[\begin{array}{cc} M & 0 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} \dot{V}^{(i)}_{n+1} \\ \dot{P}^{(i)}_{n+1} \end{array} \right\} \tag{2.58}$$

The corrector can be defined as:

$$\dot{V}^{(i+1)}_{n+1} = \dot{V}^{(i)}_{n+1} + \Delta\dot{V} \tag{2.59}$$

$$V^{(i+1)}_{n+1} = \tilde{V}_{n+1} + \frac{1}{2}\Delta t\dot{V}^{(i+1)}_{n+1} \tag{2.60}$$

$$= V^{(i)}_{n+1} + \frac{1}{2}\Delta t\Delta\dot{V} \tag{2.61}$$

$$\dot{P}^{(i+1)}_{n+1} = \dot{P}^{(i)}_{n+1} + \Delta\dot{P} \tag{2.62}$$

$$P^{(i+1)}_{n+1} = \tilde{P}_{n+1} + \Delta t\dot{P}^{(i+1)}_{n+1} \tag{2.63}$$

$$= P^{(i)}_{n+1} + \Delta t\Delta\dot{P} \tag{2.64}$$

To find $M^\star$, the original equation is linearized:

$$\left[\begin{array}{cc} M_{vv} & M_{vp} \\ M_{pv} & \varepsilon M_{pp} \end{array} \right] \left\{ \begin{array}{c} \dot{V}^{(i+1)}_{n+1} \\ \dot{P}^{(i+1)}_{n+1} \end{array} \right\} + \left\{ \begin{array}{c} N_v(V^{(i+1)}_{n+1}, P^{(i+1)}_{n+1}) \\ N_p(V^{(i+1)}_{n+1}, P^{(i+1)}_{n+1}) \end{array} \right\} = \left\{ \begin{array}{c} F_{n+1} \\ G_{n+1} \end{array} \right\} \tag{2.65}$$

After substituting the corrector:

$$\begin{aligned} &\left[\begin{array}{cc} M_{vv} & M_{vp} \\ M_{pv} & \varepsilon M_{pp} \end{array} \right] \left\{ \begin{array}{c} \dot{V}^{(i)}_{n+1} + \Delta\dot{V} \\ \dot{P}^{(i)}_{n+1} + \Delta\dot{P} \end{array} \right\} \\ &\qquad + \left\{ \begin{array}{c} N_v(V^{(i)}_{n+1} + \frac{1}{2}\Delta t\Delta\dot{V}, P^{(i)}_{n+1} + \Delta t\Delta\dot{P}) \\ N_p(V^{(i)}_{n+1} + \frac{1}{2}\Delta t\Delta\dot{V}, P^{(i)}_{n+1} + \Delta t\Delta\dot{P}) \end{array} \right\} = \left\{ \begin{array}{c} F_{n+1} \\ G_{n+1} \end{array} \right\} \end{aligned} \tag{2.66}$$

The linearized equation to be solve is now in its final form:

$$\underbrace{\left[\begin{array}{cc} M_{vv} + \frac{1}{2}\Delta tK_{vv} & M_{vp} + \Delta tK_{vp} \\ M_{pv} + \frac{1}{2}\Delta K_{pv} & \varepsilon M_{pp} + \Delta tK_{pp} \end{array} \right]}_{M^\star} \left\{ \begin{array}{c} \Delta\dot{V} \\ \Delta\dot{P} \end{array} \right\} = \left\{ \begin{array}{c} R^{(i)}_{v,n+1} \\ R^{(i)}_{p,n+1} \end{array} \right\} \tag{2.67}$$

2.6. TAHOE Results. To test the stabilized formulation and finite element method implementation of the Navier-Stokes equations using the mixed integrator in TAHOE, a sample problem has been devised. Consider a pipe with a length of $10m$ with a square cross-section of length $1m$ on the side, depicted in Fig.**??**. The following initial conditions have been prescribed for all nodes:

$$v_x = 1\frac{m}{s}$$
$$v_y = 0\frac{m}{s}$$
$$v_z = 0\frac{m}{s}$$

In terms of the boundary conditions, constant velocity $v_x = 1\frac{m}{s}$ has been prescribed at the inlet and a constant pressure $p = 0Pa$ at the outlet. There is also a no

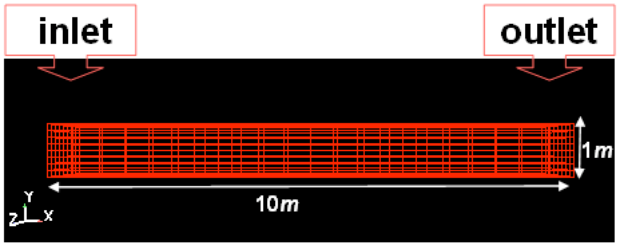


FIG. 2.1. *Sample 3D problem: square pipe.*

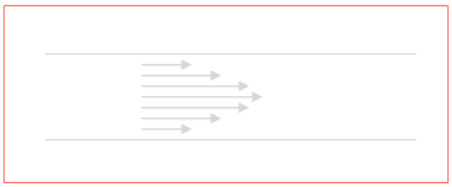


FIG. 2.2. *Expected Poiseuille flow.*

slip and no penetration boundary condition prescribed at the walls of the pipe. The Reynolds number associated with the depicted problem is roughly 22.63, indicating a laminar flow.

Prior to analyzing the results, we expect a Poiseuille flow to develop over time, roughly a diameter length into the pipe, as shown in Fig. 2.2.

A sample problem was ran in TAHOE on one processor employing the SPOOLES matrix. Looking at time step $t = 0$ in Fig. 2.3, on the bottom we can see the velocity field at the cross-section of the pipe. The first graph, on the left, depicts a velocity field at the cross-section perpendicular to the length of the pipe. The next two graphs depict the velocity and pressure field at the cross-section parallel to the length of the pipe, respectively. Examining this figure, we can easily see a uniform velocity inside the pipe and zero velocity at the walls.

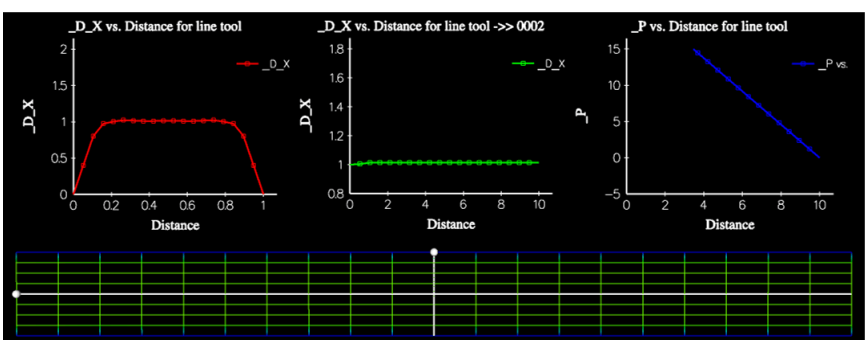


FIG. 2.3. *Sample 3D problem: result at time step t=0.*

Examining the results at the next time steps, Fig. 2.4 and Fig. 2.5, we can clearly see that over time a Poiseuille flow is being developed. Furthermore, looking at the the velocity field at the cross-section of the pipe we can also see the Poiseuille flow being developed roughly one diameter length into the pipe, as expected. We can also

see that the velocity field in the center of the pipe increases until it reaches a constant value and we have a linear pressure drop across the pipe. One can also notice some oscillation in both the velocity and pressure fields. These oscillations can be further dampened with a small adjustment of the stabilization parameters.

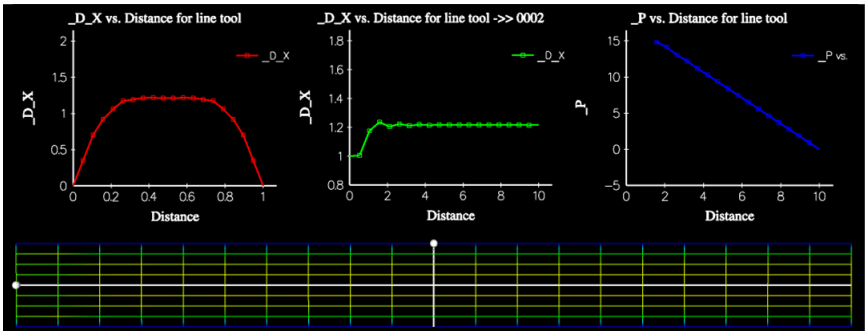


FIG. 2.4. *Sample 3D problem: result at time step t=n.*

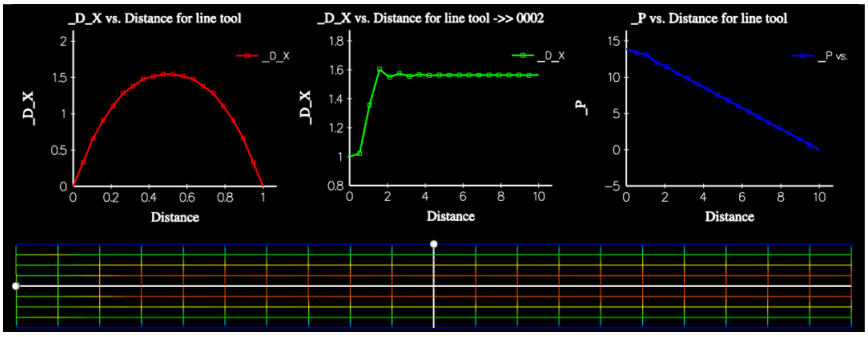


FIG. 2.5. *Sample 3D problem: result at time step t=n+1.*

3. Solid Dynamics.

3.1. Introduction. In this section, we discuss in detail the 3D finite element solid solver used in the IFEM formulation that has been already implemented within TAHOE's infrastructure. TAHOE encompassed many solid solvers, which can be linked with the fluid solver via the RKPM delta functions.

4. Conclusions. During the 10-week program, most of the effort has been devoted to implement a fluid solver within TAHOE's infrastructure. In particular, the stabilized formulation of the nonlinear Navier-Stokes equations for incompressible flows using the SUPG and the PSPG method has been implemented. Currently, my collaborators and I are verifying the implemented fluid algorithm on parallel machines. The future goal of this project will be to implement a multi-manger within TAHOE for both solid and fluid sub-domains. Furthermore, RKPM delta functions must be incorporated to enforce continuity via the interpolation of force and velocity. This method has been borrowed from meshfree methods, of which some are already implemented in TAHOE. The next step will be to further expand IFEM to the immersed electrokinetic finite element method (IEFEM). The method couples the electric field

with hydrodynamics to study the motion and deformation of flexible objects immersed in a fluid under an applied electric field. This methodology will allow the modelling assemblies of flexible, arbitrarily shaped nano/bio materials in fluid environments with complex electrode geometries, under an electric field.

REFERENCES

[1] T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear finite elements for continua and structures*, John Wiley and Sons, 2000.

[2] T. J. R. Hughes, *The finite element method*, Prentice Hall, 1987.

[3] T. E. Tezduyar, *Stabilized finite element formulations for incompressible flow computations*, Advances in applied mechanics, 28 (1991), pp. 1–44.

[4] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih, *Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements*, Computer Methods in Applied Mechanics and Engineering, 95 (1992), pp. 221–242.

[5] T. E. Tezduyar and Y. Osawa, *Finite element stabilization paramters computed from element matrices and vectors*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 411–430.

[6] T. E. Tezduyar and S. Sathe, *Stabilization parameters in supg and pspg formulations*, Journal of Computational and Applied Mechanics, 4 (2003), pp. 71–88.

[7] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, *Immersed finite element method*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 2051–2067.

[8] L. T. Zhang, G. J. Wagner, and W. K. Liu, *A parallelized meshfree method with boundary enrichment for large-scale cfd*, Journal of Computational Physics, 176 (2002), pp. 483–506.

DIFFUSIVE TRANSPORT OF FLUORESCEIN IN A NANOFLUIDIC DEVICE

S. RHIEU*, D. HUBER†, AND S. PENNATHUR‡

Abstract. This paper describes a method to estimate the zeta potential of nanofluidic T-shaped channels fabricated in quartz substrates with a nominal depth of 800 nm and width of 10 μm. First, simulations are preformed to predict the diffusion of fluorescein (FL) molecules in nanofluidic T-channels with electro-osmotically driven flow. Then, we calculated the Peclet number of fluorescein in nanofluidic channels by comparing simulations with experimental images of the resulting concentration profile. From the Peclet number, we estimate the zeta potential of the quartz nanofluidic T-shaped channels to be -72 mV.

1. Introduction. In the past decade, microfluidic systems have been extensively investigated. Such systems manipulate volumes of fluid on the order of nanoliters or even picoliters. Recently, there has been a growing interest in reducing the scale to nanometer dimensions. As the channels dimensions of the device become smaller, the behavior of the liquid in such nanometer-sized channels changes as compared to larger, micron-sized channels [1–3]. In particular, the electrical double layer (EDL), defined as an interfacial region between a charged surface and electrolytic medium, plays an important role in nanoscale systems [4]. As a result, there have been many recent investigations on a new range of phenomena dealing with the role of the electrical double layer in affecting fluid motion. In this paper we present continuum-based numerical simulations and experimental studies of the diffusive transport of fluorescein in a nanofluidic device. By comparing the experimental results against a series of simulations, we estimate the zeta potential within the channel.

2. Theory.

2.1. Governing Equations. In this section, we summarize the governing equations for transport of liquids in nanofludic channel. First, we begin with classical equations describing electrokinetic flows. The continuity equation and conservation of momentum equation that describe the flow in a nanofluidic channel can be used to solve for the velocity term, u. These corresponding continuum equations are

$$\nabla \cdot u = 0 \ , \ \rho \frac{\partial u}{\partial t} - \eta \nabla^2 u + \nabla P = f \tag{2.1}$$

$$\nabla^2 \Phi = 0 \ , \ \nabla^2 \psi = -\frac{\rho_e}{\varepsilon \varepsilon_o} \tag{2.2}$$

where u is the velocity of the liquid, ρ_e is the charge density, η is the viscosity, ε is the permittivity, and p is the pressure. Equations (2.1) describes the motion for Stokes flow subject to a body force term, f, including an electrostatic body force defined as equations (2.2), Laplaces equation and the potential field associated with the charges of the EDL.

Considering a charged molecule such as fluorescein in a nanochannel, the overall transport of the molecule is governed by the following equation.

$$\frac{\partial C_i}{\partial t} + \vec{u} \cdot \nabla C_i = -z_i \nu_i F \nabla \cdot (C_i \nabla \Phi) + D_i \nabla^2 C_i \tag{2.3}$$

*Brown University
†Sandia National Laboratories

where C refers to the concentration of the charged species, F is Faradays constant, D is the molecular diffusion coefficient, and ν is the mobility of a species defined as $U/\Phi F z$, where U is the net velocity for an ion valence z subject to an electric field Φ. The velocity obtained from equations (2.1) and (2.2) can be used in the electromigration-convective-diffusion equation (2.3) to calculate the concentration profile of fluorescein molecules in a channel.

2.2. Peclet number and Zeta potential. The Peclet number ($Pe = UL/D$, where U and L are the characteristic velocity and length scale for the flow, and D is the molecular diffusivity) is a dimensionless number which characterizes the relative strengths of diffusive and convective species transport in a flow field [6]. The Peclet number for fluorescein in our nanofluidic device can be calculated by comparing results from experiments and simulations. In particular, the effective mobility of fluorescein under the influence of an electric field, E, is defined as

$$\mu_{eff} = -\varepsilon\zeta E/\eta + D/\left(zFRT\right) \tag{2.4}$$

where $-\varepsilon\zeta E/\eta$ is the electroosmotic mobility, $D/\left(zFRT\right)$ is defined as the electrophoretic mobility and where R is the universal gas constrant. Since the characteristic velocity of fluorescein is determined by the product of the effective mobility and given electric field, the Peclet number has dependent variables z, D, Φ, and ζ. Therefore, the calculated Pe can be used to deduce one unknown parameter such as zeta potential if we are given other parameters.

3. Methods.

3.1. Experimental set up. Nanofluidic devices were fabricated using conventional micromachining techniques in quartz substrates with a nominal depth of 800 nm and width of 10 μm (Figure 3.1). We used a mercury lamp to optically monitor

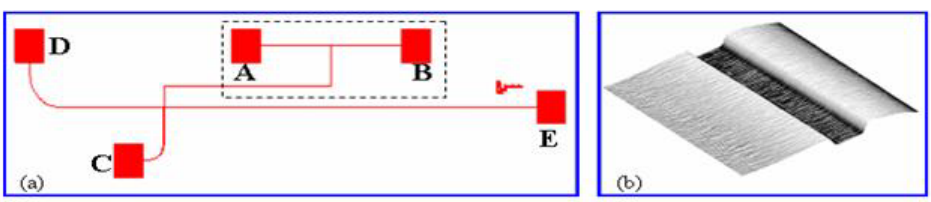


FIG. 3.1. *(a) Schematic of a nanofluidic channel. The dashed box encloses the section of the device where the diffusive transport of FL was studied. A and B are reservoirs containing 10 mM sodium borate buffer and buffer with 100 μM FL, respectively. A uniform voltage was applied to A and B while C was grounded. (b) An atomic force microscopy image of a channel with the 800 nm deep x 10 μm wide channel.*

the intensity of fluorescence of two liquid streams (10 mM sodium borate and 100 μM FL). Images were captured using a CCD camera (Cascade, Roper Scientific) with a 128 x 128 CCD pixel array and 12-bit digitization (Figure 3.2(a)). The exposure time was 50 ms and the on-chip gain was 1800. Thirty frames were taken for raw and background images, and a flatfield image was used for image correction.

3.2. Experimental Conditions. First, all reagents used in the experiments were filtered using Acrodisc syringe filter with 0.2 μm PVDF membrane (Pall Corporation). Then, the reservoirs of our nanofluidic device were filled with deionized water and the channels were flushed for 10 minutes before exchanging the deionized water

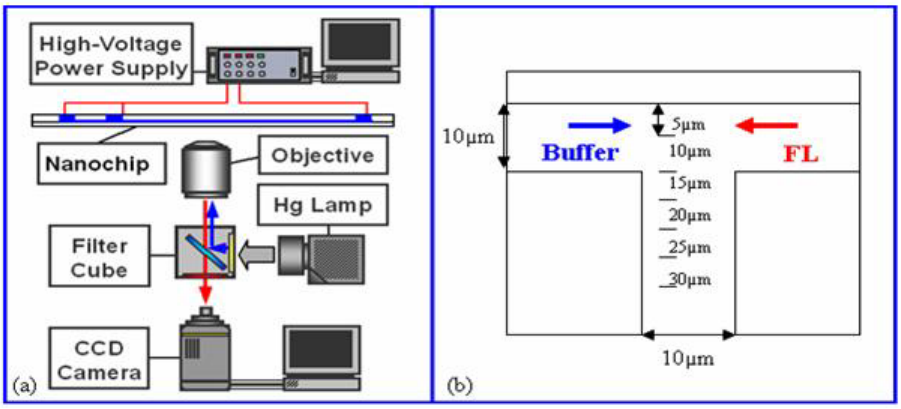


FIG. 3.2. *(a) Schematic diagram of the experimental set-up. (b) A magnified figure of the T-intersaction of the channel where the dyed and undyed streams meet. Six different detection points (with 5 μm increment from top edge of the channel) were measured from both experiment and simulation.*

TABLE 3.1
Voltage scheme used for three different electric field. Characteristic E-fields are listed in fourth column.

	Voltage$_{(A,\ B,\ E)}$	Voltage$_{(D)}$	Voltage$_{(C)}$	E-fields at detection points
Low-field	400V	400V	250V	14201 [V/m]
Medium-field	800V	800V	500V	28403 [V/m]
High-field	1584V	990V	0V	56238 [V/m]

for 10 mM sodium borate. A high voltage power supply (Sandia's μChemLab^{TM}) applied specified potentials to the three chip reservoirs via platinum electrodes. 15 μl of 100 μM fluorescein was loaded at reservoir B. The uniform voltage of 990V was applied to electrodes in reservoirs A and B while D and E were floating and C was grounded. The electric fields used in both the experiments and the simulations are listed in Table 3.1.

3.3. Image analysis. The data from the CCD camera is saved as TIFF files and image correction is preformed on these images using MATLAB. First, 30 background images and 30 raw images are obtained and averaged. CCD images are corrected by applying the following a pixel by pixel operation [5] to each image:

$$I_{corr} = \frac{I_{raw} - I_{background}}{I_{flatfield} - I_{background}} \tag{3.1}$$

where a background image is subtracted from the raw image, and this difference is normalized by the difference between a flatfield (obtained by imaging of the nanochannel filled with a homogeneous concentration of fluorescein) and the background image.

3.4. Numerical model. In our simulation approach we used Comsol Multiphysics 3.2a (Comsol, Inc.) to perform numerical simulations of the governing equations that are fully coupled. First, we used a MEMS module for Conductive Media DC corresponding to equation (2.2) above and Electrokinetic Flow corresponding to equations (2.3) and (2.4) and coupled them with Stokes flow corresponding to equation (2.1). Second, we compared the experimental results against a series of simulation

data with different zeta potentials. The same conditions used in the experiments were applied to the simulations. From the results, we found the simulation that matched closest to the experimental images and thus determined the zeta potential of the channel. We typically ran four simulations to obtain an accurate result. . Finally, we performed the simulations with the determined zeta potential at two other electric fields and compared simulation data with experimental results for validation of our method.

4. Results and Discussion.

4.1. Zeta potential estimation. Comparison of fluorescence intensity across the channel from both experiments and simulations at high electric field of 56200 V/m is presented in Figure 4.1. To determine the zeta potential found the Pe number that

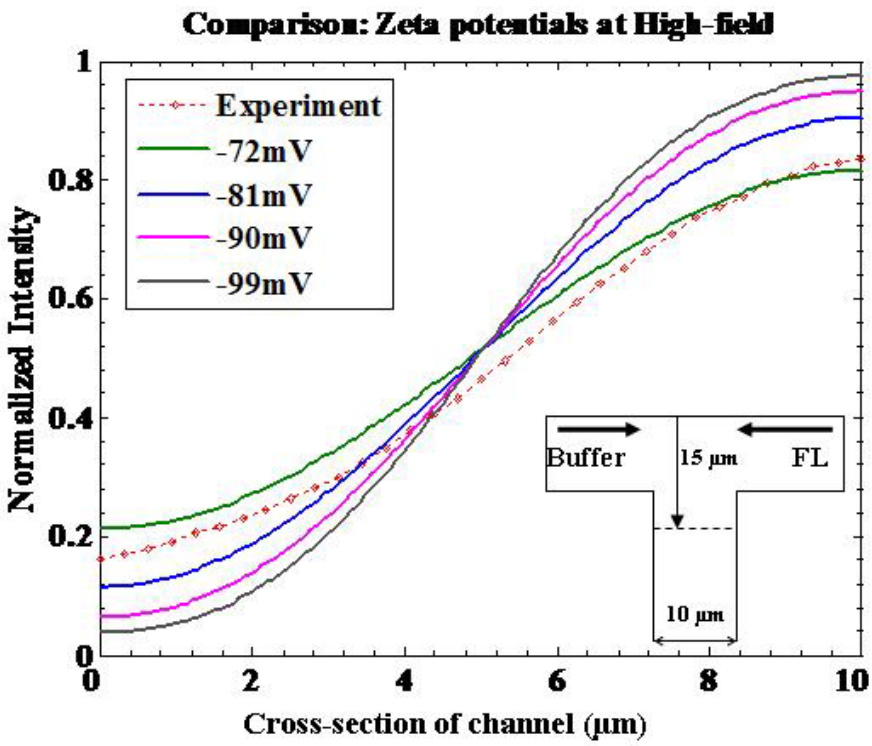


FIG. 4.1. *Comparison of concentration profile across the channel between experimental result (red dotted line) and simulation data. For high-field, experimental result shows agreement with simulation data (green line) at zeta potential -0.072 V.*

gave a good agreement in concentration profiles between experiments and simulations. Given a concentration profile of the experiment, we performed simulations by varying zeta potentials until the best matched case was found. In doing so, we estimated the zeta potential of our nanofluidic system to be approximately -72 mV.

4.2. Concentration profile across the channel. Figure 4.2 shows three pairs of concentration profiles comparing experimental image data and simulations at the junction of the channel for three different electric fields. At the electric field of 56200 V/m, the images of concentration profiles from A and B show a sharp interface between the fluorescein and background buffer compared to images at lower electric

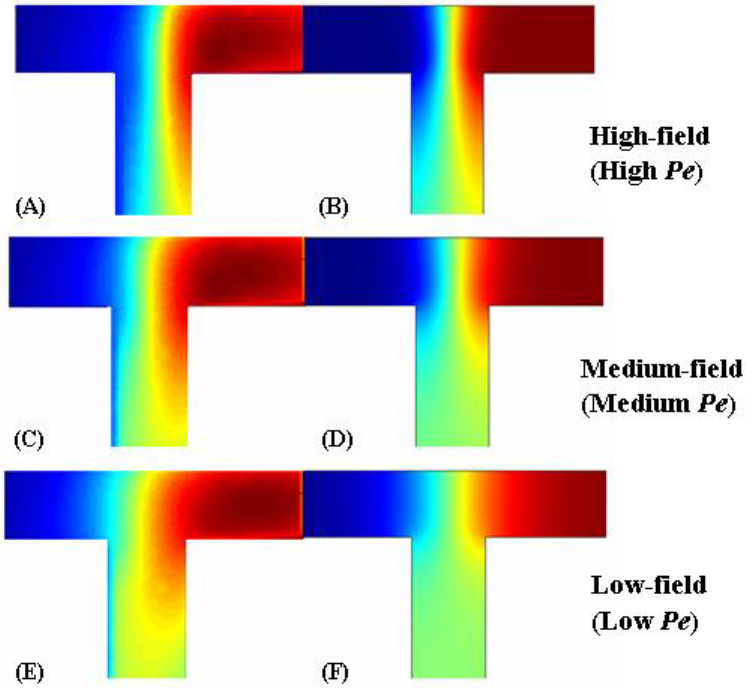


FIG. 4.2. *Concentration profiles of experimental result (A,C,E) and numerical simulation (B,D,F) in a 10 μm x 10 μm nanofluidic channel with electrokinecally-driven flow subject to three different electric fields. Red regions indicate fluorescein while blue regions show sodium borate buffer. Notice that the asymmetric concentration profiles of flows from fluorescein and buffer sides are shown at both medium-field and low-field as opposed to high-field. This is mostly due to different gradient pressures at each reservoirs of fluorescein and buffer or channel clogging which results in flow resistance.*

fields. This is most likely due to the fact that the bulk electroosmotic velocity is faster as opposed at higher electric fields. In addition, the influence of diffusion is small in the channel where higher electric field is applied. The images of A and B also qualitatively show that there is a balance of flow from both the buffer and fluorescein sides of the well. The experimental data (A) at the high field agrees best with simulation data among three pairs (data not shown) because the magnitudes of the velocity of fluorescein and buffer from both arms are the most symmetrical. Compared with the results at high electric field, the interfaces between fluorescein and buffer at both medium and low electric fields are skewed to the left (buffer side). The discrepancy between experiment (red dotted line) and simulation (green line) is consistent with what we previously observed results presented in Figure 4.1. This asymmetric concentration profiles are presumably due to different gradient pressures at each reservoirs of fluorescein and buffer. Moreover, it is possible that the misaligned filters at sample reservoirs cause some clogging so that flow resistance changes as the experiment runs.

5. Conclusions. We have presented continuum based numerical simulations and experimental results from diffusive transport studies of fluorescein in a nanofluidic device that determines zeta potential of the channels to be -72 mV. Our method is applicable not only for estimating zeta potential of the channel but also determining other useful parameters that characterize the Peclet number of a molecule in nanofluidic

systems such as diffusion coefficient, mobility, and viscosity. Future work will include developing a design strategy to estimate diffusion coefficient of DNA molecules in nanofluidic devices.

REFERENCES

[1] A. Hibara *et al.*, Anal. Chem., 74 (2002), pp. 6170–6176.
[2] K.D. Bartle *et al.*, J. Chromatography A, 916 (2001), pp. 3–23.
[3] Q. Pu *et al.*, Proc. μTAS, (2003), pp. 657–660.
[4] M.P. Hughes, *Nanoelectromechanics in Engineering and Biology*, CRC Press.
[5] Photometrics Ltd., *Charged-coupled devices for quantitative electronic imaging*, 1992.
[6] B.I. Shraiman, Physical Review A, 36 (1987), pp. 261–267.

EVALUATION OF LEVEL SET TOPOLOGY OPTIMIZATION FORMULATIONS FOR DESIGN OF MINIMUM-DISPERSION MICROFLUIDIC DEVICES

A.R. TERREL* AND K.R. LONG†

Abstract. Applying topology optimization to minimize the sample dispersion of a microfluidic device is challenging in part because the fictitious permeability used to classify topological regions tends to a smooth non-Boolean field, in contradiction to the requirement that a valid solution be Boolean. Imposing the Boolean condition must be done carefully in order to avoid spurious local minimizers or an excessively smooth solution. Here we investigate the behavior of two different formulations of level-set based topology optimization on problems where we must find the Boolean field that best approximates a specified non-Boolean field. We introduce a formulation based on inequality constraints that can effectively satisfy the Boolean condition.

1. Introduction. The goal in shape optimization is to identify the shape of a domain $\mathcal{I}$ such that when a physical problem is solved on that domain, an objective function is minimized. The most general form of shape optimization is topology optimization [2], in which the topology of the domain is allowed to vary; *i.e.,* topological holes and/or islands can appear or disappear. By its nature, topology optimization is a large-scale boolean programming problem, because each point in a larger, embedding domain is either in $\mathcal{I}$ or not. However, in practice it is rarely approached directly in that form; rather, each point in the domain is characterized by a real number to be driven, somehow, to the boolean extremes of 0 or 1. Much of the art in topology optimization is choosing this reparametrization in such a way closely approximates a boolean field without introducing spurious local minimizers.

In many problems, there is a natural reparametrization that is simple and works well. In structural topology optimization, for example, if the relationship between material density and stiffness is taken to be nonlinear and monotonic, then for certain objective functions there will be no payoff to intermediate material densities. In other words, in such problems the nature of the physical problem, objective function, and constraints drives the material density naturally to a boolean solution [2]. In this case, nothing special has to be added to the method to produce a boolean solution.

Here, though, we are interested in problems where no such lucky accident happens. The motivating problem for this study is the design of minimum-dispersion microfluidic sensors; by minimum-dispersion we mean that a sample of particles that enters the device together will not be dispersed by the flow. We represent the shape of the channel by introducing an approximately-boolean permeability. The challenging feature of this problem is that for the physics and objective function under consideration, the solution to this problem is a permeability that varies smoothly between the boolean values of zero and one, with most of the domain being at intermediate permeabilities. Such a device cannot be manufactured. In this problem, then, we must somehow externally – by means of a penalty or constraint – enforce the boolean condition. The obvious way to do this is with a penalization on any non-boolean values, but that is easily seen to be non-convex, and indeed, in practice it is found that such a penalization introduces numerous artificial local minimizers.

An improved formulation was introduced by Cunha in his thesis [4]. Cunha penalizes the *slope* of a level set function, which, as will be discussed below, allows

*University of Chicago, aterrel@uchicago.edu
†Sandia National Laboratories, krlong@sandia.gov

FIG. 2.1. *Example of a domain Ω with the black representing the exterior $\mathcal{E}$ and the white the interior $\mathcal{I}$.*

fine control of the transition region where non-boolean values can occur. We will refer to this method as the slope penalty method. Penalizing the gradient of a function is less restrictive than penalizing the function itself, so that we expect (and find) fewer artificial local minimizers with this method than with a direct penalization of non-boolean permeabilities. However, the formulation still does have artificial minimizers, and it is difficult to choose a penalty parameter loose enough to avoid these yet tight enough to enforce the boolean condition.

We have thus introduced another method, which we call the slope barrier method, in which a certain inequality constraint (to be described below) is imposed on the level set function and its slope. This inequality constraint is an even looser restriction on the problem, suggesting less problem with local minimizers, yet it can more strongly enforce the boolean condition.

In this paper we will compare the slope penalty and slope barrier method on model shape-matching problems having the difficult characteristics of minimum-dispersion microflow. In section 2 we show a level set formulation of the shape optimization problem. We motivate and describe the two methods to be compared in sections 2.1 and 2.2. We give some details in order to implement and optimize the objective function in section 3. Finally we give testing methods and results of our comparison in sections 4 and 5.

2. Formulation of the Problem. The formulation of our problem is the same as outlined in Cunha's thesis [4]. For topology optimization, we have a domain, Ω, in our case this is a square array of pixels, and want to partition it into two different subsets: the exterior $\mathcal{E}$ and the interior $\mathcal{I}$, see Figure 2.1. The interior, $\mathcal{I}$, is used to represent portion of Ω that will be the domain of a PDE equation such as the flow in our microfluidics application. The exterior, $\mathcal{E} = \Omega - \mathcal{I}$, is the rest of Ω. The interface between the two regions is given by the zero contour of a level set function, $\phi(\mathbf{x})$,

$$\phi : \mathbf{x} \in \Omega \to \mathbb{R}$$

The level set function defines a geometric shape one dimension higher than Ω, for our purposes $\phi(\mathbf{x}) \geq 0$ indicates $\mathbf{x} \in \mathcal{E}$, and $\mathbf{x} \in \mathcal{I}$ otherwise. Thus our shape is determined by the distribution χ:

$$\chi(\phi) = \begin{cases} 0 & \text{if } \phi < 0 \\ 1 & \text{otherwise} \end{cases}$$

To reach the goal of our topology optimization, we are concerned with the values of the χ distribution. Optimizing on a large boolean problem is quite hard, so we relax the problem by replacing it with a sigmoid function that maps to the interval $[0, 1]$ on the reals. The sigmoid function is a smooth interpolation of χ.

$$\sigma(\phi) = \frac{1}{2}\left(1 + \tanh\left(\frac{\phi}{\Delta}\right)\right)$$

where Δ is a given parameter. The parameter Δ gives some control over the transition of the sigmoid from values near 0 to values near 1, but this will be discussed further in Section 2.1 with the introduction of the slope penalty method.

This formulation offers several advantages over a more traditional shape optimization with parametric curves. First, the design variable that we manipulate in our optimization algorithm is ϕ and not a boundary of our mesh. This allows us to avoid remeshing because of topology changes. This also allows us to use a very rich design space without limits on the curves we produce. Finally, using a smooth sigmoid function insures that derivatives exist and give us the opportunity to use faster optimization algorithms.

While there are several advantages with this formulation, the problem is ill posed. The problem statement uses only the zero contour of the level set function ϕ, yet there are infinitely many level set functions that have the same zero contour. Second, the level set function has no restriction to be smooth. It could be a zig zag of an infinite number of sinusoidal functions or anything else as perverse. By adding some regularization to ϕ, we are able to guide the optimization to use certain types of functions. We use a Tikhonov regularization for ϕ, to produce a smooth function [1]. For our applications this regularization will be added by the term:

$$\frac{\alpha_1}{2}\int_\Omega |\nabla\phi|^2 d\Omega$$

We add another term for the regularization of σ. In our microfluidics application, it also was important that we have a smooth shape in order to have a shape that is manufacturable. To achieve this we use a total variation diminishing regularization on σ:

$$\alpha_2\int_\Omega (\nabla\sigma^2)^{\frac{1}{2}}$$

The total variation diminishing regularization is used because Tikhonov is restrictive of jumps in slope [1].

2.1. Slope Penalty Method. Because we relaxed χ to a sigmoid function, there is a range for which our results will not be near 0 or 1. Let us call this range the bandwidth, Λ. In other words Λ is the measure of the set $\{\mathbf{x}\in\Omega|\varepsilon<\sigma(\phi(\mathbf{x}))<1-\varepsilon\}$ for some given ε. This means that Λ is dependent on both Δ and $|\nabla\phi|$. To enforce the criteria that the resulting shape be as boolean as possible, we prefer to have Λ small. In order to control this at runtime we wish Λ dependent only on Δ. This could be achieved by the constraint $|\nabla\phi|=1$, but such a constraint would be inhibitive of the Tikhonov regularization. To balance between these two controls on $\nabla\phi$, we use a penalty method to give an approximate constraint,which is characterized in the objective function by adding the following term:

$$\frac{\beta}{4}\int_\Omega (|\nabla\phi|^2-1)^2 d\Omega$$

This approximate constraint is more a concession that we want $|\nabla\phi|\approx 1$. The β term allows us to control how much to enforce this approximate constraint, when $\beta>\alpha_1$ we enforce the approximate constraint more, and when $\beta<\alpha_1$ we enforce Tikhonov regularization more. This penalty gives a method to increase or decrease the bandwidth of σ at runtime with the Δ parameter, with respect to our level of Tikhonov regularization.

2.2. Slope Barrier Method. The slope penalty method controls Λ by setting a restriction of the slope of ϕ over all of Ω. However, the slope outside the bandwidth does not affect the value of σ since it will be closer to 0 or 1 than our tolerance; any work done in meeting the unit slope condition outside the transition band is therefore wasted. We therefore introduce an alternative method, in which instead of a penalty we use an inequality constraint:

$$\left(\frac{\phi}{\Delta}\right)^2 + (\nabla\phi)^2 \geq 1 - \varepsilon$$

This term strictly regulates the slope of ϕ near the zero contour yet lets it vary freely elsewhere.

To enforce this inequality constraint we use a barrier method, in which we put a singular barrier at the boundary between the feasible and infeasible regions. This lets us use an unconstrained optimization algorithm. Rather than the more conventional log barrier [7], we use a piecewise but differentiable barrier that goes to zero a small distance from the constraint surface. The sharpness of the curve for the barrier is a controlled by the weight factor γ. The larger we allow γ the larger the tail leading up to the barrier, which will possibly put some control on the size of Λ allowed.

3. Implementation. In the previous section, we have introduced ways of describing shapes in terms of a level set function, and means of regulating the smoothness and the sharpness of the boolean transition. We also need to add some measure of quality of the shape produced, *i.e.* an objective function. For testing purposes we are only going to look at matching a specified target shape, $\mathcal{E}^*$, and thus we will use a Heaviside Distance function to compare our shape with the target shape, that is:

$$d(\mathcal{E}, \mathcal{E}^*) = \frac{1}{2}\int_\Omega (\sigma - \sigma^*)^2 d\Omega$$

In addition, we will have the terms described above.

Now our full objective function consists of the following terms: a Heaviside Distance, a Tikhonov Regularization for ϕ, a total variation regularization for σ, a slope penalty term. and a slope barrier term. Or in equation form:

minimize the function $\mathcal{F}$ where:

$$\mathcal{F} = \frac{1}{2}\int_\Omega (\sigma - \sigma^*)^2 d\Omega + \frac{\alpha_1}{2}\int_\Omega |\nabla\phi|^2 d\Omega + \alpha_2 \int_\Omega (\nabla\sigma^2)^{\frac{1}{2}} d\Omega + \\ + \frac{\beta}{4}\int_\Omega (1.0 - \nabla\phi^2)^2 d\Omega$$

augmented to the inequality constraint:

$$\nabla\phi^2 + \left(\frac{\phi}{\Delta}\right)^2 \geq (1 - \varepsilon)$$

To test our different controls of the slope penalty and the barrier weight we can adjust β and γ, respectively. Since we are not concerned so much about the level of the regularization terms, we have fixed both α_1 and α_2 at 1.0×10^{-4}. Doing this we have a baseline configuration, $\beta = \gamma = 0$, for when the Tikhonov regularization will be the dominant contribution to the control of ϕ. Now we turn our attention the optimization algorithms that we used.

3.1. Local Optimization. For local optimization we use the adaptive limited-memory BFGS algorithm of Byrd and Boggs [3].

We tried to use ways of controlling ϕ and σ so that there would be few artificial minimizers, that is minimizers that are not physically significant but that have been introduced by our slope control methods. But it must be noted that there is a trade off between the matching of our shape and keeping a boolean result. Furthermore, in many problems there will also be physically significant local minimizers. We have therefore found it necessary to embed the local optimization routine in an outer global optimization loop.

3.2. Global Optimization. Generally speaking, our global optimization scheme is that upon each successful local minimization, we try to "tunnel" in a randomly-chosen direction to a lower function value. However, doing so blindly would be a hopeless task with very low probability of finding a fruitful search direction; the reason for this is that the vast majority of random perturbations one could make to the function ϕ are at high spatial frequency, having little macroscopic effect on the objective function. To find search directions giving a macroscopic effect, we favor perturbations at low spatial frequencies, which is easily done by means of a truncated Fourier series.

We generate search directions by updating our design variable ϕ as follows:

$$\phi \leftarrow \phi + \sum_{m=-M,n=-N}^{M,N} A_{m,n} e^{-i\left(\frac{\pi m}{L}x + \frac{\pi n}{L}y\right)}.$$

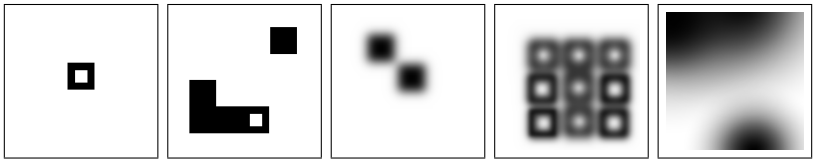
Here L is the size of one side of our square grid of pixel and we take an M and N much smaller than the number of pixels per axis. The coefficients $A_{m,n}$ are chosen at random from different points on the complex plane. This give us a new direction that is chosen from the subspace of smoothly-varying random fields.

With a direction computed in this way, we can try to tunnel to a better minimizer by doing a line search in that direction. In practice, searching even in this restricted subspace has a low probability of finding a better minimizer. Therefore, in practice we often conditionally accept an uphill step; the probability for accepting an uphill step can be determined by any convenient distribution; we use a Boltzmann factor. As in simulated annealing [8], the temperature in the Boltzmann factor can be reduced as the procedure progresses, thereby allowing frequent uphill steps early on, but more stringently rejecting them later.

3.3. Software Implementation. The code is implemented using the Sundance [6] and Trilinos [5] software packages.

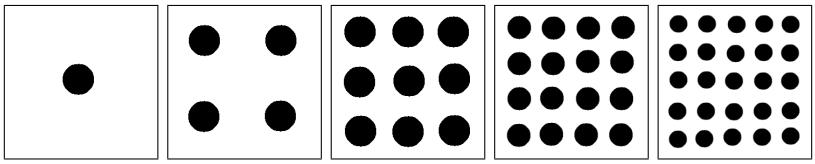
4. Testing the methods. In order to test the two methods, we start from a variety of images and attempt to match a set of target images, of different degrees of difficulty. To demonstrate some properties of the two methods, we use a host of target images, see Figure 4.1. These images provide a set of complex and simple boolean and non-boolean values to compare. One of the target images is a smooth, non-boolean picture based on the unconstrained solution to a microflow optimization problem; a challenge will be to satisfy the boolean constraint when matching this decidedly non-boolean image.

Another factor of interest is the dependence on the initial guess. We tested this property by running the tests on a series of 1x1, 2x2, 3x3, 4x4, 5x5 grids of circles, see Figure 4.2. We have discovered that to an extent the more disconnected shapes



(a) (b) (c) (d) (e)

FIG. 4.1. *The target images. (a) Simple boolean image, (b) Complex boolean image, (c) Simple non-boolean image, (d) Complex non-boolean image, (e) Smooth solution to Unconstrained Minimum Dispersion Problem.*



(a) (b) (c) (d) (e)

FIG. 4.2. *The input images. (a) 1x1, (b) 2x2, (c) 3x3, (d) 4x4, (e) 5x5.*

initially given to the algorithm, the more accurate the output thus prompting these initial shapes.

The final property studiedis the speed of convergence of the method to a better solution. For each test we see how well it is able to handle a single line search to give a local minimum and a number of additional line searches to find a better global minimum. The first local minimum gives a sense of how the method responds in a head to head comparison, whereas using a method of getting an new direction for the global search will produce quite different directionss to search for each configuration. One problem with a global search is always how long to let it run, whereas the first local line search can be give a stopping criteria based on the convergence of the objective function. For our experiments, we decided to stop the global search if one of the following conditions were met: 1) the objective function was evaluated more than 5000 times, 2) there were more than one hundred directions searched, or 3) after testing 100 directions no direction was found that further reduced the objective function. These criteria were chosen to simulate an engineer optimizing with limited time and computational resources.

5. Results. We will see from our experiments that both methods do well at matching the boolean images. As might be expected, the slope penalty method does better at matching non-boolean shapes; however, recall that the purpose is *not* to match a smooth shape perfectly, but rather to match it well while satisfying the boolean condition. We expect the output to produce an image that is both a boolean and conforms, at least in coarse outline, to the shape of the target. As we will show sometimes it is easy to fulfill either part of this requirement but not both at the same time. In our results we can differentiate the performance of the different methods by the number of iterations required to yield a given reduction in objective function.

5.1. Boolean targets. With boolean targets we see that both methods match the shape very well. The variation between methods comes in the form of global minimizers, convergence rate, see Figure 5.1, and the transition, Λ, between $\mathcal{I}$ and $\mathcal{E}$.

The results show that using only the Tikhonov regulation, see Figure 5.2, pro-

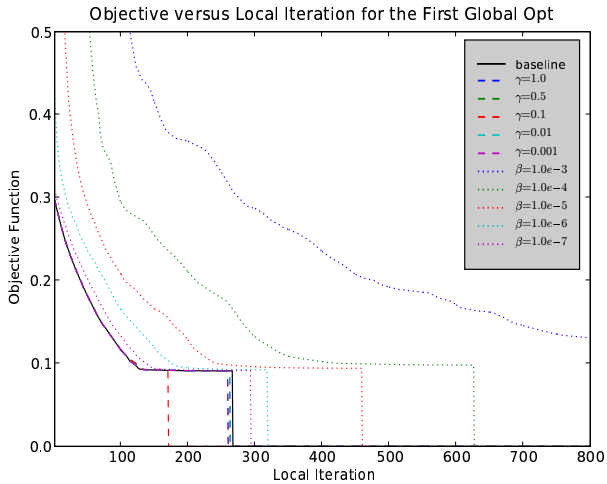
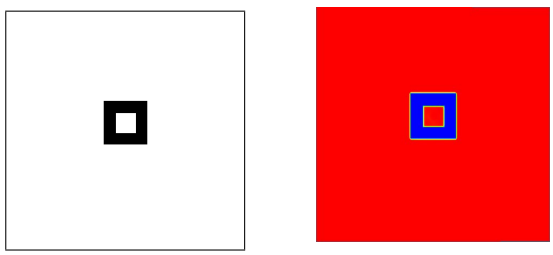


FIG. 5.1. *Sample convergence speed comparison*

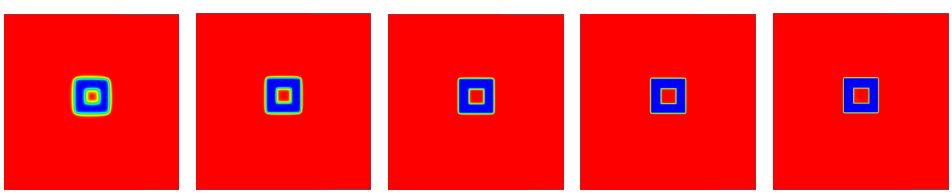


(a) target (b) baseline

FIG. 5.2. *Sample Target and baseline configuration*

duced a well matched shape and a small Λ. At the same time it did not produce a large amount of global minimizers.

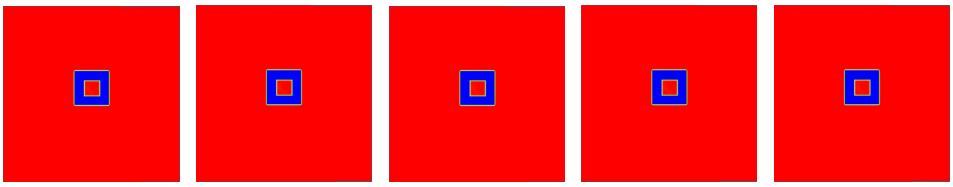
The slope penalty method, see Figure 5.3, tended to produce less shapely boolean contours. Superficially, it also appears to converge more slowly, although that is difficult to assess because the objective function is altered by the introduction of the penalty term. The slope barrier method, see Figure 5.4, successfully reproduced the sharp boolean images.



(a) $\beta = 1.0e - 3$ (b) $\beta = 1.0e - 4$ (c) $\beta = 1.0e - 5$ (d) $\beta = 1.0e - 6$ (e) $\beta = 1.0e - 7$

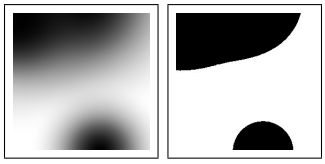
FIG. 5.3. *Sample slope penalty configurations*

5.2. Non-boolean targets. The most important test for our purposes is to be able to find good boolean approximations to non-boolean images. We use as a target



(a) $\gamma = 1.0$ (b) $\gamma = 0.5$ (c) $\gamma = 0.1$ (d) $\gamma = 0.01$ (e) $\gamma = 0.001$

FIG. 5.4. *Sample slope barrier configurations*



(a) target (b) solution

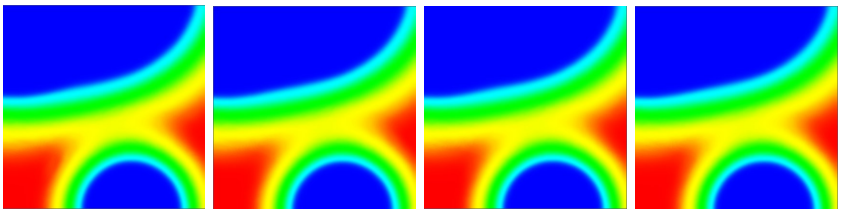
FIG. 5.5. *Smooth target and boolean solution*

image the smooth solution to our unconstrained microfluidics problem. The "true" boolean solution can be found by taking everything that is at least 50% black and giving it the value 1 and the rest 0, see Figure 5.5. This example almost reversed all the results of the boolean targets, where the narrow Λ was true to the shape of the target. We see in Figure 5.6 the results with a slope penalty method. As can be seen, it fits the target image quite well, but does *not* satisfy the boolean condition.

For non-boolean problems the slope barrier method converged more slowly than the slope penalty method and (as expected) for a small Δ it resulted in a large Heaviside distance from the boolean solution. The global optimization process also a large number of global minimizers giving the optimization even more problems to overcome. As can be seen in Figure 5.8, the solutions are, while sharply boolean, also quite ugly and plagued by local minimizers.

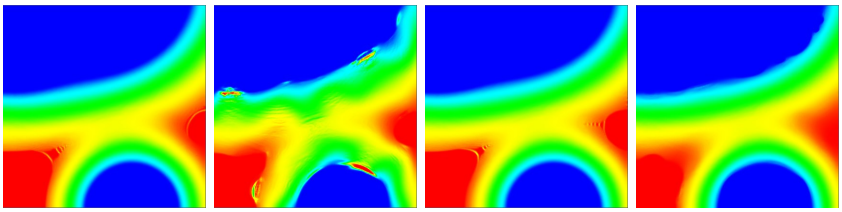
This performance showed that the small values of Δ, usually 12 pixels, that we used for the boolean targets were unsuitable for the non-boolean targets. By increasing Δ we were able to get performance closer to the slope penalty method, but this still did not result in a boolean image that we would be able to use in our application. Therefore we implemented a variable Δ in the optimization loop, in which we gradually reduce Δ after a number of outer iterations. To give a fair comparison we used a total number of global optimization steps equal to the non-variable Δ. This method did considerable better that the static version, and was able to achieve a more boolean shape that was closer to the shape of the solution, see Figure 5.9.

6. Conclusions. For boolean targets, the methods are closely comparable in performance and in effectiveness. There is a tradeoff between faithfully matching a non-boolean image and meeting a boolean constraint. The slope penalty method provides a better "fit", but does not easily meet the boolean condition. The slope barrier method does a better job of meeting the boolean condition; in our microfluidics applications that is the more important consideration. Finally, we observed that a gradual reduction in bandwidth parameter Δ is essential to getting clean results from the slope barrier method.

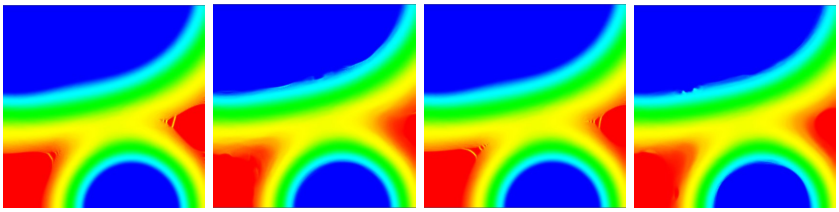


(a) 4 circles (b) 9 circles (c) 16 circles (d) 25 circles

FIG. 5.6. *Larger slope penalty ($\beta = 1.0e-4$) Final Global iteration*

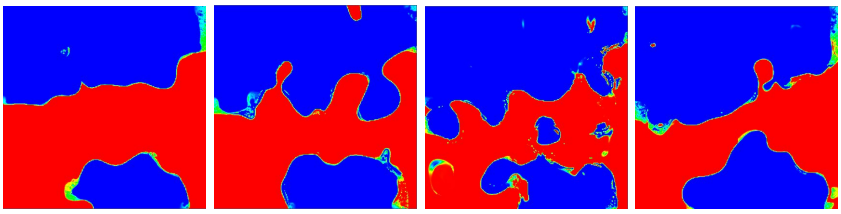


(a) 4 circles, first (b) 4 circles, final (c) 9 circles, first (d) 9 circles, final



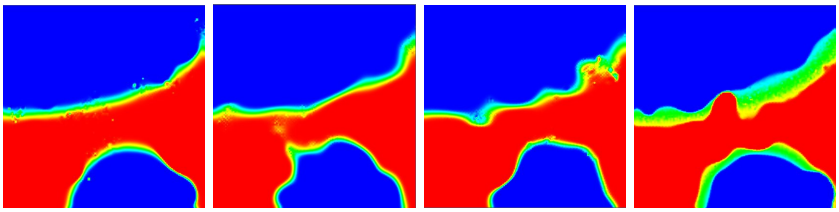
(e) 16 circles, first (f) 16 circles, final (g) 25 circles, first (h) 25 circles, final

FIG. 5.7. *Smaller slope penalty $\beta = 1.0e-6$, First and Final Global iteration*



(a) 4 circles (b) 9 circles (c) 16 circles (d) 25 circles

FIG. 5.8. *Slope Barrier $\gamma = 0.001$, Final Global iteration*



(a) 4 circles (b) 9 circles (c) 16 circles (d) 25 circles

FIG. 5.9. *Slope Barrier Reduction Method ($\gamma = 0.001$, reduces = 20, opts = 5, reduction = 0.9)*

REFERENCES

[1] V. Akcelik, G. Biros, O. Ghattas, K. Long, and B. vanBloemenWaanders, *A variational finite element method for source inversion for convective-diffusive transport*, Finite Elements in Analysis and Design, 39 (2003), pp. 683–705.

[2] M. P. Bendsøe and O. Sigmund, *Topology Optimization: Theory, Methods and Applications*, Springer-Verlag, 2003.

[3] R. Byrd and P. Boggs, *The alm-bfgs method*. A work in progress.

[4] A. Cunha, *A Fully Eulerian Method for Shape Optimization with Applications to Navier-Stokes Flows*, PhD thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, 2004.

[5] M. Heroux, R. Bartlett, V. Howle, R. Heokstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams, *An overview of the trillinos project*, ACM Transactions on Mathematical Software, 31 (2005).

[6] K. Long, *Sundance 2.0 tutorial*, Tech. Report SAND2004-4793, Sandia National Labs, 2004.

[7] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, 2000.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, 2 ed., 2002.

MULTISCALE METHODS FOR NONLINEAR GAS CHROMATOGRAPHY

GREG VON WINCKEL*, LOUIS ROMERO†, AND EVANGELOS A. COUTSIAS‡

Abstract. Gas chromatography attempts to separate the components of a gas mixture in order to perform chemical analysis. Its current appeal relates to the possibility of miniaturization and inclusion of gas chromatographs In easily portable field devices. Computational modeling of a gas chromatograph in general requires solving a nonlinear partial differential equation in three spatial dimensions and time for a geometry with highly dissimilar scales. In particular the length of a chromatograph column is typically many times larger than the average cross sectional radius which is in turn two or three orders of magnitude greater than the microscopic liquid layer which lines the inside of the column. Solving the full problem is computationally very expensive, however, the current work aims to determine the long time behavior of a sample in a chromatograph while considering only the geometry of the cross section. By combining analytical techniques such as the normal forms reduction and high-order hp curvilinear spectral element discretizations, we reduce the problem to two linear partial differential equations in two variables.

1. Introduction. Gas chromatography attempts to separate the components of a gas mixture in order to perform chemical analysis. Its current appeal relates to the possibility of miniaturization and inclusion of gas chromatographs (GCs) In easily portable field devices. Obvious application include the fast and reliable detection of chemical leaks and fast, on-site identification of chemical and biological agents. However, many theoretical issues arise as the technology is scaled down.

GCs are long capillary tubes, typically of circular or rectangular cross section, whose inner walls are uniformly coated with a retentive liquid. A carrier gas flows steadily down the tube and gas packets of the analyte are injected in its stream at some entry location and are advected along the tube. The carrier velocity has a typical laminar viscous profile, vanishing at the (unmoving) coating interface. The analyte gas particles diffuse as they are being transported, and they are adsorbed into the coating. Assuming the diffusing species are locally in thermodynamic equilibrium, a balance is established between adsorption and desorption, and the fraction of concentrations in the two phases at the interface is equal to the partition coefficient, K. The latter can be assumed to be constant for small enough analyte concentrations in the vapor phase, but in general it is a nonlinear function of the concentration at the flowing gaseous phase, C_f. Mathematically, this results in a non-linear boundary condition at the interface between the liquid and gas phases.

Under desirable operating conditions, a packet spreads while it is being transported and eventually approaches a symmetrical Gaussian profile. However, if a high enough concentration of analyte is initially injected into the column, the nonlinearity in the boundary conditions will cause the packet profile to deviate significantly from a symmetric Gaussian shape. This steepening of the Gaussian profile is referred to as column overloading in the gas chromatography community, and can be highly detrimental to optimal performance. The understanding of column overloading could play an essential role in learning how to make reliable gas chromatographs at miniaturized scales.

2. Modeling: objectives and challenges. The main purpose of our proposed analysis is to use techniques from dynamical systems theory (normal forms, inertial

*University of New Mexico, gregvw@math.unm.edu,
†Sandia National Laboratories, lromero@sandia.gov,
‡University of New Mexico, vageli@math.unm.edu

manifold reduction) to arrive at a reduced complexity description of Non-linear Gas Chromatography (**NGC**), with a particular emphasis on predicting when column overloading will occur.

Without the use of such techniques the modeling of non-linear GC columns poses serious computational challenges, stemming principally from the large number of disparate scales: the typical length of the tube, $L \approx 30m$, is much larger than the typical cross section radius, $R \approx .1mm$, which is in turn much larger than the typical coating thickness, $d \approx .1\mu m$. In addition, the spatial extent $l(t)$ of a packet grows as it is advected, so that $R << l << L$, at typical velocities of $< u > \approx 1m/sec$ introducing a third dimensionless length ratio in the problem. The spatial resolution and timestep requirements imposed by this wealth of scales are severe, as numerical stability dictates employing comparable discretizations in all spatial dimensions. Although some of these limitations may be addressable by improved numerical techniques that can circumvent some of the stability issues, full 3D simulations of the time-dependent initial-boundary value problem at the coating layer length scale can be prohibitive. On the assumption that as the packet is advected it spreads and its peak concentration becomes sufficiently low, a usual simplification is to approximate the nonlinear BC by a linear one. However, it is known that the nonlinearity can indeed affect the long-time evolution of the packet [4], and modern applications may require operating the system at overloading conditions.

Previous works have been limited to the analysis of the linear problem for simple geometries [5]. Even then, the complete analysis requires a deep understanding of the subtle phenomenon of Taylor dispersion, and can lead to rather surprising results. Due to the inherent multiscale character of the phenomenon with the balance that it implies between the very disparate diffusive and advective scales, small perturbations in geometric characteristics of the flow region over long scales can lead to large deviations in system performance [3].

The problem of determining the extent of column overloading can be stated concisely as the problem of determining the long time asymptotic behavior of an initial distribution of analyte. For the case of linear GC columns this three dimensional time dependent problem can be reduced to the problem of determining an effective diffusion coefficient D_{eff} , and an effective advection rate u_{eff} for the analyte. These quantities can be determined by solving a two dimensional eigenvalue problem involving a cross section of the GC column ([7]).

Using the theory of inertial manifolds, we have outlined a technique that we believe can rigorously predict the long time asymptotic behavior of non-linear GC columns. As with the case of linear columns, we can predict this behavior by solving the proper two dimensional problem for a cross section of the column. The solution to this two dimensional problem will yield a constant k_{eff} (in addition to D_{eff} and u_{eff} that shows that the long time asymptotic behavior of the solution is governed by the equation

$$\partial_t c + u_{\text{eff}} \partial_z c = D_{\text{eff}} \partial_{zz} c + k_{\text{eff}} c^2 \tag{2.1}$$

Once this equation in known, it is straightforward to determine the extent of column overloading.

It should be noted that the problem of determining k_{eff} is challenging both theoretically, and numerically. When analyzing real GC columns the two dimensional problem will still have disparate length scales associated with the thin liquid layer coating the walls. The technique we are proposing should still be computationally

intensive when one realizes that our purpose is not to analyze a single GC column, but actually carry out parameter studies for a whole class of GC columns. We believe that such parameter studies would be impractical without the use of such reduced order models.

3. Governing Equations. We give a brief discussion of the governing equations of NGC for the case of a straight tube. The cross section of the tube is composed of two regions: the flow region, Ω_1, separated by the boundary $\partial\Omega_{12}$ from the thin annular coating region, Ω_2, which is also bounded by the tube wall, $\partial\Omega_{wall}$. The flow down the tube is assumed steady and laminar. We ignore transverse components of the velocity. The tube has length L, radius R with aspect ratio $\varepsilon := R/L << 1$. The coordinate system is chosen with the z-axis along the tube, and the x, y axes in the transverse direction. Thus the z-component of the velocity, $u(x, y)$ satisfies the equation

$$\mu \triangle u = \frac{\partial p}{\partial z} \text{ in } \Omega_1 \tag{3.1}$$

$$u = 0 \text{ on } \partial\Omega_1 := \partial\Omega_{12} \tag{3.2}$$

where $\partial p/\partial z$ is the constant pressure head driving the flow. The velocity assumes the familiar parabolic profile in a circular tube, while for more general tube shapes the velocity needs to be solved for numerically. In general the laminar/steady assumptions are retained, but the geometrical complications associated with shapes of interest, such as tubes of slowly varying cross section or helical tubes must be accounted for in an efficient fashion.

The concentrations of a solute advected by the carrier flow (which is assumed to occur only in Ω_2) and adsorbed into the coating are described by the coupled diffusion equations

$$\partial_t C_1 + v\partial_z C_1 = D_1 \nabla^2 C_1 \text{ in } \Omega_1 \tag{3.3}$$

$$k_1 \frac{\partial C_1}{\partial n} = k_2 \frac{\partial C_2}{\partial n} \; C_2 = KC_1 + \alpha C_1^2 =: f(C) \text{ on } \partial\Omega_{12} \tag{3.4}$$

$$\partial_t C_2 = D_2 \nabla^2 C_2 \text{ in } \Omega_2 \tag{3.5}$$

$$\frac{\partial C_2}{\partial n} = 0 \;\; \text{on } \partial\Omega_{wall} \; . \tag{3.6}$$

4. Reduced order model for NGC by systematic normal form reduction. The fundamental reduction procedure that we propose for the problem is exemplified, in the simplest case where the retentive coating is absent, that is the case of classical Taylor Dispersion. Although there are many different approaches to deriving the result for Taylor dispersion, the normal form approach has the advantage that it can be extended to the non-linear case.

The first step in this approach is to expand the function $c(x, y, z, t)$ using

$$c(x, y, z, t) = \sum_{k=0}^{\infty} a_k(t, z)\phi_k(x, y) \tag{4.1}$$

where the functions ϕ_k are eigenfunctions of the reduced problem

$$D\nabla^2 \phi_k = \sigma\phi_k \; , \;\; \partial_n \phi_k|_{\partial\Omega} = 0$$

where $\Omega = \Omega_1 \cup \Omega_2$, with appropriate modifications so that some of the interfacial conditions are automatically satisfied. The expansion is carried out under the assumption

that the characteristic length R of a cross section of the tube is much smaller than the length of the tube, that is $\varepsilon = R/L << 1$.

Assuming smallness of z-derivatives, we non-dimensionalize by the scaling $z \to lz$, $(x, y) \to R(x, y)$, and we introduce

$$v(x, y)\phi_i = \sum_j \gamma_{ij}\phi_j$$

to arrive at an infinite system of coupled PDEs for the coefficients $a(t, z)$

$$\dot{a}_i + \sum_i \gamma_{ik} a'_k = \lambda_i a_i + a''_i$$

or in vector form

$$\dot{\mathbf{a}} = \Lambda \mathbf{a} + \varepsilon^2 \mathbf{a}'' - \varepsilon \Gamma \mathbf{a}' \ .$$

The mode a_0 associated with the eigenvalue $\lambda_0 = 0$ is the only mode that is not decaying. However, when determining the long time asymptotic behavior, it is not sufficient to ignore the coupling of this mode with the other modes. Instead, following the methodology of normal form theory, a sequence of near identity transforms

$$\mathbf{a} = \xi + \varepsilon^n \mathcal{P}_n \xi \ ,$$

is applied that decouples the mode a_0 from all of the other modes to any order of ε that we desire. That is, we can find a transformation that differs from the identitiy by $O(\varepsilon)$ that decouples the equation to order ε; we can then apply a transformation to the new system of equations that differs from the identity by $O(\varepsilon^2)$ that decouples these equations to $O(\varepsilon^2)$. In practice it is only necessary to decouple the equations to second order in order to get the long time asymptotic behavior. For the case of classical Taylor dispersion [1, 2], the result of this normal form reduction is to show that the first mode obeys an equation of the form

$$\partial_t c + u_{\text{eff}} \partial_z c = D_{\text{eff}} \partial_{zz} c + O(\varepsilon^3) \tag{4.2}$$

where $u_{\text{eff}} = O(\varepsilon^2)$, and $D_{\text{eff}} = 1 + O(\varepsilon^2)$.

Our goal is to carry out a similar reduction in the non-linear case, for arbitrary shaped cross sections. This is conceptually similar to the normal form reduction for the case of Taylor dispersion, except the near identity transforms now will be non-linear. In this case we will be able to decouple the mode a_0 from the rest of the modes to arrive at an equation of the form (2.1) that is valid to order $O(\varepsilon^3)$. This equation will be sufficient to predict the long time asymptotics of the nonlinear equations, and hence predict the amount of solute that can be injected before we get column overloading.

It is expected that the construction of the successive near-identity transformations will require the numerical solution of a sequence of two-dimensional boundary value problems. This is the most computationally intensive stage of this process. For each cross-sectional profile considered, such problems will need to be solved by a high fidelity method for each term of the operator $\mathcal{P}$, i.e. $\mathcal{O}(n^2)$ such problems in order to decouple the first n equations, with the optimal value of n depending on the profile.

This technique can be thought of as a variant of the Proper Orthogonal Decomposition (POD) method, which employs a Galerkin projection on empirical orthogonal

eigenfunctions. The POD method is widely used in the study of spatially extended systems [8], but generally it lacks an adequate theoretical foundation. Here, the successive diagonalization of the operator will allow us to tailor our reduced order basis to the inertial manifold describing the essential dynamics of the NGC. Our results will help provide a theoretical justification of POD-type methods for problems where a large disparity exists between the various scales, such as flows in high aspect ratio domains.

5. Numerical methods. There are two components to numerically solving the reduced governing equations, the meshing and the PDE discretization. First we perform a two-stage meshing of the domain. It is assumed that the inner and outer boundary curves can be described by parametric functions which need not be expressed analytically.

Because the outer, liquid, domain is so narrow away from corners, it is undesirable to mesh it using triangles. The reason being that for a well-conditioned stiffness matrix, triangular elements should have angles as close to 60 degrees as possible. When meshing a narrow region with triangles, one must either opt for a very large number of small elements, or use sliver-shaped triangles with one angle close to zero. Quadrilateral elements with a large aspect ratio do not similarly suffer from ill-conditioned stiffness matrices, provided the corner angles are kept close to 90 degrees.

There are complications associated with curvilinear triangles with greater than cubic mappings. For these reasons, we use curvilinear quadrilateral to mesh the liquid region where the element size is chosen such that they divide the boundary into pieces with uniform arc length. A layer of quadrilaterals is also meshed around the interior side of $\partial\Omega^I$ so that both smooth boundary curves are only bordered by curvilinear quadrilaterals. Fortunately, there exists a well-conditioned mapping for quadrilaterals considering only the $4(n-1)$ boundary nodes if we compute the coefficients of the mapping using a generalized Vandermonde matrix where the modes are constructed from the Chebyshev-Serendipity functions and the boundary nodes are the zeros of the Jacobi polynomials $P_n^{(2,2)}$ along the edges plus the four vertices. The condition of such a Vandermonde appears to grow logarithmically with the number of points. The basis functions and nodes are depicted in Figure 5.

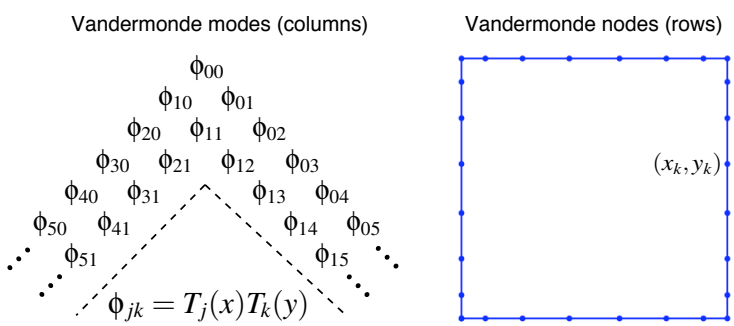


FIG. 5.1. *Modes and nodes associated with curvilinear quadrilateral mapping.*

The remaining interior space is then meshed using straight-edged triangles starting with an equilateral grid and then using r-refinement to obtain an spring force equilibrium mesh similar to the method introduced by Persson [9]. The combination of the two meshes The PDEs are discretized using a nodal hp-spectral element method [10],

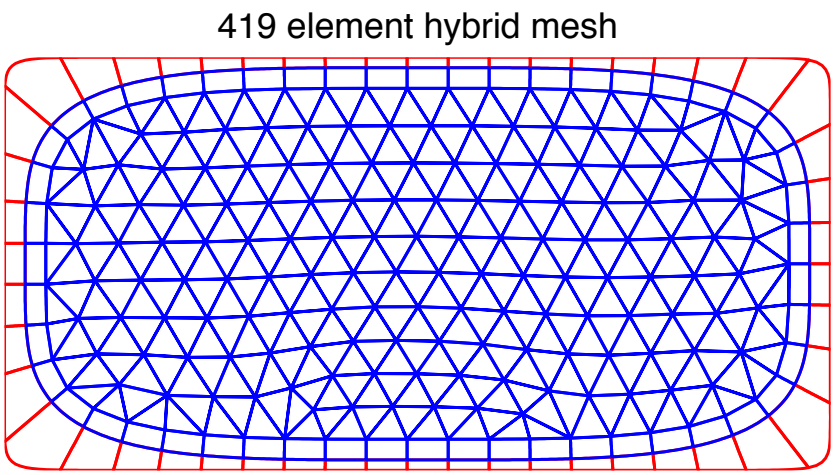


FIG. 5.2. *Sample mesh of rounded rectangular geometry where fluid region is thicker in the corners.*

where on the quadrilateral elements, the interpolation points are a Legendre-Gauss-Lobatto product grid and and on the triangles, we compute nodes to give a small Lebesgue number [11].

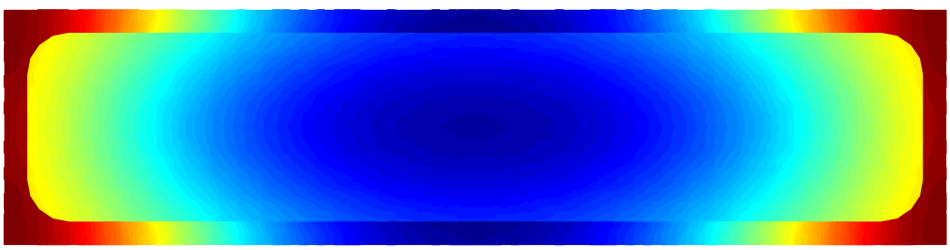


FIG. 5.3. *Example solution for the first order correction to the concentration function in a rectangular column.*

Although the methods used presently allow arbitrary polynomial orders in the Galerkin trial space, in practice quintic polynomials appear to be sufficient to resolve the solutions well beyond the precision to which the physical parameters are known for even a moderate number of elements such as in Figure 5. Figure 5 shows depicts the discontinuity in the first-order correction to the concentration across the the gas-liquid interface for a simple rectangular column cross section. Although the domain boundary appears faceted in the plot, this is an artifact or MATLAB's delaunay triangulation surface plotting where the actual numerical approximation is at least globally differentiable.

6. Conclusions and future work. We have developed analytical means of determining the critical long time behavior of the analyte in the column as well arbitrary order hybrid spectral elements for numerically solving the relevant PDEs. To date,

we have validated the numerics for the problem of simple Taylor dispersion in circular and parallel plate columns. The next phase of the work is to validate the NGC reduced order model to a full three-dimensional time-dependent code for a simple geometry and ultimately consider the situations where the column exhibits curvature or variable cross-sectional area as a function of distance.

REFERENCES

[1] G.I. Taylor *Dispersion of Soluble Matter in Solvent Flowing Slowly Through a Tube*, Proc Roy Soc Lond A, **219** no 1137, 186-203 (1953)

[2] R. Aris *On the Dispersion of a solute in a fluid flowing through a tube*, Proc Roy Soc Lond A, **235** no 1956, 67-77 (1956)

[3] G.N. Mercer and A.J. Roberts, *A Centre Manifold Description of Contaminant Dispersion in Channels with Slowly Varying Properties*, SIAM J. Appl Math, **50** no 6,pp1547-1565 (1990).

[4] A. Jaulmes, C. Vidal-Madjar, A. Laderelli, G. Guiochon, *A Study of Peak Profiles in Nonlinear Gas Chromatography 1. Derivation of a Theoretical Model*, J Phys Chem, **88**, 5379-5385 (1984).

[5] V. Balakotaiah, H. Chang, *Dispersion of chemical solutes in chromatographs and reactors*, Phil Trans R. Soc Lond A **351**, 39-75 (1995).

[6] M.J.E. Golay, *Theory of Chromatography in open and coated tubular columns with round and rectangular cross sections*, in *Gas Chromatography: 1958*, ed D.H. Desty, 36-53, Butterworths Publications LTD.

[7] L.A. Romero, M.L. Parks, J. Whiting, *Gas Chromatography in Tubes of Arbitrary Cross Section*, Preprint (2005).

[8] A. Deane, I. Kevrekidis, G. Karniadakis, S. Orszag,, Low-dimensional models for complex geometry flows: applications to grooved channels and circular cylinders, *Phys. Fluids A*, **3**(10), 2337-2354 (1991).

[9] P.-O. Perrson, G. Strang, *A Simple Mesh Generator in* Matlab, SIAM Rev, 146 no 2, 329-345 (2004)

[10] G.E. Karniadakis, S.J. Sherwin, *'Spectral/hp Element Methods for CFD*,' Oxford University Press, March 1999.

[11] T. Warburton *An Explicit Construction for Interpolation Nodes on the Simplex*, submitted to J Eng Math (2005)

The Nano-to-Micro Interface

The incorporation of nanotechnology into systems with dimensions at the micro-, and even macroscopic scales is an area of high importance. While it is clear that physical mechanisms active at atomic dimensions need specialized experimental techniques and simulation methods in order to be understood and characterized, a degree of uncertainty exists regarding how nanoscale information can be included within engineering models. In some instances, engineering quantities are ill-defined at this small scale. Furthermore, even when such translational tools have been developed, the wise practice of uncertainty quantification has either been performed in a very limited fashion, or not at all. This section presents articles describing work performed by NECIS researchers aimed at characterizing and probing the response of advanced nanoscale materials, often under applied stresses, and integrating the information obtained at the nanoscale into meaningful engineering analysis. For example, the article by Burgess, Zimmerman and Delph attempts to develop an approach towards modeling the nanoscale spatial variation of elastic moduli in materials that contain nanoscale inhomogeneities. The concept of elastic modulus in inherent to continuum mechanics, but this method, once perfected, should enable engineering materials design to be performed at the nanoscale. Another example is the article by Vanderzee, Parks and Knupp, which details work to quantify the bounds associated with the error of using the quasicontinuum method, a technique for performing coupled continuum (finite element) and atomistic (molecular statics) analyses. This article presents both accurate estimates on error bounds and new variations on the use of the quasicontinuum method that minimize these bounds and increase the method's accuracy. Through these and other papers, materials models, computational methods, and experimental capabilities have been developed to convey how physics at the atomic scale impacts material behavior within continuum and sub-continuum frameworks.

Jon Zimmerman

October 30, 2006

LOCAL ATOMIC-SCALE ELASTIC MODULI

N. BURGESS*, J. ZIMMERMAN†, AND T. DELPH‡

Abstract. This paper puts forth a methodology for predicting the elastic moduli in heterogeneous materials at the nano-scale via displacement correlation functions calculated from molecular dynamics simulations. This approach is in the process of being verified for a crystalline material modeled with the Lennard-Jones inter-atomic potential. The motivation for doing this is to characterize variations in elastic moduli at the nano-scale as well as provide a quantitative measure of material stability, which can be used to predict the formation of defects.

1. Introduction. The ability to calculate spatial variation of elastic moduli is essential for providing quantitative measures of when and where defects will occur in a material as a result of stressing the material. These defects can be inclusions, voids and crack growth. Calculating the variation of the elastic field at the atomic scale will also increase the accuracy of continuum material modeling by giving a mapping of elastic moduli that could be applied to a continuum calculation that would normally assume the elastic moduli were a constant. A method for calculating elastic moduli using molecular dynamics (MD) results has already been proposed by Meyers *et al* [3]. However, in that work the authors calculate elastic moduli representative of the entire atomic system. In this paper, we put forth a method to extract locally-defined material elastic moduli. This method is used to examine systems containing inhomogeneities. These systems are run with isothermal MD using a Lennard-Jones inter-atomic potential representative of crystal Argon.

2. Theory. A method for computing the change in elastic moduli for a continuum has been proposed by Yang [7]. We propose to extend this method to the atomic scale. We wish to determine over what length scale these variations of elastic moduli are significant. We begin by reviewing equation 6 in [7],

$$G^*_{pi}(\mathbf{X},\mathbf{x}) = G_{pi}(\mathbf{X},\mathbf{x}) - \int_\Omega \left[G^*_{pj,k}(\mathbf{X},\mathbf{r})\,\Delta C_{jksh}(\mathbf{r})\right]_{,h} G_{si}(\mathbf{x},\mathbf{r})\,dV. \tag{2.1}$$

In this relation, $G_{pi}(\mathbf{X},\mathbf{x})$ is known as a Green Function and represents the displacement in the ith direction at material point $\mathbf{x}$ due to a unit force applied in the pth direction at material point $\mathbf{X}$. For clarity, uppercase letters will be used when referring to the material point at which the force is applied (also called the source point) while lowercase letters will be used when referring to the material point at which displacement occurs (also called the field point). $\mathbf{G}$ refers to this displacement per force quantity in the homogeneous material while $\mathbf{G}^*$ refers to the material containing heterogeneities. The quantity $\Delta C_{jksh}(\mathbf{x})$ is defined as

$$\Delta C_{jksh}(\mathbf{x}) = C_{jksh}(\mathbf{x}) - C^*_{jksh}(\mathbf{x}), \tag{2.2}$$

where $\mathbf{C}(\mathbf{x})$ and $\mathbf{C}^*(\mathbf{x})$ are the material tangent moduli evaluated at material point $\mathbf{x}$ in the homoegeneous and hetergeneous materials, respectively.

Equation (2.1) is known as Dyson's equation, which, upon application of the

*Lehigh University, nburgess3@mail.gatech.edu
†Sandia National Laboratories, jzimmer@sandia.gov
‡Lehigh University, tjd1@lehigh.edu

Divergence Theorem and discretization, becomes

$$G^*_{pi}\left(\mathbf{X}^\beta,\mathbf{x}^\gamma\right)-G_{pi}\left(\mathbf{X}^\beta,\mathbf{x}^\gamma\right)=\frac{\Omega}{N}\sum_{\alpha=1}^{N}G^*_{pj,k}\left(\mathbf{X}^\beta,\mathbf{r}^\alpha\right)\Delta C_{jksh}\left(\mathbf{r}^\alpha\right)G_{si,h}\left(\mathbf{x}^\gamma,\mathbf{r}^\alpha\right)dV. \tag{2.3}$$

In (2.3), the Green's Functions $\mathbf{G}$ and $\mathbf{G}^*$ are now referred to as Lattice Green's Functions as they are defined as the tensor field quantifying the displacement of atom γ due to a unit force applied to atom β. It was shown by Lajzerowicz and Dobrzynski [2] that Lattice Green's Functions can be calculated within a molecular dynamics simulation using the expression

$$G_{ij}\left(\mathbf{X}^\alpha,\mathbf{x}^\beta\right)=\frac{\left\langle U_i^\alpha u_j^\beta\right\rangle}{kT}, \tag{2.4}$$

where $U_i^\alpha\equiv X_i^\alpha(t)-X_i^\alpha(0)$, $u_j^\beta\equiv x_j^\beta(t)-x_j^\beta(0)$ and $\langle A\rangle$ refers to the time average of variable A .

While Dyson's equation is formulated at the atomic scale in (2.3), quantities such as stress and elastic constants are known to be ill-defined at such scales. We can recast (2.3) as

$$G^*_{pi}\left(\mathbf{X}^b,\mathbf{x}^g\right)-G_{pi}\left(\mathbf{X}^b,\mathbf{x}^g\right)=\frac{\Omega}{N}\sum_{a=1}^{N}G^*_{pj,k}\left(\mathbf{X}^b,\mathbf{r}^a\right)\Delta C_{jksh}\left(\mathbf{r}^a\right)G_{si,h}\left(\mathbf{x}^g,\mathbf{r}^a\right)dV, \tag{2.5}$$

where 'a', 'b' and 'g' now refer to finite-sized cubic volumes, or cells, that contain an arbitrary number of atoms. The size of these cells (and the number of atoms contained within them) can be adjusted such that variations in Green's Functions and elastic moduli behave as smoothly varying, continuous functions. It can be easily proven that these "coarse-scale" Green's Functions can be defined as

$$G_{ij}\left(\mathbf{X}^a,\mathbf{x}^b\right)=\frac{\left\langle U_i^a u_j^b\right\rangle}{kT}, \tag{2.6}$$

where $U_i^a=\frac{1}{N_a}\sum_{\alpha=1}^{N_a}U_i^\alpha$, $u_j^b=\frac{1}{N_b}\sum_{\beta=1}^{N_b}u_j^\beta$, and N_a and N_b are the number of atoms that lie within cells 'a' and 'b', respectively.

For this work, the coarse-scale Dyson's equation is solved to determine elastic moduli differences within a face-centered-cubic (FCC) material. For all cubic materials, the elastic moduli tensor only contains three independent elements and is known to have the form

$$\Delta C_{jksh}=\Delta\lambda\delta_{jk}\delta_{sh}+\Delta\mu\left(\delta_{js}\delta_{kh}+\delta_{jh}\delta_{sk}\right)+\Delta\mu'\delta_{jksh}, \tag{2.7}$$

where $\Delta\lambda=\lambda-\lambda^*$, $\Delta\mu=\mu-\mu^*$, $\Delta\mu'=\mu'-\mu'^*$, δ_{ij} is the conventional Kronecker delta, $\delta_{1111}=\delta_{2222}=\delta_{3333}=1$ and $\delta_{ijkl}=0$ otherwise.

Combining (2.5)and (2.7) gives a total of $3N$ unknowns. However, (2.5) makes it clear that there are $6N^2$ equations. Therefore the system is inherently overdetermined. Thus, the Least Squares Method [5] is applied in the following way to give $3N$ equations for $3N$ unknowns:

$$[A]\cdot\{x\}=\{b\} \tag{2.8}$$

$$[A]^T[A]\cdot\{x\}=[A]^T\{b\} \tag{2.9}$$

Once this has been applied, Gaussian Elimination with partial pivoting is implemented to invert the coefficient matrix and get the values of the elastic moduli.

3. Results. Using the code ParaDyn [4], isothermal MD simulations of crystal Argon (lattice parameter = 5.31Å) were run. These systems were set at a temperature of 20K, approximately 29% of its melt temperature. A shifted, truncated form of the Lennard-Jones potential [1, 6] was used for the underlying material model,

$$\phi(r) = \phi_{LJ}(r) - \phi_{LJ}(r_c) - (r - r_c)\frac{\partial \phi_{LJ}}{\partial r}|_{r=r_c}, \tag{3.1}$$

where

$$\phi_{LJ}(r) = 4\varepsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right). \tag{3.2}$$

The potential cutoff distance, r_c, is set to 8.52055Å. The total system consists of 3,375 unit cells of Argon (15 x 15 x 15), a total of 13,500 atoms. For the homogeneous system, $\varepsilon = 0.012023$ eV was used for all atomic interactions. For the heterogeneous system, the central unit cell was designated to contain "stiff" Argon-like atoms, as compared with the "soft" atoms of the surrounding material. For stiff-stiff atomic interactions $\varepsilon = 0.018034$ eV, while for stiff-soft atomic interactions $\varepsilon = 0.014725$ eV (the geometric mean of the soft-soft and stiff-stiff ε values). The value of $\sigma = 3.40822$Å was used for all atomic interactions.

After bringing the system to equilibrium, atomic positions were output every 20 time steps for 200,000 time steps. At a time step of .001 picoseconds this totals to 200 picoseconds, which should allow for a very large sampling of the phase space of the lattice. Using the positions as a function of time, the coarse-scale Greens Functions are generated using a special program developed by the lead author. The spatial cells used for volume averaging are constructed at the same scale as the FCC lattice unit cells (5.31Å on a side), and thus contain approximately four atoms each.

For a homogenous system the on-diagonal Greens Functions (G_{11}, G_{22} and G_{33} are expected to behave as 1/r from the point of load application. Figure 3.1 indicates this behavior for the G_{11} component. For this case, the source point is considered to

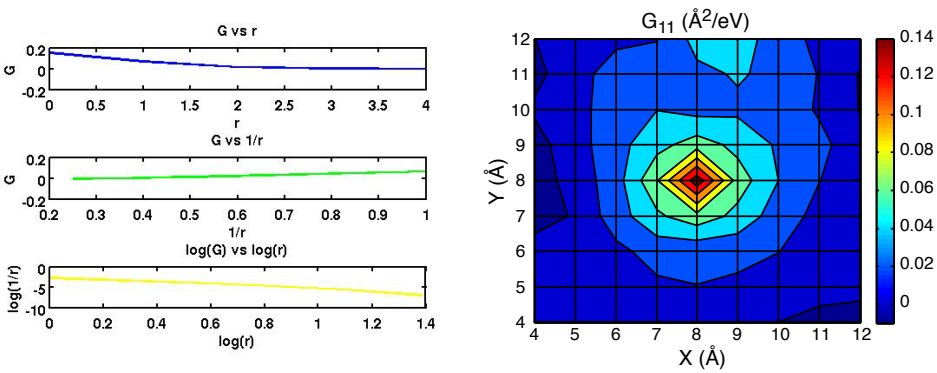


FIG. 3.1. *Left: Line plots for G_{11} with the source cell at the system center. Right: 2-D contour plot for G_{11} with the source cell at the system center.*

be the central cell. We also anticipate symmetry of the Green's Functions along any crystallographic directions. Figure 3.1 also shows a two-dimensional contour plot of G_{11} for the homogeneous system evaluated on the $z = 0$ plane, which indicates that there is symmetry with respect to the 100 and 010 crystallographic directions. It was

verified that G_{22} and G_{33} components displace identical behaviors as well as similar magnitudes.

Figures 3.2 shows similar line and 2-D contour plots of G_{12} for the homogeneous system where the source point is the center cell. These plots no longer display 1/r

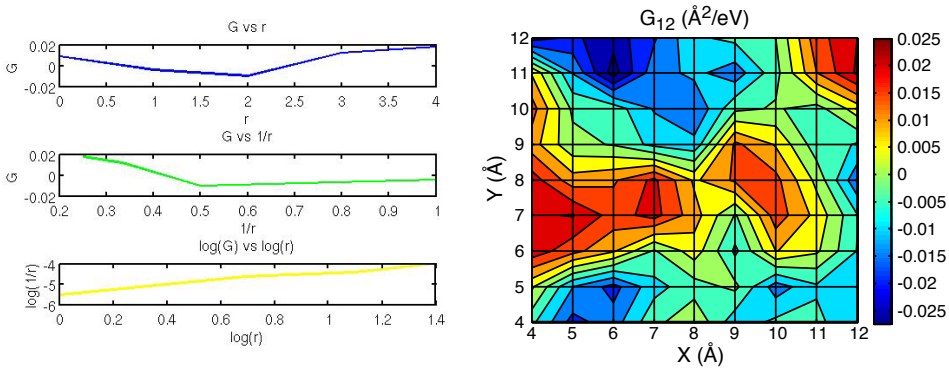


FIG. 3.2. *Left: Line plots for G_{12} with the source cell at the system center. Right: 2-D contour plot for G_{12} with the source cell at the system center.*

behavior or symmetry with respect to the cubic directions. The reason for this is unknown, but will be investigated in future work. Also note that the magnitude of the G_{12} is an order of magnitude lower than the G_{11}. This may be attributable to a Poisson-like effect, but needs to be investigated further.

For a heterogeneous system it is unknown how the Greens Functions will behave. However, we would expect that they will differ only slightly from the homogenous as we have assumed that the change in elastic moduli goes to zero on the boundary of the system. In order for this to be true we expect that the heterogeneous Greens Function be similar to the homogenous Greens Function at and near the boundary. Figure 3.3 shows line and 2-D contour plots for G^*_{11}, confirming this expectation. Note that the symmetry of the homogeneous Greens Function (G_{11}) with respect to

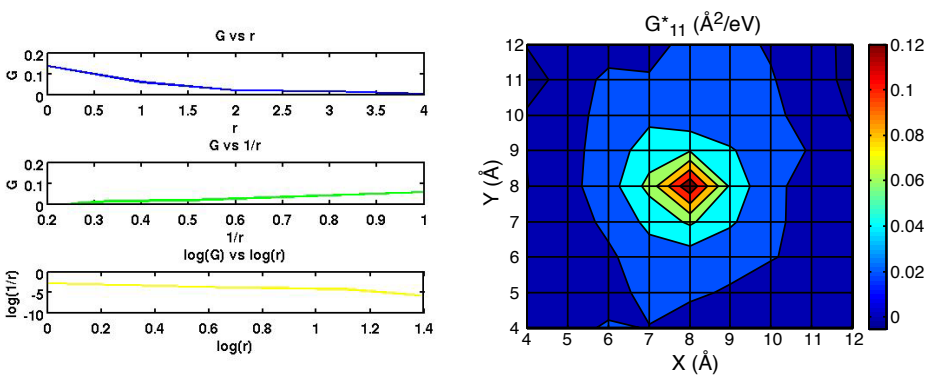


FIG. 3.3. *Left: Line plots for G^*_{11} with the source cell at the system center. Right: 2-D contour plot for G^*_{11} with the source cell at the system center.*

the cubic directions no longer holds. Also, we notice that on the outer edges of the domain, both the heterogeneous and homogenous Greens Function are approximately

zero.

The method for computing the Lattice Greens Functions has been proven to produce reasonable results which leave us to go about the business of solving Dysons equation. The verification of the lattice Greens Function is a crucial step towards getting reasonable results for the local elastic moduli. In solving Dysons equation, (2.5), we have assumed that only the cells surrounding the region of stiffer material will be affected by its presence and have non-zero values of $\Delta\lambda$, $\Delta\mu$ and $\Delta\mu'$. Using a partial pivoting Gaussian elimination technique, values for the elastic moduli were found for the 27 volumes (cubes) surrounding the defected cell.

Unfortunately the results of the solution of Dysons equation are inconclusive thus far. Equation (2.2) indicates that because the heterogeneous system contains stiffer material in the center of the domain, the change in elastic moduli for all cells should be either zero or negative. However, Figure 3.4 shows some positive changes in the elastic moduli. In addition, because the inter-atomic potential used for these

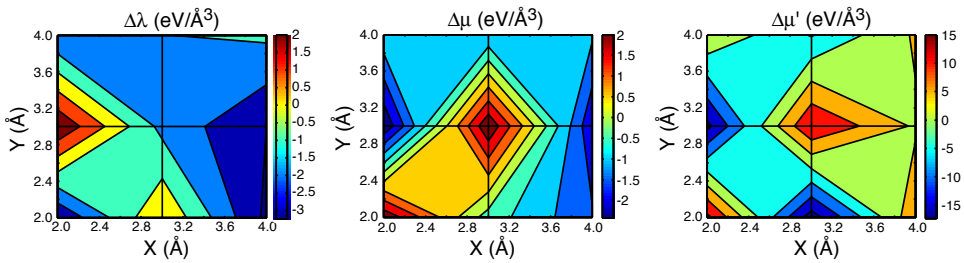


FIG. 3.4. *2-D Contour plots for $\Delta\lambda$, $\Delta\mu$ and $\Delta\mu'$.*

calculations is a summation of strictly pair-wise interactions, the results should display Cauchy symmetry: $\lambda = \mu$, $\lambda^* = \mu^*$ and $\Delta\lambda = \Delta\mu$. However, Figure 3.4 clearly does not display this symmetry.

4. Conclusions. A method for predicting spatial variations of elastic moduli at the nano-scale has been formulated, implemented and investigated. The procedure for computing the Lattice Greens Functions has been verified. However, the method for computing the change in the elastic moduli still needs improvement and verification. To date, our predictions for spatial variations in elastic moduli are not consistent with expectations based on our formulation of the problem. Our calculations show positive and negative variations in elastic moduli, while the formulation of Dyson's Equation presented in the Theory section of this paper indicates strictly non-positive variations should occur.

The error observed from our calculations could originate from a variety of sources. The calculations of the coarse-scaled Green's Functions are based on temporal and ensemble averages of displacement correlation functions. This statistical method may contain an inherent flaw. In addition, a single system size (15 x 15 x 15 FCC unit cells) and cell size (equal to FCC unit cell) were analyzed. Various combinations of system size and cell size need to be examined. It has not yet been proven that Green's Functions are continuous and differentiable at the scale studied thus far. This assumption needs to be more vigorously verified. Finally, although the line and 2-D contour plots of G_{11} shown in Figures 3.1 and 3.3 look promising, the lack of symmetry in the plot of G_{12} shown in Figure 3.2 may indicate a deficiency in the formulation of Green's Functions.

Once this method is improved, we propose that it could be used to investigate systems that possess a variety of number and types of heterogeneous defects. Also, maps of elastic moduli fields could be created for the purpose of conducting a continuum level analysis to predict material instability.

REFERENCES

[1] J.M. Haile, *Molecular Dynamics Simulation: Elementary Methods*, John Wiley & Sons, Inc., New York, 1992.

[2] J. Lajzerowicz and L. Dobrzynski, *Correlation functions of crystal atoms as a function on the distance to a free surface*, Physical Review B, 14 (1976), pp. 2695–2697.

[3] M.T. Meyers, J.M. Rickman, and T.J. Delph, *The calculation of elastic constants from displacement fluctuations*, J. Applied Physics, 98 (2005), p. 066106.

[4] S.J. Plimpton, *Paradyn, Sandia National Laboratories: http://www.cs.sandia.gov/∼sjplimp/download.html#pd*, 2006.

[5] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, and M. Metcalf, *Numerical Recipes in Fortran 90*, Cambridge University Press, Cambridge, Great Britain, 1996.

[6] D.C. Rapaport, *The Art of Molecular Dynamics Simulation*, Cambridge University Press, Cambridge, Great Britain, 1995.

[7] B. Yang, *Defect green's function of multiple point-like inhomogeneities in a multilayered anisotropic elastic solid*, J. Applied Mechanics, 71 (2004), pp. 672–676.

PERIDYNAMIC FRAMEWORK FOR SIMULATION OF CRACKS

JOSEPH RIENDEAU* AND RICHARD LEHOUCQ†

Abstract. Peridynamics (PD) is a reformulation of continuum mechanics where particles interact with nonlocal pairwise forces. The benefit of using nonlocal forces is that the spatial derivatives used in continuum mechanics are no longer necessary. This allows us to consider cracks without prior knowledge of the crack location. PD models have much in common with molecular dynamics (MD). Our report culminates by demonstrating that PD can be implemented within the MD code LAMMPS.

1. Introduction. Continuum mechanics makes several fundamental assumption. Some of these assumptions include:

- A 'point' has continuous displacement and one stress (where the 'point' has an associated differential area and force)
- Continuous displacement fields
- Tractions inside the body and on the surface must be compatible

Cracking is a phenomena where we can no longer assume that displacement fields are continuous. The partial differential equations of continuum mechanics require taking the derivatives of displacement, which may not exist. There are methods that reformulate the problem, such that the PDEs can still be solved, but this requires prior knowledge of crack location. In contrast, Peridynamics [2] is a reformulation of continuum mechanics where particles interact with nonlocal pairwise forces. Since peridynamics (PD) is a discrete method and traditional continuum mechanics is a continuous method, we consider some of the differences between discrete and continuous mechanics.

In the continuum mechanics view, the body has no sub-divisions. This body consists of differential volumes, where properties are transferred among the differential volumes by the differential equations that govern the properties. Decomposing the body into sub-volumes creates elements that have densities of properties that do not change based on length scale. The discrete mechanics model of the body is a set of particles connected to each other by potentials. Densities of properties change based on length scale for discrete systems. For example as the length scale approaches 0 the mass density of a discrete system is either infinity or 0.

The concept of stress lends itself well to continuum mechanics. Newton's second law applied to a bar element for example is

$$\sigma_{ext} A_{xs} + A\sigma = 0, \tag{1.1}$$

where σ_{ext} is the external force per unit area (stress), A_{xs} is the area the external stress is applied to, A is the internal area, and σ is the internal stress. Then a constitutive relation can be used to relate displacement to stress. In a discrete view the concept of stress is more abstract. Each particle is a point so a force per unit area is not defined. A pseudo stress can be found by placing a surface in a body and calculating all of the forces that pass through the surface among particles.

2. Peridynamic Framework. Now we will consider the application of PD to a 1D system of particles. First we note the major differences between PD and MD. A PD material has two configurations: the reference configuration, which will generally

*Renssalaer Polytechnic Institute, riendj@rpi.edu
†Sandia National Laboratories, rblehou@sandia.gov

be its unstressed state; and the current configuration, which will change with time. In PD the potential is a function of both of these configurations. In MD potential is generally a function of only the current configuration, but there are some potentials, like bonds, that depend on natural lengths as well. In PD if two particles do not interact in the reference configuration, they will never have an interaction (unless these particles collide). Two particles that initially interact will no longer interact if they separate beyond a horizon, or cut-off region. When this occurs these two particles will no longer interact (unless they collide). Furthermore cutoff distances in PD are determined in the reference configuration, whereas in MD cutoff distances are used in the deformed configuration. Another difference is potentials used in PD do not have to be true inter-atomic potentials. PD, a continuum theory, simulates the behavior of particles unlike MD that simulates atoms or bonds.

The PD equations of motion are

$$\rho\ddot{\mathbf{u}} = \int \mathbf{f}(\eta,\xi)dV' + \mathbf{b}(\mathbf{x},t), \tag{2.1}$$

where we assume that initial conditions are specified. The vector function $\mathbf{f}(\eta,\xi)$ denotes the force per unit reference volume squared exerted on a particle $\mathbf{x}$ by the particle $\mathbf{x}'$ in the reference configuration, and the vectors η and ξ denote the relative displacement and reference position. The following equations summarize these relationships

$$\begin{aligned}
\mathbf{y} &= \mathbf{x} + \mathbf{u} \\
\mathbf{y}' &= \mathbf{x}' + \mathbf{u}' \\
\mathbf{y}' - \mathbf{y} &= \mathbf{x}' - \mathbf{x} + \mathbf{u}' - \mathbf{u} \\
&\equiv \xi + \eta,
\end{aligned}$$

where the last vector defines the current relative position between $\mathbf{x}$ and $\mathbf{x}'$ in the Lagrangian or deformed configuration. Refer to Figure 2.1 for an illustration of these relationships. The vector $\mathbf{b}(\mathbf{x},t)$ is the loading force density, and the mass density is denoted by ρ.

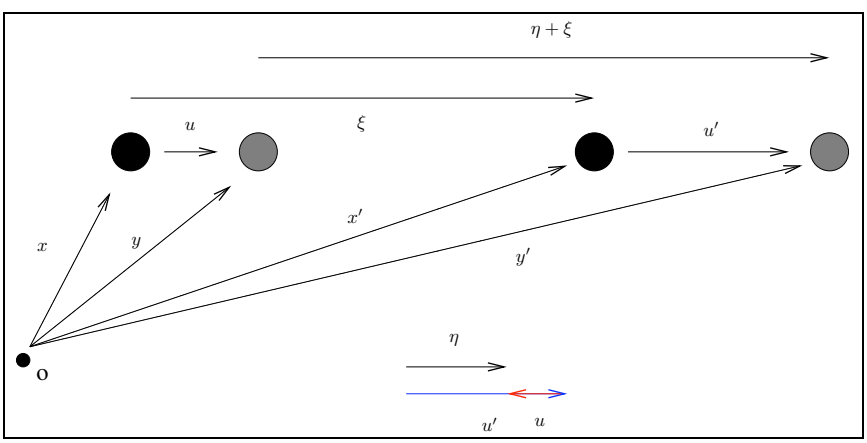


FIG. 2.1. *Nomenclature of two arbitrary particles at time* 0 *and time* t. *O is the origin of some reference system.*

In order to make this process concrete we fit a simple force equation into the pairwise force function in Equation (2.1). The potential (per unit volume squared)

$$\mathbf{\Phi}(\eta,\xi) = -\frac{\mu}{2}\left(||\eta+\xi|| - ||\gamma||\right)^2 \tag{2.2}$$

is used, where $\mu > 0$ is the stiffness per unit volume squared and $||\gamma||$ is the un-stretched length of this spring. The negative gradient of (2.2) gives a pairwise force function of

$$\mathbf{f}(\eta,\xi) = -\nabla_\eta \mathbf{\Phi}(\eta,\xi) = \frac{\mu(||\eta+\xi|| - ||\gamma||)}{||\eta+\xi||}(\eta+\xi). \tag{2.3}$$

In the equation above γ is set equal to ξ, resulting in the material being pairwise equilibrated in the reference configuration ($f(0,\xi) = 0$). This force function acts within a certain horizon δ in the reference configuration. Any two particles that separate to a distance larger than δ results in $f(\eta,\xi) = 0$.

3. Implementation of 1D Peridynamics. Our example is a bar with a uniform cross-sectional area A. This bar has 50 particles that are somewhat arbitrarily placed throughout the uniaxial length of the bar in the reference configuration. We associate a volume with each particle. We set the horizon of each particle to be

$$\delta = \frac{3L}{50}, \tag{3.1}$$

where L is the length of the bar. In other words the horizon extends *three* average particle distances in any direction. The leftmost particle is forced to have zero displacement. The rightmost particle is free. An approximation to the force density in (2.1) is

$$\int \mathbf{f}(\eta,\xi)dV' \approx \sum_{\mathbf{x}'\in\delta(\mathbf{x})} \mu\frac{||\eta+\xi|| - ||\xi||}{||\eta+\xi||}(\eta+\xi)A\Delta L' \tag{3.2}$$

where $\Delta L'$ and $\Delta V' = A\Delta L'$ are the length and volume, respectively, associated with the particle x'.

4. LAMMPS Results. Several modifications of the LAMMPS [1] code are necessary to account for the differences in PD and MD. For this simulation three changes were made in the LAMMPS code:

- The original locations of particles are stored.
- The bond harmonic potential now uses the reference configuration to determine the un-stretched length of the springs.
- During the integration step, the pairwise force function is multiplied by the associated volume.

LAMMPS takes as input: the position of the particles; the volumes associated with each particle; the initial velocities of each particle; the stiffness per unit volume squared; the density of each particle; and a list of the interacting particles in the reference configuration.

Several test cases were run to validate the results of the code. The 1D case discussed earlier was then simulated. This fairly general case was easy to input and implement in LAMMPS. The success of the simulation shows that LAMMPS is a viable tool for PD.

One might like to run a PD simulation based on data from continuum. Strictly speaking this is not possible as the PD force function contains information not typically available in continuum mechanics. If there is already a specified model for the PD force function, then μ can be specified from stiffnesses. μ is

$$\mu = \frac{\kappa}{\Delta V \Delta V'}, \tag{4.1}$$

where κ is the spring constant between particles, and $\Delta V, \Delta V'$ are the volumes associated with the two particles. If μ varies for each pair of particles, then LAMMPS requires separate μ bond types.

5. Conclusions. The similarities and differences between PD and MD have been discussed. We have demonstrated that PD can be implemented within LAMMPS. This allows someone familiar with MD to effectively simulate continuum mechanics. The requirements on the user to run PD code are small, which allows one to quickly input the required data into LAMMPS for PD simulations. Furthermore the structure of LAMMPS facilitates user modification of MD potentials or addition of new PD potentials.

6. Acknowledgements. We thank Michael Parks for his assistance in implementing the changes in the LAMMPS for PD simulations, and Steve Plimpton and Stewart Silling for helpful discussions.

REFERENCES

[1] S. J. Plimtpon, *Fast parallel algorithms for short-range molecular dynamics*, J. Comp. Phys., 117 (1995), pp. 1–19. Available at www.cs.sandia.gov/ sjplimp/lammps.html.

[2] S. A. Silling, *Reformulation of elasticity theory for discontinuities and long-range forces*, Journal of the Mechanics and Physics of Solids, 48 (2000), pp. 175–209.

CONSTRUCTION OF POLYCRYSTALLINE MICROSTRUCTURES FOR CRYSTAL PLASTICITY SIMULATION

JOEL STINSON*, ESTEBAN MARIN†, AND DOUG BAMMANN†

Abstract. This work investigates the use of various geometric tools to create accurate 2–D and 3–D digital representations of microstructures for use in crystal plasticity simulation. Materials composed of aggregates of crystalline grains are intrinsically inhomogeneous on mesoscopic and microscopic scales. In order to numerically simulate these materials accurately realistic representations of their microstructures are necessary. Existing software is used to create a 2–D mesh using an image of a real microstructure and three 3–D meshes are created and tested in finite element simulations using a developed crystal plasticity model. This work represents a mesoscale consideration of polycrystalline microstructure and is an initial step towards the use of improved digital microstructures in crystal plasticity simulations.

1. Introduction. The behavior of polycrystalline materials can be analyzed based on the consideration of their composition at various size levels. Multiscale modeling attempts to examine materials on small scales and make assumptions that can be used to simplify simulation on larger scales while still preserving accuracy. Smaller scales most accurately predict material behavior, but they also require considerably more computational complexity. Often the increased computation is impractical at smaller size scales. This work considers crystal plasticity behavior at the grain level, or mesoscale. At this level, polycrystalline materials are inherently heterogeneous due to their granular composition and resulting grain boundaries.

Mesoscale studies attempt to create representative volume elements (RVE), which can be defined as volumes which contain the minimum number of grains required to accurately predict material behavior. Anything larger and geometrically similar to a RVE should have the same macroscopic behavior regardless of how many additional grains are included. Mesoscale crystal plasticity methods explicitly model discrete grains and slip systems accounting for the anisotropy of single crystal properties and texture evolution that contribute to the anistropic macroscopic response of crystalline solids. Slip system level constitutive equations for dislocation glide kinetics and work hardening are based on phenomenological models. Compared to macroscopic methods, mesoscopic approaches are more predictive and robust since they take into account the evolution of crystallographic texture and model both anistropic elasticity and plasticity. Mesoscale theories use a large number of internal state variables to represent the material making them more computationally expensive than macroscopic plasticity models. This has limited their widespread use in the solution of system level engineering problems. However, as a mesoscale approach these theories provide a very predictive and robust theoretical framework to better understand polycrystal behavior and produce improved continuum plasticity models.

When considering materials at the grain level, accurate representation of grain size and shape, grain boundaries, orientation, and texture can have a significant impact on the macroscopic behavior of the material. Predictive capability can be improved by having initial configurations as close as possible to real grain structures when creating digital microstructures. Most experimentally obtained microstructure information is in the form of 2–D sections and thin samples. This means that basing a 2–D digital microstructure on an actual micrograph is relatively straightfoward; software

*Mississippi State University
†Sandia National Laboratories

is available to create 2–D finite element meshes directly from micrographs. However, it is more difficult to construct accurate 3–D digital microstructures when only 2–D micrographs are available. In the construction of a 3–D microstructure sample, the size and position of the grains, the crystallographic orientation of the grains, and the boundaries of the grains must all be specified. When using a limited number of 2–D samples uncertainty will be present in the construction of a 3–D RVE. As a result, 3–D digital microstructures are often created using statistical means to generate the grain structure rather than using actual micrographs.

2. Microstructure Formulation. The crystal plasticity model used in this work was formulated to describe the isothermal, quasi-static, large deformation of polycrystalline materials. The kinematics of this model are based on the elasto-plastic deformation of single crystals assuming that crystallographic slip is the dominant deformation mechanism. The deformation gradient is decomposed into an elastic and a plastic component with the elastic component being further broken down into an elastic stress tensor and a rotation tensor. The crystal constitutive equations are composed for finite elasticity and specialized for the case of small elastic strains since FCC metals usually have elastic strains that are considerably smaller than plastic strains in well-developed flow. The application of this model to the mesoscale approach used here can be considered small scale since the materials studied will have significant heterogeneity of deformation over the dimension of an aggregate of crystals. The final equations used to formulate the crystal plasticity model are shown below. A detailed derivation of this model is available [1]. The crystal plasticity model was written as an *abaqus* [ABAQUS] user material routine in order to examine the microstructures.

Small elastic strains: $\tilde{E}^e = \frac{1}{2}(V^{eT}V^e - I), \quad V^e = I + \varepsilon^e, \quad \|\varepsilon^e\| << 1$

Free Energy: $\tilde{\Psi} = \hat{\tilde{\Psi}}(\tilde{E}^e, \varepsilon_s^\alpha) = \frac{1}{2}\tilde{E}^e : \tilde{C}^e : \tilde{E}^e + \frac{1}{2}\sum_1^M \mu_{ef}\, c_\kappa \varepsilon_s^\alpha \varepsilon_s^\alpha$

Kinematics: $d = \overset{\nabla}{\varepsilon}^e + \tilde{D}^p, \quad \overset{\nabla}{\varepsilon}^e = \dot{\varepsilon}^e + \varepsilon^e\tilde{\Omega}^e - \tilde{\Omega}^e\varepsilon^e$

$w = -\mathrm{skw}(\dot{\varepsilon}^e\varepsilon^e) + \tilde{\Omega}^e + \tilde{W}^p$

Elasticity: $\tau = \tilde{C}^e : \varepsilon^e \Rightarrow \begin{cases} \mathrm{dev}\tau = \tilde{C}_d^e : \mathrm{dev}\varepsilon^e + \tilde{H}_{dv}^e \varepsilon_{kk}^e \\ p_\tau = \tilde{H}_{dv}^e : \mathrm{dev}\varepsilon^e + M_v^e \varepsilon_{kk}^e \end{cases}$

Plasticity: $\tilde{D}^p = \sum_{\alpha=1}^M \dot{\gamma}^\alpha \mathrm{sym}(\tilde{Z}^\alpha) \qquad \tilde{W}^p = \sum_{\alpha=1}^M \dot{\gamma}^\alpha \mathrm{skw}(\tilde{Z}^\alpha) \qquad \dot{\gamma}^\alpha = \Phi(\tau^\alpha, \kappa_s^\alpha)$

$\tau^\alpha = \tau : \mathrm{sym}(\tilde{Z}^\alpha) = \tau : \tilde{Z}^\alpha \qquad \dot{\varepsilon}_s^\alpha = \Theta(\dot{\gamma}^\alpha, \varepsilon_s^\alpha), \qquad \kappa_s^\alpha = \mu_{ef} c_\kappa \varepsilon_s^\alpha$

where: $\tilde{\Omega}^e = \dot{R}^e R^{eT} \quad \tilde{D}^p = R^e \bar{D}^p R^{eT} \quad \tilde{W}^p = R^e \bar{W}^p R^{eT}$

In this work, the *OOF2* [OOF2] software developed by the National Institute of Standards and Technology (NIST) was installed and used in the creation of a 2–D digital microstructure. *OOF* stands for "Object-Oriented Finite element analysis of real material microstructure" and has as its goal the conversion of an image of a hetergeneous material into a 2–D finite element mesh. With this software, a real or simulated image of grain structure is used and materials and properties are assigned to pixels of the image using various color selection algorithms. The real material micrograph we used is shown in figure 2.1 along with the resulting *OOF2* mesh with

the grains shown in different colors for clarity. This mesh demonstrates the ease by which an accurate 2–D digital microstructure can be created using a real microstructure. Unfortunately, we were not able to conduct simulations with this mesh since our crystal plasticity model is only applicable to 3–D problems. Our intention was to extrude the mesh elements a small amount to effectively simulate a 3–D mesh and allow the use of our plasticity model, but this proved to be unfeasible. However, it is apparent that the mesh closely matches the grain structure of the real material and should therefore model it accurately. Eventually, methods may be investigated to use a 2–D mesh of this type based on a real micrograph and extend it to three dimensions using statistical means to fill the volume.

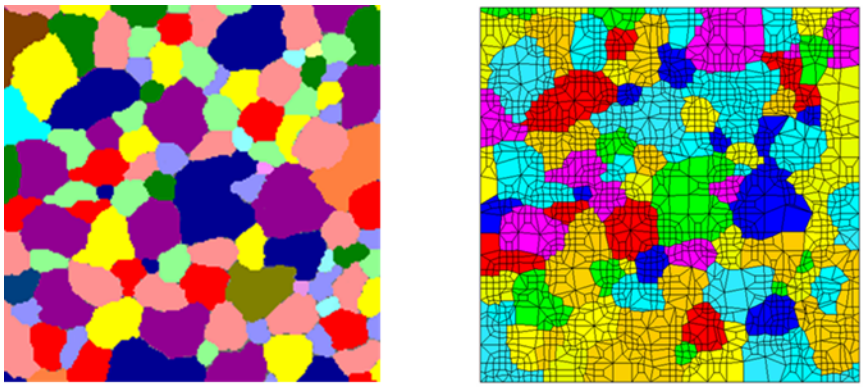


FIG. 2.1. *Real microstructure image and resulting mesh*

Three methods were used to simulate 3–D microstructures and create finite element meshes to be analyzed using the crystal plasticity model. The three meshes were given the same dimensions of 1 x 2 x 1 in the 1, 2, and 3 directions, respectively, and were created to have as near the same number of grains as possible. The mesh with the fewest grains was assigned material properties based on randomly generated Euler angles given to the *abaqus* user material routine for the crystal plasticity model. These materials were then used for the other two meshes of higher grain number with some materials repeated and assigned to the remaining grains. In this way, all three meshes were composed of materials of the same Euler angles; however, since the grain number differs slightly between the three some materials were used more often in some of the meshes.

The first 3–D RVE was created using simple cubes of brick elements in *abaqus* to represent grains. A mesh consisting of 54 cube–shaped grains composed of 2500 *abaqus* type C3D8 mesh elements was created. This mesh was intended to be the simplest, least representative digital microstructure. The brick mesh is shown, along with the other two 3–D meshes, in figure 2.2.

The second 3–D mesh was created using dodecahedra to model grains instead of simple cube elements. Twelve-sided rhombic dodecahedra of this type are often used to model materials with equiaxed grain structure due to their ability to fill a space without voids when stacked. This mesh was created using software from Cornell University. It consists of 47 grains composed of 480 *abaqus* type C3D10 elements.

The third mesh was formulated using a voronoi tesselation process and was also created using software from Cornell University. Voronoi tesselation represents an attempt to statistically simulate the structure of an aggregate of grains in a poly-

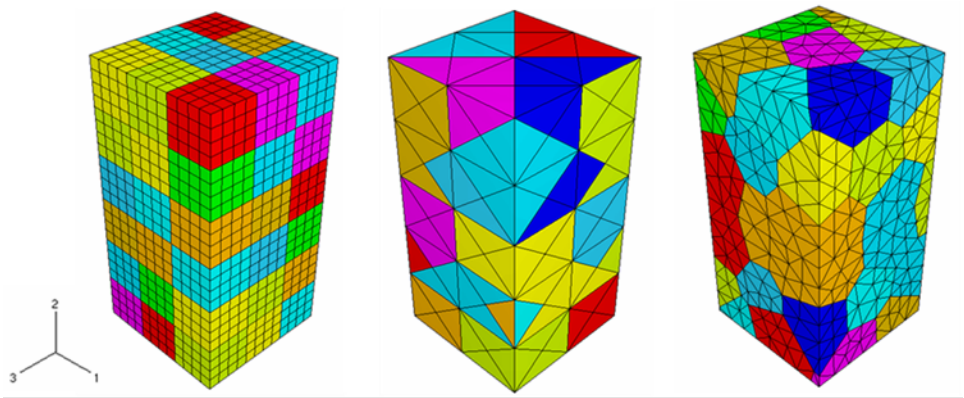


Fig. 2.2. *3–D RVE meshes (a) Brick (b) Dodecahedra (c) Voronoi*

crystalline material. Unlike the brick and dodecahedra meshes, a mesh composed of voronoi cell elements does not require all the grains to be the same size and shape. This should make the voronoi elements closer to the inhomogeneous aggregates of grains in real polycrystalline materials. The cell structure is based on a Poisson point distribution process in which points are distributed throughout the space and convex polygons and polyhedra are created around the points to fill the space with simulated grains. Each point produces a grain that can be assigned material properties and meshed to the desired resolution. The voronoi mesh used here contains 44 grains made up of 4500 *abaqus* type C3D10 mesh elements.

To compare the 3–D meshes, three loading cases were simulated in *abaqus* using the crystal plasticity model. The first case was a tensile load applied with a constant strain rate. This condition was imposed by fixing all the nodes on the lower faces in the 2 direction. One central node on the lower face was also fixed in the 1 and 3 directions to prevent rigid body movement of the mesh while allowing the base to deform as a displacement of magnitude 1 was applied to all the nodes on the top face of the mesh. An amplitude rate input was specified in *abaqus* for this displacement to simulate a constant strain rate of 1. The deformed meshes are shown for the brick, dodecahedra, and voronoi cases in figure 2.3. True stress versus true strain graphs for these cases are shown in figure 2.4.

The second loading case was a compressive load imposed by again fixing all the nodes on the lower face in the 2 direction and fixing one central node on the lower face in the 1 and 3 directions. A displacement of -1 was applied in the 2 direction to all nodes on the upper face of the meshes again specifying an amplitude rate input to simulate a constant strain rate, this time equal to -1. The resulting deformed meshes are shown in figure 2.5, and the true stress versus true strain graphs are shown in figure 2.6.

The third loading case was intended to simulate shear loading conditions by fixing all nodes on the lower faces of the meshes in all three directions and applying a displacement of 1 in the 3 direction to all the nodes on the top face. The deformed shear meshes are shown in figure 2.7, and the true stress versus true strain graphs are shown in figure 2.8.

Examination of the true stress versus true strain plots for the different 3–D meshes and loading cases shows that the brick and voronoi meshes performed similarly. The

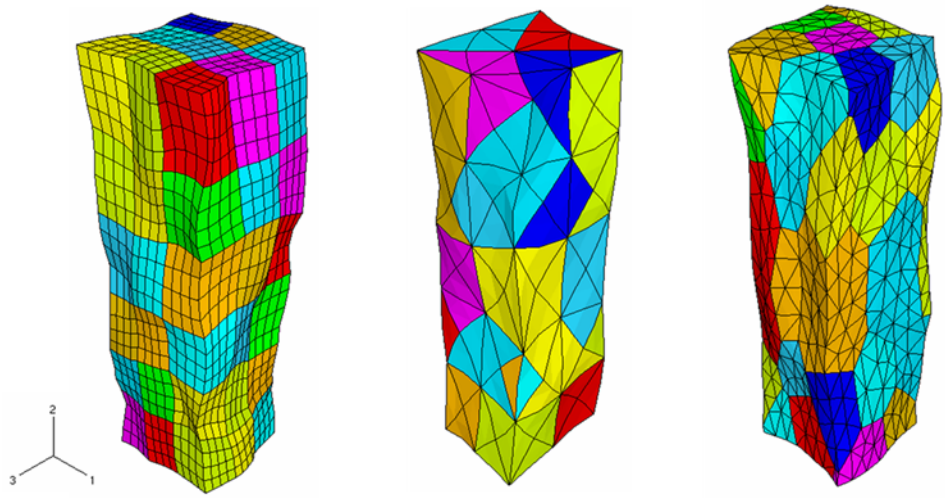


FIG. 2.3. *Deformed tensile loaded meshes*

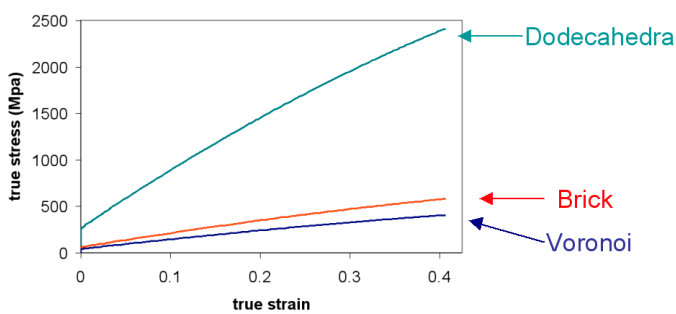


FIG. 2.4. *True stress vs true strain for tensile loaded meshes*

brick mesh showed a slightly higher true stress than the voronoi mesh in all three loading cases, but both appear to model the material's response with reasonable accuracy. The difference between the behavior of the voronoi and brick meshes demonstrates the importance of accurate microstructure representation since simply modeling the grain structure as simple brick elements changes the resulting behavior of the material simulation. The dodecahedra mesh appears to be considerably stiffer than the brick and voronoi meshes producing maximum true stress values from approximately 4 to 6 times higher. The reason for this discrepancy is uncertain and should be investigated further. The dodecahedra mesh used the same type of mesh element, *abaqus* type C3D10, as the voronoi mesh implying that this element type should not have caused this incorrect result. The dodecahedra mesh did have considerably fewer mesh elements, only 480, compared to approximately 3500 and 4500 mesh elements for brick and voronoi meshes; however, it is unlikely that this difference was the sole cause of the inaccurate behavior of the dodecahedra mesh.

3. Conclusions. Microstructure is an important factor in the determination of various properties for materials composed of aggregates of crystalline grains. On mesoscopic and microscopic scales, materials are intrinsically inhomogeneous due to the

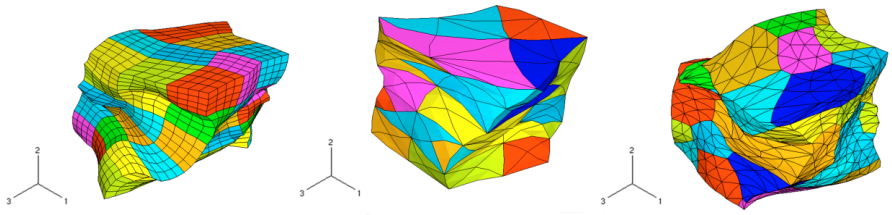


FIG. 2.5. *Deformed compressive loaded meshes*

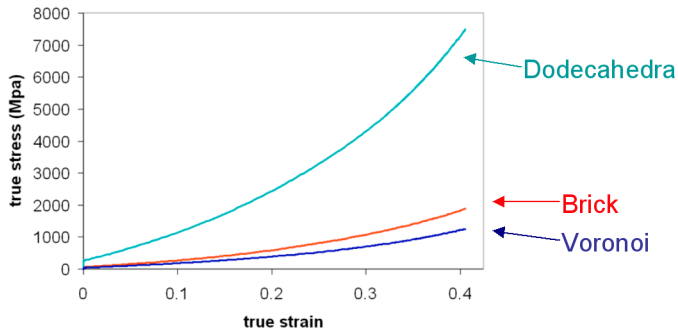


FIG. 2.6. *True stress versus true strain for compressive loaded meshes*

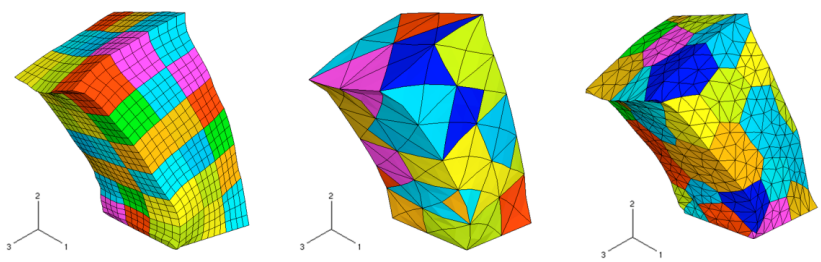


FIG. 2.7. *Deformed shear loaded meshes*

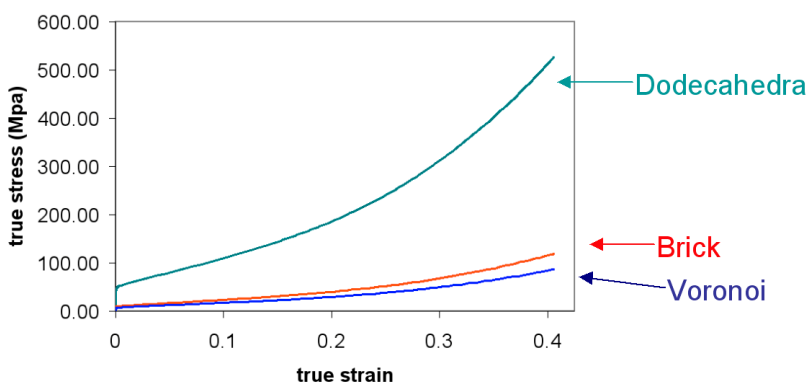


FIG. 2.8. *True stress versus true strain for shear loaded meshes*

presence of grain boundaries. Crystal plasticity theories form the basis of mesoscale approaches to materials modeling using multiscale strategies. The numerical simulation and modeling of polycrystalline materials using these theories will be more predictive if the initial microstructure configuration used resembles realistic grain structures. Realistic 2–D microstructure information is easily attained experimentally and software is available to generate 2–D RVEs based on real microstructures. We used the *OOF2* software to create a 2–D digital microstructure mesh as an *abaqus* type input file. Three different types of 3–D RVE models were also created and were investigated in *abaqus* using the developed crystal plasticity model. The brick mesh and voronoi mesh simulations showed similar behavior producing similar true stress versus true strain data under tensile, shear, and compressive loading. The dodecahedra mesh exhibited considerably stiffer behavior showing a true stress about 5 times higher than the other two meshes. The reason for this discrepancy is uncertain at this time. This work is an initial step toward the use of available geometric tools to generate and use digital microstructures in our crystal plasticity simulations and to improve continuum plasticity models through better understanding of polycrystalline material structure.

REFERENCES

[1] E.B. Marin, *On the formulation of a crystal plasticity model*, SAND report, (2006), pp. 1–160.

NUMERICAL EXPERIMENTS FOR LOCAL QUASICONTINUUM ANALYSIS

EVAN B. VANDERZEE*, MICHAEL L. PARKS†, AND PATRICK KNUPP†

Abstract. The local quasicontinuum method is an important technique for coupling atomistic-level detail with a finite element method, allowing accurate simulations to be performed on larger length scales than can currently be modeled with fully atomistic simulation. Though the local QC method is quite widely used, little work has been done to analyze its error. Here we present some numerical experiments that assess the accuracy of the local quasicontinuum method and test some of the existing mathematical analysis. We investigate some of the details of proper implementation of the local QC method in two dimensions and explore some of the mesh dependencies of the local QC error. We also suggest the novel concept of using the local QC method near boundaries, demonstrating that it may be possible to accurately reproduce the atomistic behavior without resorting to nonlocal QC or full atomistics.

1. Introduction. The local quasicontinuum (QC) method is a popular method for bridging the gap between the finite element method and nanoscale simulations of molecular statics [5]. The local QC method has been used to simulate deformation of solid materials at the nanoscale with some success, but little mathematical work has been done to analyze the error of the local QC method. Moreover, the mathematical analysis that has been done is not supported by numerical experiment. We set out to verify the bounds given in the mathematical analysis that has been done and to determine whether the actual error is mesh dependent.

2. The One-Dimensional Problem. In one of the initial papers on the numerical analysis of local QC, Ping Lin [2] bounds the error of the local QC method for a one-dimensional crystal without external forces. In Lin's model, atoms interact according to the Lennard-Jones potential

$$W(\alpha) = 4\varepsilon\left(\left(\frac{\sigma}{\alpha}\right)^{12} - \left(\frac{\sigma}{\alpha}\right)^{6}\right). \tag{2.1}$$

where σ is proportional to the lattice constant of the crystal and ε is a constant determining the strength of the the interaction [1]. The Lennard-Jones potential energy is a pairwise interaction between atoms, and the variable α represents the distance between the pair of atoms. Lin introduces the Lennard-Jones potential without the parameter ε, so we take $\varepsilon = 1/4$ when considering his analysis. For ease of computation, we take $\sigma = 1$, rescaling the units of the lattice constant as necessary.

Lin's analysis for the one-dimensional problem yields an error bound of the form

$$\|U_{i_k} - u_{i_k}\| \leq KN\sigma\left(\frac{\eta}{4}{F_c}^{-1} + {F_c}^{-5}\right). \tag{2.2}$$

U_{i_k} refers to the position of representative atom k (rep. atom k) as computed by the local QC method, and u_{i_k} is the position of the same atom as computed by the fully atomistic method. The norm, which is either the infinity norm $\|\cdot\|_\infty$ or a normalized 2-norm $(1/\sqrt{m})\|\cdot\|_2$, is taken over the vector of all rep. atoms $k = 1, \ldots, m$.

The right-hand side of (2.2) contains a variety of variables defined in Lin [2]. The constant σ is the same that appears in (2.1). $\eta = 2W'(8\sigma/5)\sigma/15 \approx 0.0262$ is a fixed constant — σ cancels out when the expression is expanded. The total

*University of Illinois at Urbana-Champaign, vanderze@uiuc.edu
†Sandia National Laboratories, mlparks@sandia.gov, pknupp@sandia.gov

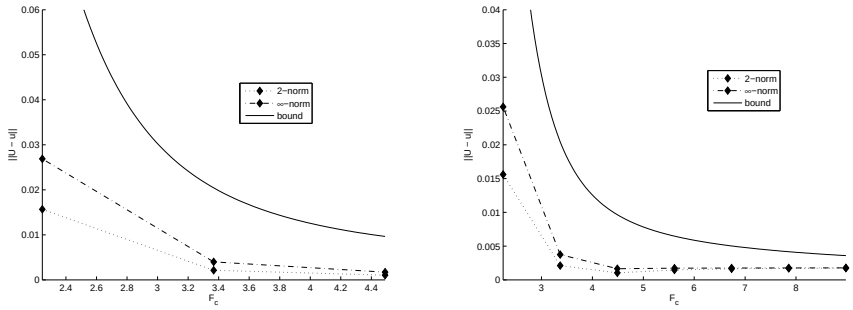


FIG. 2.1. *Local QC Error — Actual vs. Lin's Bound: (left)* $m = 12$*, (right)* $m = 6$*.*

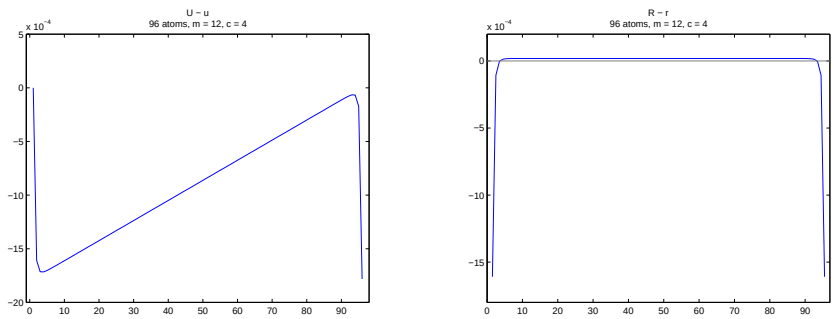


FIG. 2.2. *Actual Local QC Error*

number of atoms in the one-dimensional crystal is given by N. F_c stands for the cutoff fraction used for local QC. When each representative atom computes its energy from its $2n$ nearest neighbor atoms, $F_c \approx n\sqrt[6]{2}$. The constant K is an "arbitrary constant" that Lin uses in his analysis [2] to quantify the combined effects of several bounds. For each bound he demonstrates that some constant exists but avoids the tedium of determining precise values of the constant involved. Lin does not discuss whether problem parameters influence the optimal value of K. The analysis suggests that the optimal value of K is influenced by the particular cutoff fraction used, but there is some upper bound on K independent of cutoff fraction.

Although the mathematical analysis Lin gives of the one-dimensional local QC method in [2] is quite thorough, he does not check his calculations by performing numerical experiments. We have implemented the fully atomistic method and the local QC method in order to compare the results of the two methods and see whether the bounds he gives hold up in practice. An explicit value for K is necessary for this endeavor. In Lin's analysis $K \geq 1.557$ is reasonable. In our preliminary experiments we found that $K = 0.05$, though much smaller, was more than sufficient to bound the actual error of the method. Figure 2.1 shows the results of numerical experiments with $N = 96$. The number of elements is $m = 12$ for the graph on the left and $m = 6$ for the graph on the right. $K = 0.05$ is used to graph the error bounds.

It is worth noting that Lin's analysis only bounds the error for the position of the rep. atoms. The assumption of the local QC method is that all atoms within an element are equally spaced. This assumption breaks down at the ends of a one-

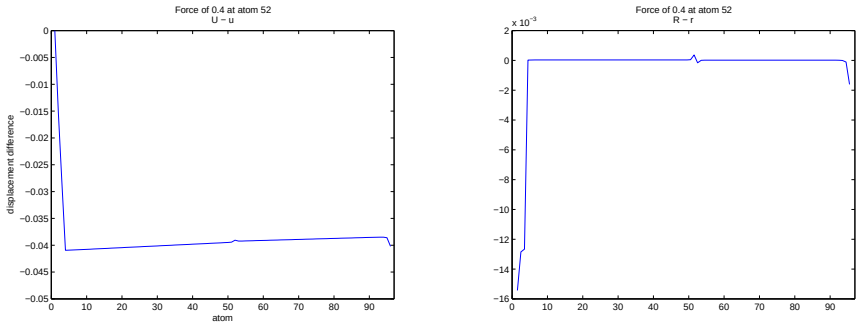


FIG. 2.3. *Actual Local QC Error in the Presence of External Force*

dimensional crystal, where surface effects come into play. Because Lin places rep. atoms at the center of the one-dimensional elements, he does not need to worry much about the surface effects. The graphs in Figure 2.2 show two different views of the error of the local QC method, using the assumption of equally spaced atoms within elements to compare the solution at all atoms. The error is computed as the value of a quantity as computed by the local QC method minus the value of the same quantity as computed by the fully atomistic method. For the left the quantity compared is the position u_i of each atoms. The right graph shows the error in $r_{i+1/2} = u_{i+1} - u_i$, the distance between adjacent pairs of atoms. This graph, in particular, shows that the greatest error is made near the surfaces of the one-dimensional crystal.

Besides being somewhat limited in bounding the error only at rep. atoms, Lin's analysis is limited in that it applies only to the case of no external forces. We are curious about how well the local QC method approximates the solution under moderate external force. Figure 2.3 shows the actual local QC errors when a force of 0.4 is applied at atom 52, one of the rep. atoms near the center of the crystal. The graphs are the same types as those in Figure 2.2. The error is significantly greater, but the local QC solution is still reasonable. It is interesting to note that the largest errors are not near atom 52, but rather near the atom fixed at the origin.

3. The Two-Dimensional Problem. Lin analyzed the two dimensional case in [3], and we wanted to check the bounds he gives for this case as well. Thus we implemented local quasicontinuum and the fully atomistic method in two dimensions.

Most of our tests were performed on a 49×49 grid of atoms. It would be good to use a larger grid for testing as well, and there are plans to do such testing in the near future, but this grid size allowed us to find solutions fairly quickly using limited computer resources. The grid, which has a spacing of $(r_0 + r_1)/2$, where r_0 is the point at which the Lennard-Jones potential reaches its minimum value and r_1 is the point of inflection of the Lennard-Jones potential, is shown in Figure 3.1 on page 197. For boundary conditions for the problem we fixed the boundary atoms of this grid in place. We also used this grid as our initial condition as well, since Lin's analysis led us to believe that there was a solution fairly near this configuration.

It should be noted that an unconstrained regular square grid (i.e., with no boundary conditions) is not a stable configuration of the Lennard-Jones potential. A triangular lattice, on the other hand, is a stable configuration. We chose to impose these boundary conditions and use this initial condition in order to satisfy assumptions of Lin's analysis. The algorithm to minimize the energy computed with full atomistic de-

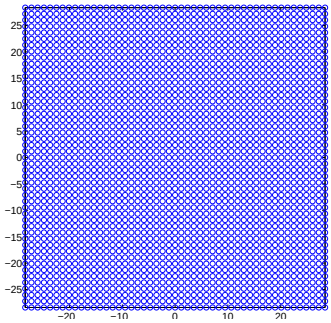


FIG. 3.1. *Initial Condition*

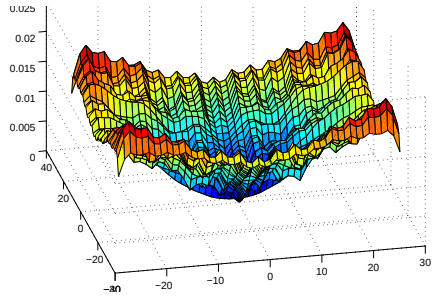


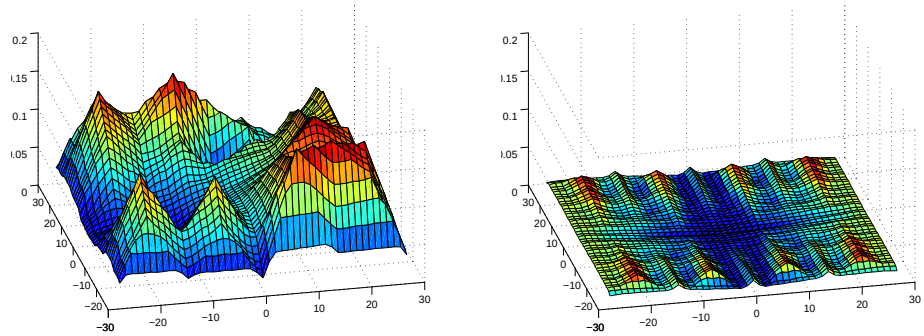
FIG. 3.2. *Deformation at Atomistic Solution*

tail, which uses Newton's method, converges to a solution in just a few iterations; the solution of the fully atomistic calculation is very near the initial condition. Looking at a graph of the location of the atoms at the atomistic solution, it is difficult to see any difference between the initial condition and the atomistic solution. The atoms do adjust a small amount, however. Figure 3.2 shows the norm of the difference between the final and initial locations of each atoms as a function of the initial position of the atoms. As one might expect, the surface effects near the edges cause the atoms nearest the edges and corners to move the most.

3.1. Partial Atoms Per Element. When initially implementing the local quasicontinuum for the two-dimensional problem, we used an algorithm that assigned each atom to a unique mesh element. The representative atom of the element was considered to represent the energy of every atom assigned to the element. Thus each representative atom represented an integer number of elements. It seemed from Lin's analysis that this is what he intended should be done.

After working with this assumption for quite some time, we discovered, however, that one can get much more accurate solutions using a fractional number of atoms per element. Using an idea derived from the quasicontinuum overview of Miller and Tadmor [4], we changed our local quasicontinuum code such that each representative atom represented a number of atoms equal to the total number of atoms multiplied by the area of its element as a percentage of the total area of the domain. The difference was remarkable. Figure 3.3 compares the norm of the error of the local quasicontinuum solution for the same mesh using an integer number of atoms (left) and a fractional number of atoms (right). The largest error is more that five times larger using an integer number of atoms than using a fractional number of atoms. The error graphed here and elsewhere in this paper is the Euclidean norm of the distance between the location of the atom at the local quasicontinuum solution and the location of the atom at the fully atomistic solution. The quantity is computed at each atom and graphed as a function of the initial location of the atom (i.e., a Cartesian grid with spacing $(r_0 + r_1)/2$).

The result was not unique to this mesh, either. For every mesh that we tried, the error was significantly smaller when using a fractional number of atoms than when using an integer number of atoms per element. In fact, for several fairly refined meshes, the local QC energy minimization algorithm would not converge when we used an integer number of atoms, but did converge when we used a fractional number of atoms. We also observed that for meshes in which every representative atom was



(a) Integer Number of Atoms (b) Fractional Number of Atoms

FIG. 3.3. *Integer vs. Fractional Number of Atoms*

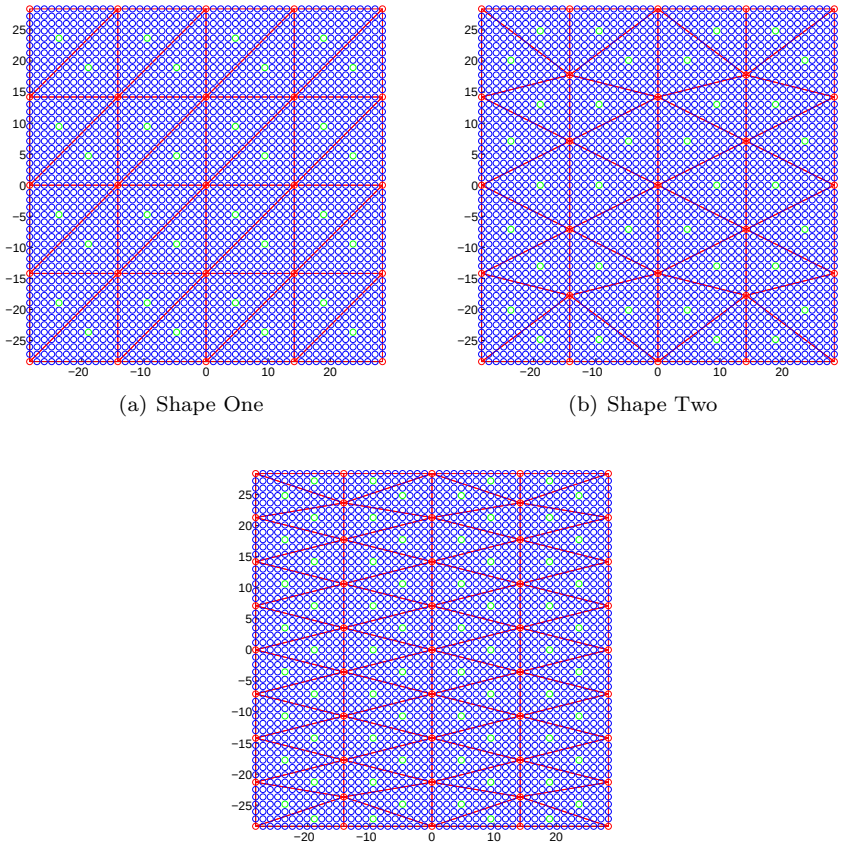
more than a cutoff radius distance from the boundary, when a fractional number of atoms was used, the uniform grid of the initial condition was the local QC solution. We expect this to be universally true. When an integer number of atoms was used, the initial condition was never the local QC solution, and the maximum error was more than five times as much as at the initial condition.

Another concern we had with using an integer number of atoms was that the local QC solution did not reflect symmetry of the underlying mesh. In contrast to the solution in Figure 3.3(a), the solution shown in Figure 3.3(b) has symmetry across the x and y axes. The underlying mesh, a more refined version of the mesh shown in Figure 3.4(b), exhibits the same symmetry. This is an example of the general observation that the local QC solution does reflect the symmetry of the mesh when the number of atoms is a fraction assigned according to the area of the mesh elements.

3.2. Mesh Dependence. One purpose of our local QC investigation was to determine whether the local QC solution depends on the choice of mesh in any significant way. Lin's error bound, though somewhat mesh dependent, does not capture the influence of the mesh. The error bound he gives can easily be bounded above by an expression that involves only the largest triangle edge in the mesh and the cutoff radius. In order to test the mesh dependence of local quasicontinuum, we tried several meshes based on each of three main triangle shapes.

The coarsest mesh used for each triangle shape is shown in Figure 3.4. Most atoms of the mesh are shown in blue, but representative atoms are green, and atoms that are nodes of mesh elements are red. The element edges are drawn in red overlaying the atomic grids. The cutoff radius we used for our tests was $3r_1 \approx 3.733$, so each representative atom takes into account about three atoms in any direction when calculating its energy. Note that for meshes one and two at the coarsest level, every representative atom is more than three atoms distant from the boundary. These representative atoms do not sense any surface effects, since their cutoff disk is completely filled with atoms.

When the meshes are further refined, however, the local quasicontinuum does take surface effects into account. The position of the representative atoms relative to the surface and the shape of the elements near the surface are important in determining how accurately a representative atom reflects the average energy of the atoms it represents. Figure 3.5 shows the norm of the error per atom at the local quasicontinuum solution for the finest meshes (twelve elements per boundary edge) of types one and



(a) Shape One

(b) Shape Two

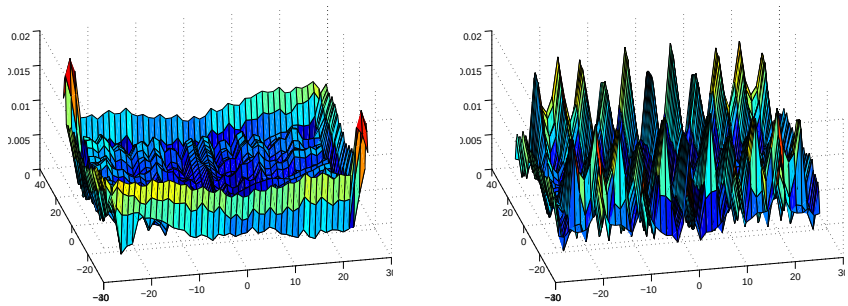
(c) Shape Three

FIG. 3.4. *Coarsest Meshes of Each Triangle Shape*

two.

The results for the mesh composed of right triangles (Figure 3.5(a)) are excellent. Except for the two corner triangles that are fixed in place at the northwest and southeast corners of the mesh, the error is quite small. The results are not as good for the mesh with nearly equilateral triangles (Figure 3.5(b)), but the results are not terrible either. These are encouraging results, and we suspect that it may be possible, particularly if the mesh and representative atoms are well chosen, to resolve surface effects with fair accuracy using the local quasicontinuum method. This would be a novel result; historically those who use local quasicontinuum have suggested that it is inappropriate to use the local quasicontinuum method near boundaries and claimed that resorting to nonlocal quasicontinuum or fully atomistic calculation is necessary.

There still is some concern about the convergence of the local quasicontinuum method, however. The results shown in Figure 3.5(b) are for a solution that is not fully converged. The particular implementation of optimization algorithm certainly influences the convergence properties, and it is possible that a different optimization algorithm would yield better results. Our local QC implementation used the conjugate gradient method to minimize the energy, and the energy near the local minimum we are trying to find is barely changing more than roundoff error. It might be more



(a) Shape One (b) Shape Two

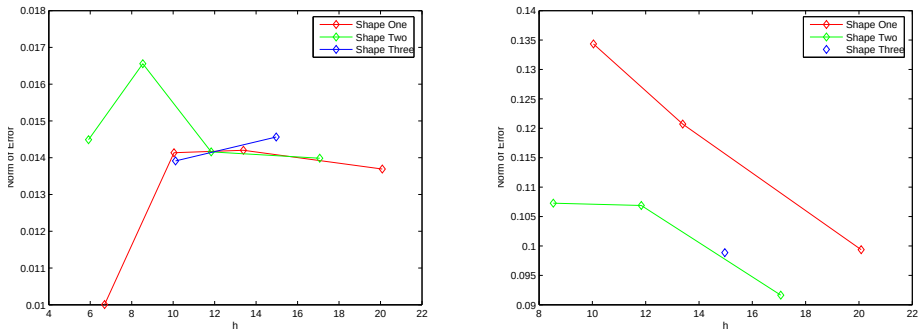
FIG. 3.5. *Results for Finest Meshes*

effective to use a more explicitly Newton-like method that seeks a zero of the gradient. The solution shown here is the iterate with minimum norm of the gradient found by the conjugate gradient method. The norm of the gradient at this point is slightly smaller than 0.01. Starting from this point and any other point we've tried, the conjugate gradient method converges to a nonphysical configuration of smaller energy than the local minimum we seek, in each case corresponding to some configuration that inverts elements and moves most interior atoms outside of the boundary atoms.

This concern about convergence of the local quasicontinuum gives pause with regard to Ping Lin's analysis, though. It may be possible to improve convergence by using a different optimization algorithm. Perhaps it would help even to change the conjugate gradient implementation to use the standard gradient modification of Fletcher and Reeves instead of the alternative formula of Polak and Ribiere, which we used here. Regardless, the difficulty in converging to the local minimum suggests that there is little local convexity near the minimum. Lin takes great care in his analysis to make assumptions that lead to the local convexity of the energy, allowing him to conclude that the minimum we seek exists and is unique. It is unclear where Lin's assumptions might break down in this case.

In any case, it is interesting to note the differences between the two solutions shown in Figure 3.5. With the mesh for shape one, the error is much smoother and somewhat less than the error for shape two. Although it's hard to see from 3.5(b), the largest errors for the shape two mesh are along the north and south boundaries, with much less error along the east and west boundaries. We do not completely understand the causes for the differences between these results, but we believe that the placement of representative atoms and the shape of the elements near the boundary are crucial. It is likely that some of the thin elements near the north and south boundaries of the shape two mesh are "overestimating" surface effects. This phenomenon is not as evident, however, in the shape three mesh result (not shown), so further investigation is necessary.

3.3. Lin's Error Bound. One goal of the study was to see how accurate Lin's bounds are for a two-dimensional material. The bound Lin gives involves several constants, and it is nontrivial to determine appropriate values for those constants. It is possible, though, to compute the error that Lin is trying to bound. Figure 3.6(a) shows the results of these computations, summarizing the error for the three different triangle shapes at different levels of refinement.



(a) Fractional Number of Atoms (b) Integer Number of Atoms

FIG. 3.6. *Summary of Local QC Errors — Lin's Error Norm*

h in the graph refers to the length of the longest edge in the mesh, shown here as length in the same units as σ. (Recall that we took $\sigma = 1$.) The error Lin uses is a Euclidean norm scaled by a factor for the number of atoms. Also, his bound is for a sum of the norm of the positional error with the norm of the error of the discrete derivatives with respect to x and y. It is interesting to note that the discrete derivatives he computes for the atomistic solution are mesh dependent. Thus even though the local quasicontinuum solution is the same (coinciding with the initial condition) for the largest two meshes for triangle shape one, the error as computed for Lin's bound is different.

The errors shown here are not unreasonable according to the bound given by Lin, especially since the constants he gives may be fairly large. These results are much less satisfying than the results in one dimension, though. As mentioned earlier, the bound Lin gives can be bounded above by a simple expression involving the largest edge of the mesh (i.e., h) and the cutoff radius. Since we have used the same cutoff radius for all our numerical experiments, this reduces to a linear function of h in our case. The results, however, are quite irregular. It is impossible to discern any sort of linear trend in the actual errors. One might suppose that Lin was assuming an integer number of atoms per element and that the error trend depends on that assumption. However, Figure 3.6(b) shows the errors for the tests we did in that case, and there is no obvious linear trend for that situation either. Furthermore, one can compare the error between the cases of using a fractional number of atoms and using an integer number of atoms, verifying the earlier claim that using a fractional number of atoms reduced the error in every case.

4. Conclusion. We have performed some interesting and enlightening experiments for the local quasicontinuum method in both one and two dimensions. The results in both one and two dimensions suggest that it may be possible to carefully use the local quasicontinuum method to accurately estimate the atomistic solution near boundaries and under moderate levels of force. This is an exciting possibility which we hope to investigate further.

Working in two dimensions it became clear that the local quasicontinuum method produces a much better result when representative atoms are assigned a fractional number of atoms according to the volume of the containing element rather than an integer number of atoms. We also notice some interesting differences between the

results that depend on the mesh that is used. We wish to study this in more depth and learn more about how the mesh used affects the accuracy of the results.

REFERENCES

[1] J.E. Lennard-Jones. Cohesion. In *Proceedings of the Physical Society*, volume 43, pages 461–482, 1931.

[2] Ping Lin. Theoretical and numerical analysis for the quasi-continuum approximation of a material particle model. *Mathematics of Computation*, 72(242):657–675, June 4, 2002.

[3] Ping Lin. Convergence analysis of a quasi-continuum approximation for a two-dimensional material. to appear in *SIAM Journal of Numerical Analysis*, 2006.

[4] Ronald E. Miller and E. B. Tadmor. The quasicontinuum method: Overview, applications and current directions. *Journal of Computer-Aided Materials Design*, 9(3):203–239, October 2002.

[5] E. B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Philosophical Magazine*, A(73):1529–1563, 1996.

The Nano/Bio Synergy

This target area supports Sandia's growing efforts in biology, leveraged by its traditional strengths in engineering and modeling. Bio-inspired solutions to physical science and engineering problems also fall within this area's scope.

We note that adding biological cells or molecules to nanosystems creates several new challenges both for experiement and modeling. First, the systems become more complex: solids plus fluids, semiconductors plus organic materials, etc. Second, new kinds of interfaces are formed and must be understood. And third, biological molecules are big, move at slower timescales, and their atomic-level interactions with other materials and water are not as well characterized as they are for traditional MEMS materials (semiconductors, metals, etc).

The papers in this section deal with various aspects of these challenges. The first paper by Hinze and Bennan describes a lock-in amplifier designed to control a micro-biodetector used to analzye molecular components of saliva. The paper by Marshall, Medlin, and Batasz describes work on metal-insulator-semiconductor sensors for hydrogen containing gases. The sensors include organic self-assembled monolayer (SAM) coatings which boost their specificity by controlling surface chemistry.

The final 3 papers are modeling oriented. Mehne and May developed a JAVA-based took for visualizing the metabolic networks of cells and the output of their dynamic simulation in the Xyce electronic circuit simulator. Mukherjee and Crozier describe rigid-body dynamics algorithms they added to the LAMMPS molecular dynamics package via RPI's POEMS toolkit to enable large protein molecules to be simulated in a coarse-grain fashion at longer timescales. Finally, the paper by Shuttleworth, Howle, Long, Templeton, and Tuminaro discusses a class of numerical methods (block preconditioners) used to speed-up the solution of the Navier-Stokes fluid equations for electro-osmotic systems. This kind of fluid flow is of particular relevance for micro-fluidic channels that transport biological molecules and cells.

Steve Plimpton

October 30, 2006

QUANTUM DOT BIOCONJUGATES FOR CANCER DETECTION USING μCHEMLAB^{TM}

O. ELBOUDWAREJ* AND V. VANDERNOOT†

1. Introduction. The use of quantum dots (QDots) as fluorophores in biomedical research is rapidly expanding due to their unique optical advantages over traditional fluorescent dyes. QDot properties include strong resistance to chemical- and photo degradation, large extinction coefficients (∼10-100x that of organic dyes) that allow for excitation at low light intensity, high resistance to photobleaching over minutes or hours, and broad absorption with narrow emission spectra [1]. This offers a significant advantage over organic fluorophores that tend to have narrow absorption spectra, broad absorption/emission profiles, and low photobleaching thresholds.

Given that *in vitro* studies have shown QDots to be relatively non-toxic in animal cells, these fluorophores can be used to study cellular activity and detect the presence of enzymes. One promising application of QDots is the detection of certain proteases that play a role in cancer progression. Specifically, cancer cells use proteolytic degradation of the extracellular tissue matrix as a means of migrating to neighboring healthy tissues; since proteases allow cancers to metastasize, high levels indicate poor clinical outcomes for patients.

The detection of the protease is done indirectly through the formation of a QDot bioconjugate. The water-soluble surface of QDots allows for binding to target molecules and stable complex formation. The molecular rearrangement created by binding of the QDot to an organic dye molecule leads to a shift in the emission profile of the QDot that indicates conjugation. The presence of a protease that cleaves the peptide linkage between the two fluorophores would diminish the emission shift over time and create free QDot molecules that would fluoresce at their conventional wavelengths. The narrow emission spectrum of free QDots minimizes overlap with the conjugate compound and allows greater sensitivity for protease detection.

In order to really make use of the Qdot approach for protease profiling and cancer staging, we need to have an instrument to assay for protease activity. The chemical-analysis system being used to detect QDot fluorescence is Sandias μChemLab^{TM} since the natural fluorescence of these agents makes a fluorescent platform like the μChemLab^{TM} ideal (see Figure 1.1). This microfluidic system is smaller than bench-top-sized CE instruments and its portability allows for bedside medical diagnosis of patients. The advantage of the μChemLab^{TM} for protease detection is that as little as 5-10 μL of sample are required for analysis and the speed of the sample analysis (the μChemLab^{TM} has the ability to run currents of up to 100 μA at 5000 V) makes it possible to get accurate results for protease activity without worrying about the effects of enzyme denaturation at room temperature [2]. Unlike commercial benchtop-sized microfluidic instruments that require the microfluidic chips to be replaced after several analyses, there is no limit to the number of analyses that can be run on theμChemLab^{TM} on a daily basis. All of these μChemLab^{TM} characteristics make the QDot approach to cancer detection a practical technique in the real world.

2. Materials and Methods.

*University of California at Berkeley
†Sandia National Laboratories

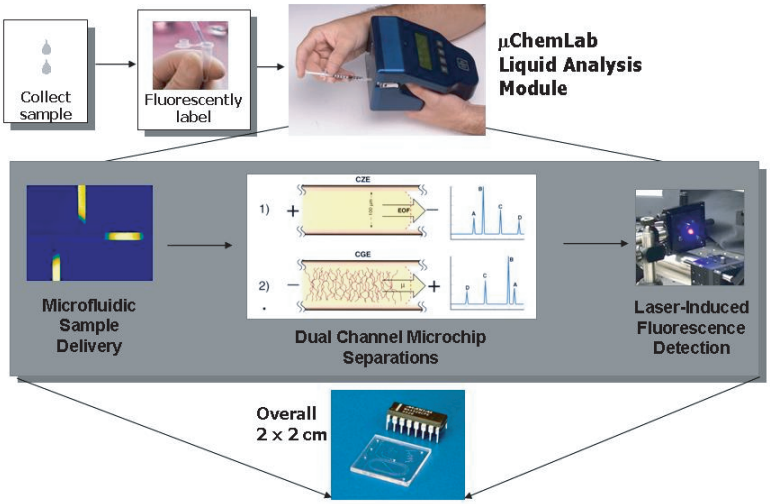


FIG. 1.1. *TheμChemLab^{TM} system which combines electrophoretic separations with sensitive laser induced fluorescence detection.*

2.1. Quantum Dot synthesis. The quantum dots used are colloidal nanocrystals prepared by adding a tri-butyl phosphine buffer containing $Cd(CH_3)_2$ and Se to a flask containing tri-octyl phosphine solvent (TOP) and tri-octyl phosphine oxide (TOPO) ligand. The mixture is heated to 360°C and the precipitate that forms is a QDot with a highly crystallized CdSe core and a protective ZnS shell that prevents oxidation and creates a quantum confinement effect. Since these QDots are not intrinsically water soluble, surface functionalization is required through the addition of hydrophilic ligands. Specifically, a silane shell can be created in which the thiol end binds to the QDot and the opposite polar end favorably interacts with water. This silanization process not only increases water solubility but the silane groups also serve as points of attachment for other functional groups (i.e. amines, carboxylic acids) that can create a charged surface. The QDots used in this study were negatively charged with phosphate groups.

2.2. QDot Bioconjugate. Rhodamine 6G (R6G) was the organic dye molecule the QDot was complexed with. The R6G was covalently bonded to an antigen and the antigen-R6G complex was attached to the QDot through a cross-linker. The QDot was added to the cross-linker in NaPi buffer (pH= 7.45) and allowed to react for one hour. A buffer exchange was then performed in which free cross-linker was separated from the QDot-cross-linker conjugate through a microfilter. The antigen-R6G complex was then added to the concentrated solution of 1 μM QDot-cross-linker and allowed to react overnight before doing any analysis (Figure 2.1). The protease used to cleave the antigen-R6G complex from the bioconjugate was an enzyme specific for the antigen.

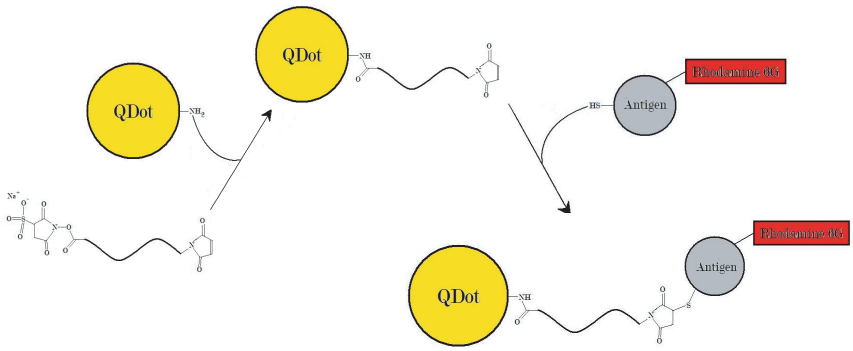


FIG. 2.1. *Formation of the QDot bioconjugate.*

2.3. Detection system. The samples were pressure injected onto a fused-silica microfluidic chip to perform capillary zone (CZE) and capillary gel electrophoresis (CGE) separation using laser-induced fluorescence (LIF) detection. LIF detection sensitivity of QDots was at nanomolar (10^{-9}) concentrations, but detection limits for the protease were not determined at the time of this writing. The free QDot molecules were initially tested in the μChemLab^{TM} to create a library of their migration times. There is a different fluorescence for each class of QDots depending on size of the particle; increasing size tends to redshift the emission spectrum. The four detectors used for the different QDots were the purple (∼400 nm), 488 nm, 532 nm, and the red (∼635 nm).

3. Results and Discussion. Individual sample analysis of the QDot and R6G was done using CGE because of the negatively charged nature of the molecules. Thus, movement was determined primarily by particle charge since the viscosity of the sodium dodecyl sulfate buffer minimized the effects of electro-osmotic flow:

Detector	Analyte	Migration Time (sec)
Purple/488 nm	QD490	210
Purple	QD465	204
488 nm	QD515	185
488 nm	Rhodamine 6G	191
532 nm	Y41	351
532 nm	QD600	192
532 nm	QD630	200
532 nm	G55	345
Red	675	blank result
Red	625	blank result

QD490 was used to make the QDot bioconjugate. Gel electrophoresis revealed that the negative charge of the QD490 was negated when complexed with the PSA-R6G. This problem was overcome by using CZE for sample testing so that the electro-osmotic flow would transport the bioconjugate through the channel in the microfluidic chip. To test if there was a shift in the emission profile, the sample was first tested on a fluorimeter and a shift in the emission spectrum was detectable, indicating that the bioconjugate was functioning. The sample was then diluted to 40 nM and tested

on the μChemLab^{TM} (see Figure 3.1). The sharp peak at 49 sec appears to be free

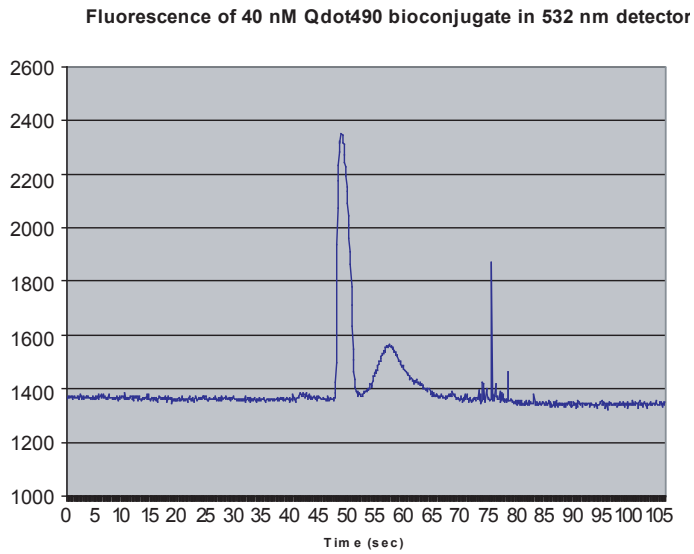


FIG. 3.1. *QDot bioconjugate migration in μChemLab^{TM}*

PSA-R6G dye that failed to complex while the small broad peak at 57 sec corresponds with the QDot bioconjugate. The sample was not tested with PSA enzyme at the time of this writing to see if there was a shift in the relative magnitudes of each peak that would indicate protease activity.

4. Conclusions. Research done up to this point indicates that the QDot bioconjugate works and that the emission shift can be detected on the μChemLab^{TM}. Additional research needs to be done with PSA enzyme to see if protease activity can be detected and what the sensitivity levels are on the μChemLab^{TM}. The indirect detection of such proteases is appealing because such information could lead to earlier cancer detection, especially given that certain proteases are expressed before tumors are visible. Other uses include assessing the stage of cancer development and determining the effectiveness of protease inhibitors as a class of cancer drugs. There is also the possibility of developing "multichannel" assays where multiple enzymes may be detected if there is minimal spectral emission overlap. Current migration times would tend to obscure individual behavior.

REFERENCES

[1] I.L. Medintz, H.T. Uyeda, E.R. Goldman, and H. Mattoussi, *Quantum dot bioconjugates for imaging, labelling and sensing.*, Nature Mater., 4 (2005), pp. 435–446.

[2] Ronald F. Renzi, James Stamps, Brent A. Horn, Scott Ferko, Victoria A. VanderNoot, Jay A.A. West, Robert Crocker, Boyd Wiedenman, Daniel Yee, and Julia A. Fruetel, *Hand-held microanalytical instrument for chip-based electrophoretic separations of proteins*, Analytical Chemistry, 77 (2004), pp. 435–441.

LOCK-IN AMPLIFIER FOR THE NIH BIODETECTOR

B. HINZE* AND J. BRENNAN†

Abstract. I have developed a small embedded lock-in amplifier for the NIH Biodetector. The lock-in controls the laser stimulus, processes the signal, and communicates digitally with the other electronics and computer. The heart of the lock-in is a 100 MHz TMS320F2808 DSP chip which interfaces with the main board using an I2C bus, and controls the 16 bit ADC and digital potentiometer through a SPI. The parameters of analog offset and gain are controlled digitally from the computer. Once the signal is converted from analog, all signal processing remains digital thus reducing analog circuit noise. Although the lock-in is not yet field tested, preliminary simulations indicate that overall processing delay is 0.1 seconds, and the signal to noise ratio is improved 300 times.

1. Introduction. The lock in amplifier that I am designing will work in the NIH Biodetector used to detect diseased molecules in a patients saliva. In the Biodetector fluorescent tagged antibodies mix with saliva and flow through a channel in a microchip under electrophoresis. The antibodies attach themselves to biomarkers present in the sample. These pairs separate from other molecules as they flow through the channel. A blue laser excites the pairs into producing a green fluorescence. A PMT receives this fluorescence and converts it to a voltage signal. The magnitude of this voltage ultimately determines the concentration of biomarkers in the sample, therefore indicating if the sample is diseased [2].

FIG. 1.1. *NIH Biodetector*

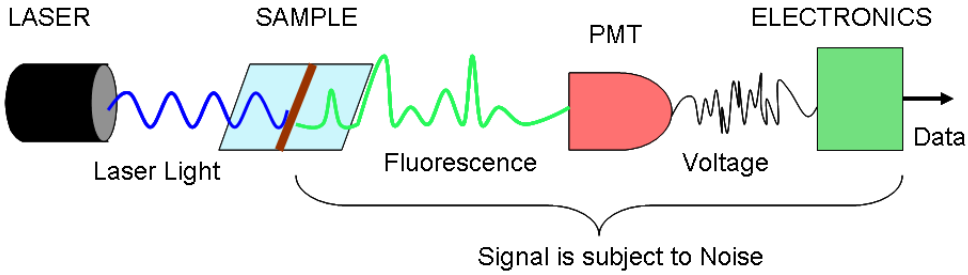


FIG. 1.2. *Fluorescence experimental apparatus*

The fluorescence from the sample can be very weak, with a great deal of optical and electrical noise present. This can make an accurate reading difficult. Using the

*San Jose State University
†Sandia National Labs

techniques of a lock-in amplifier the signal quality is greatly improved. A lock-in modulates the experimental stimulus at a given frequency, and then demodulates the response, thereby eliminating noise at other frequencies.

2. Noise and the Lock-In. Noise can be separated into two main categories. Deterministic noise is caused by known voltage signals and EM fields in the lab and typically has a discrete frequency spectrum. Random noise is caused by thermal and electronic randomness and is spread out through the frequency domain.

- Deterministic Noise: Effects of stray electromagnetic fields
  - Line noise from lights and outlets at 60Hz and harmonics
  - Capacitive coupling from switched mode power supplies, pulsing lasers, or other time varying voltage or EM field
  - Stray light incident on the PMT from laser or other source
- Random Noise: Inherent in natural and electrical systems [4]
  - $V_{noise(rms)} = \alpha/f \quad 1/f$ noise
  - $V_{noise(rms)} = \sqrt{4kTR\Delta f}$ Johnson noise
  - $I_{noise(rms)} = \sqrt{2Iq\Delta f}$ Shot noise

Deterministic noise can be reduced by careful design of the experimental equipment. The 60Hz line noise is reduced by isolating the power supplies and lights from the experiment. Separating signal and control lines and filtering stray light will also reduce this noise.

Random noise is more difficult to deal with. The greatest noise voltage is at low frequencies. As seen in Figure 2.1, (for the random noise in an op-amp) flicker noise is dominant at low freq and decreases with 1/f until the broadband white noise takes over. This can cause problems because the fluorescence signal changes very slowly,

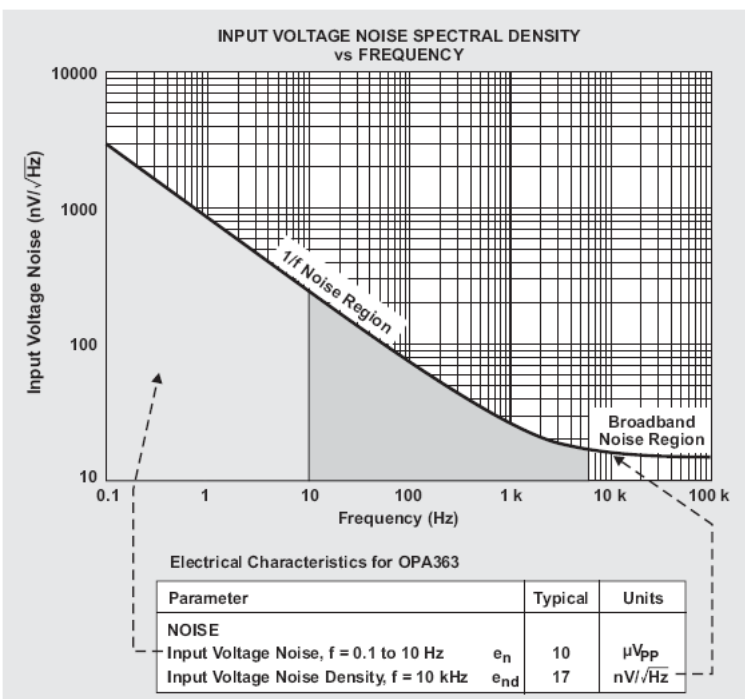


FIG. 2.1. *Random noise in an op-amp [1]*

having a frequency on the order of 1 Hz.

But, using the lock-in technique the signal is raised to a higher frequency. *A Lock-In modulates the experimental stimulus at a given frequency, and then demodulates the response, thereby eliminating noise at other frequencies.*

3. Lock-In Design Goals.

- Highest Signal/Noise possible
- Small as possible
- Precise digital control of analog gain and offset
- Digital data out through I2C bus
- Automatic laser modulation

4. Design Choices.

- Analog section uses precise, low noise components:
  - Low noise amplifier for analog gain: LT6234
  - Digital potentiometer to control gain and offset: AD5235
  - Anti-aliasing filter to condition signal for ADC: LTC1569
- Analog to digital converter is the 16 bit ADS8327 sampling at 10kHz with a range of 4.096 volts
- Microcontroller is the 100MHz, 32 bit TMS320F2808 DSP chip
  - Pulses laser at 220Hz
  - Communicates with ADC and digipot through SPI
  - Communicates to the main board through I2C
  - Digitally demodulates and filters response with 2000th order FIR filter
  - Fully programmable using C and Assembly language
  - Processor runs at 50Mhz and consumes around 400mA power
- Goal is to eventually fit the circuit on a 2X1.5 or smaller board

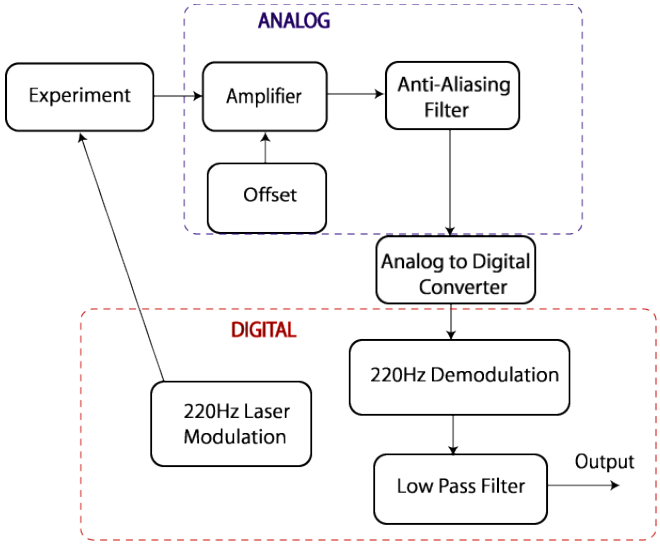


FIG. 4.1. *Lock-in block diagram*

5. Lock-In Simulations. My lock-in design began with a number of Matlab simulations [3]. These show how the lock-in works as well as the predicted improvements on the signal to noise ratio. Figure 5.1 and Figure 5.2 below show an ideal

signal, and the same signal as it appears lost in noise. In order to transform the

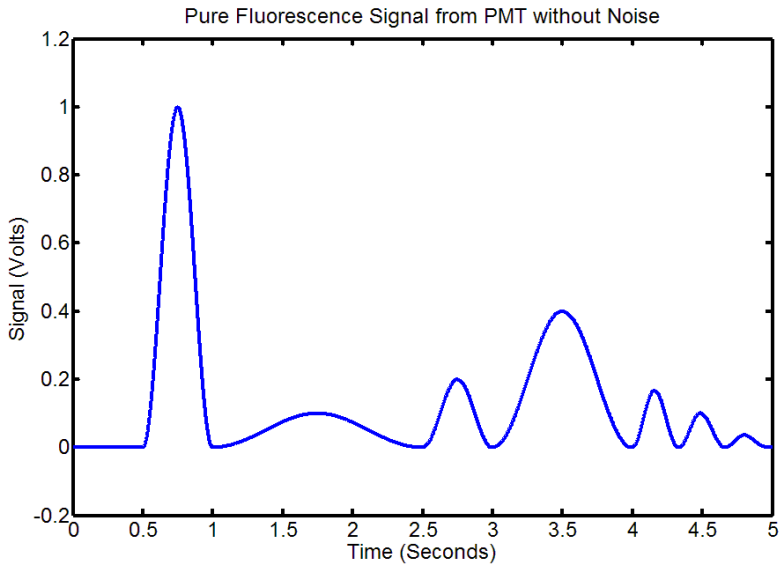


FIG. 5.1. *Ideal output signal*

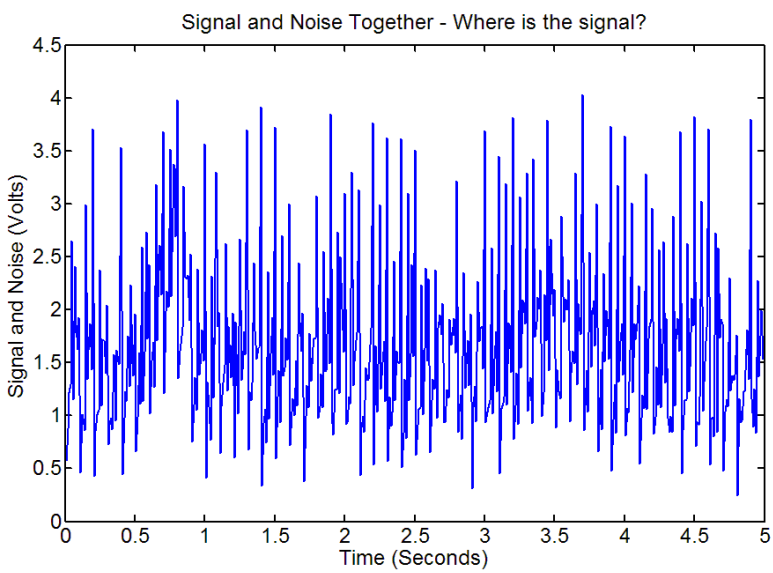


FIG. 5.2. *Signal with noise more than 4 times greater*

response out of the noise the microcontroller modulates the pulsing of the laser at 220Hz. Figure 5.3 below is a close-up view of the pure response signal after modulation. The signal travels through the analog gain and offset and anti-aliasing filter. The job of the analog components is to prepare the signal for the 16 bit ADC. The TI chip receives the digital signal from the ADC through an SPI. The TI chip then

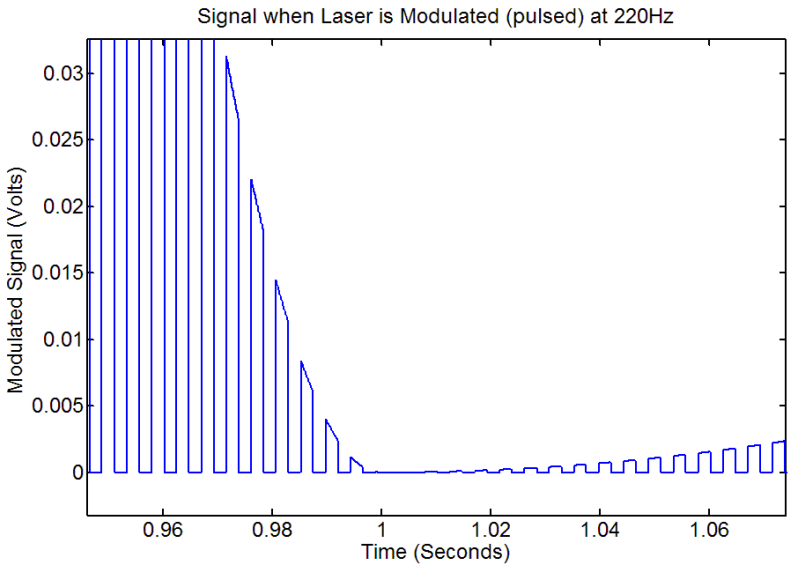


FIG. 5.3. *Close up view of modulation*

digitally demodulates the signal by multiplying it by a 220Hz sine wave. In accordance with equation (5.1) below, components of the signal around 220Hz are mapped to DC, and other frequency components are mapped elsewhere.

$$V_{sig}\cos(\omega_s)\,V_{dem}\cos(\omega_d) = V_{sig}V_{dem}\left[\cos(\omega_s+\omega_d)+\cos(\omega_s-\omega_d)\right] \tag{5.1}$$

After the demodulation a low pass filter cancels all higher frequencies to yield the

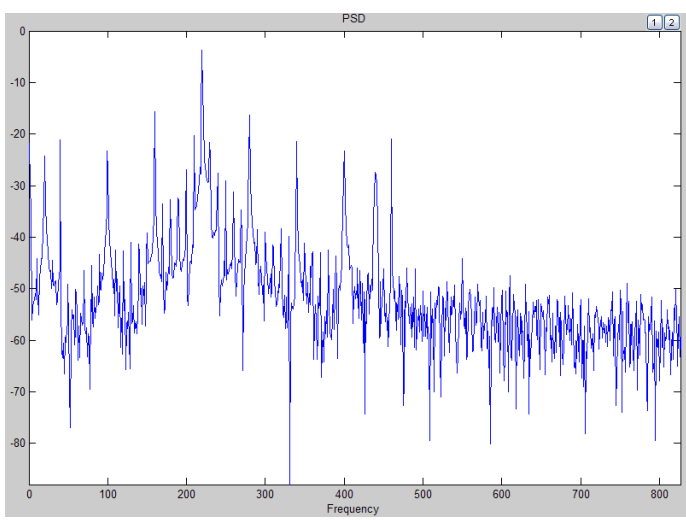


FIG. 5.4. *Frequency Domain plot of the demodulated signal showing the transformation of DC noise to 220Hz and the 220Hz signal back to DC.*

original signal. Im using a 2000th Order Finite Impulse Response Low Pass Filter.

It multiplies the last 2000 points of the input by coefficients, as shown in Figure 5.5 below, to pass only a narrow bandwidth around the signal.

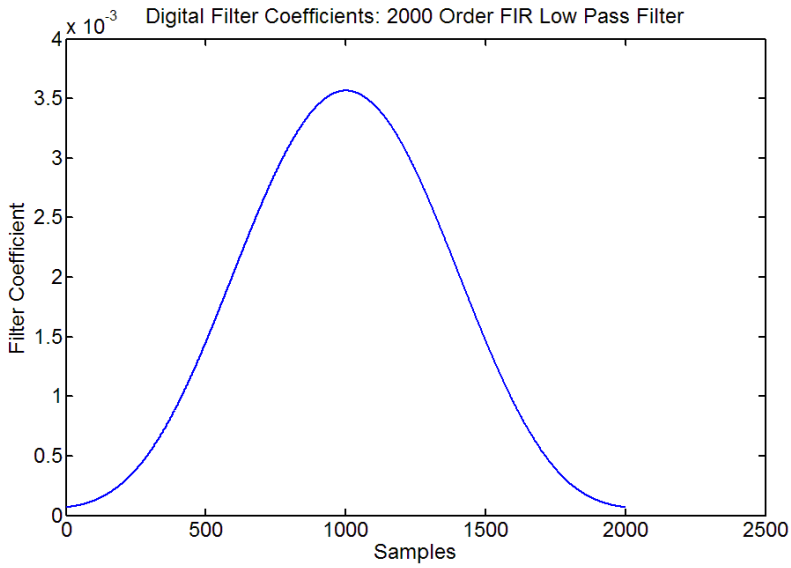


FIG. 5.5. *The 2000th order FIR DSP Filter.*

6. Conclusions. Shown in Figure 6.1 is the final output, filtered from noise. The signal to noise ratio is improved ∼300 times. Delay of signal reception due to

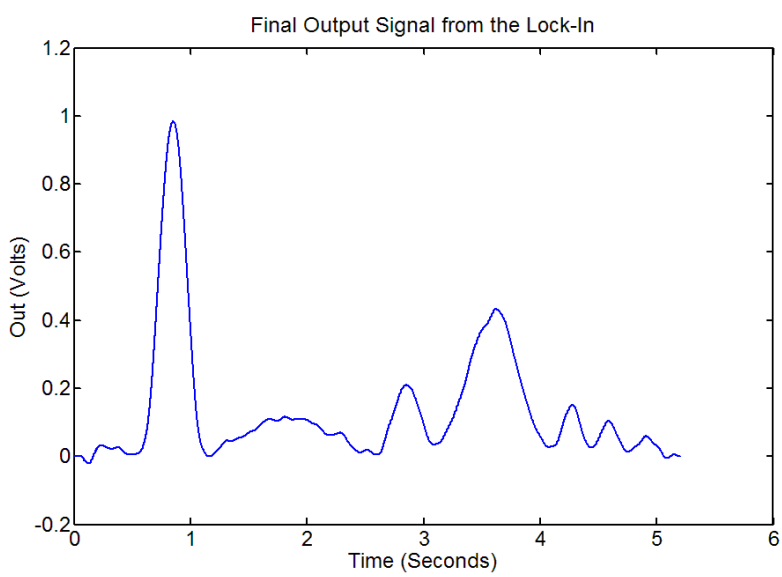


FIG. 6.1. *The final output; signal to noise is improved 300 times.*

the filtering is 0.1 seconds. Data points are produced 50 times a second.

Improvements: Lock-In performance can be improved by keeping noise around 220Hz to an absolute minimum. Therefore, the pulsing laser and control wires should be isolated from the output.

Trade-offs: Noise takes time to filter out. A more accurate reading will require more time. Also, if the experimental signal itself has a large bandwidth then it can be difficult to resolve with accuracy.

REFERENCES

[1] B.C. Baker, *Matching the noise performance of the operational amplifier to the adc*, tech. report, Texas Instruments, 2006.
[2] J. Brennan, 2006. private communications.
[3] The Mathworks, *Matlab r2006a with signal processing toolbox*, tech. report, 2006.
[4] Stanford Research Systems, *Srs 830 lock-in amplifier manual*, 2001.

DESIGN AND FABRICATION OF ROBUST GAS SENSORS USING METAL-INSULATOR-SEMICONDUCTOR DEVICES

STEVE MARSHALL*, WILL MEDLIN†, AND ROBERT BASTASZ‡

1. Overview. The ability to selectively detect gases in mixed environments using inexpensive tools is a highly sought after venture. Transformer headspaces, hydrogen fuel cells, and hydrogen plasmas are just a few environments where the detection of hydrogen and hydrogen containing compounds by a robust sensor would be very valuable. Metal-insulator-semiconductor (MIS) sensors represent a potential means to cheaply allow for the real-time detection of these gases. These devices were first discovered in the 1970s and have been the topic of ongoing research today.

MIS sensors utilized are thin film capacitors which consist of a layer of doped silicon and a layer of a catalytic gate metal separated by a thin (< 100nm) layer of silicon dioxide. For the work described here, n-type silicon is utilized; however, p-type silicon would work as well. When a gate metal is used that can dissociate hydrogen, such as Pd, Pt, or Ir, hydrogen atoms adsorbed on the surface can diffuse to the metal-SiO_2 interface and cause a shift in the capacitance-voltage (CV) curve for the device. Responses are commonly characterized by measuring the change in voltage required to maintain a constant capacitance at the inflection point of the CV curve. Previous work has shown that the change in the CV curve is related to the hydrogen partial pressure for many decades of hydrogen pressure [3]. Due to the high sensitivity of these devices, large surface areas are not required, which makes the use of MIS sensors amenable to gas sensing within small spaces in machinery such as transformers and reactors.

Despite the sensitivity of these devices for hydrogen, numerous drawbacks have prevented their widespread use. A major drawback is in the nature of the catalytic gate metal. Metals such as Pd and Pt excel at dissociating hydrogen; however, they also excel at catalyzing less desirable reactions. Species such as oxygen, CO, and unsaturated hydrocarbons can react with hydrogen or directly react with the surface to change the response. In some cases, this action allows for detection of gases other than hydrogen; however, it adds complexity to interpreting sensor response and can mask the presence of hydrogen. Many solutions to this problem have been proposed including the use of thin oxide coatings [1] and polymers [6] to filter out the effect of contaminant gases. Current work focuses on the use of alkane thiol self assembled monolayer (SAM) coatings and bimetallic gate metals to tailor surface chemistry and control these effects.

2. Sensor Fabrication. Sensors are prepared by evaporating material via a dual electron beam evaporator (Angstrom Sciences, EBES-67369) at Sandia. In this system, metals are evaporated using a high power electron beam. The beam is produced using a 10kV filament with a typical power output around 300W and positioned onto metal inside a crucible using magnetic and electric fields. The use of these fields allows the beam to be bent out of the trajectory of ionized and accelerated atoms, preventing sputtering and contamination of the sample. Electrons striking the metal impart energy and cause the metal to melt and evaporate. Close monitoring of the

*University of Colorado at Boulder
†University of Colorado at Boulder
‡Sandia National Laboratories

filament current allows for control of the deposition rate with precisions of approximately 0.5 A/s. The system at Sandia contains two electron beams to allow for the simultaneous depositions of two metals to form disperse alloys. Controlling the deposition rate of one metal in relation to the other allows for the creation of precise weight percentages.

The deposition rates and thicknesses are measured using a system of three quartz crystal microbalances (QCM, Inficon XTC/2): one for each of the individual metal contributions and one for the substrate. The QCM works by measuring changes in the resonant frequency of a quartz crystal. In the simplest case, the change in resonant frequency of the quartz crystal on addition of metal to the crystal surface is linearly related to the mass of material added. This relation allows for the simple calculation of the thickness of the layer (assuming uniform deposition) and the rate of deposition. Both of these operations are performed automatically within the Inficon device.

Substrates consisting of a 4" diameter n-type (10 ohm-cm) Si wafer with 50nm SiO2 on top and a 200 nm ohmic contact (Al) on back are utilized for the initial deposition and then diced to 1/8" x 1/8" squares to make individual sensors. Metals deposited on these substrates include Pd, Pd/Ni, Pd/Cu, Pd/Ag, and Pd/Au. Contacts on the deposited metal side are formed using conductive epoxy. Sensors prepared at Sandia will be tested with a custom flow cell device and a variety of surface techniques at the University of Colorado such as temperature programmed desorption (TPD), Auger electron spectroscopy (AES), and atomic force microscopy (AFM).

3. Selectivity Modification Through Self Assembled Monolayer Coatings. Recent work at the University of Colorado has shown that C18SH alkane thiol self assembled monolayers (SAMs) deposited on the surface of a Pd-MIS sensor cause a signification increase in the sensitivity to acetylene. Whereas uncoated sensors showed no observable response to acetylene at 50 C, thiol coated sensors showed a significant, 60mV response. To test whether the effect of the SAM was due to Pd-S bonds or the hydrocarbon tail of the alkane thiol, sensors were also prepared with sulfur on the surface by dissociative adsorption of hydrogen sulfide. This sulfur coated sensor also showed an increased response to acetylene; however, it was not as dramatic as the thiol coated sensor. An additional test was performed with a sensor saturated with CO to determine if the primary mechanism for the increased acetylene hydrogenation was site blocking. The lack of response to acetylene with the CO saturated sensor indicates that sulfur and thiol coatings give heightened acetylene response through a more complex mechanism. This work highlighted the unique benefits of the SAM coating; however, there was a significant drawback.

When exposed to oxygen, both the thiol coated and sulfur coated sensors showed a significant degradation in performance. Whereas the thiol coated sensor before oxidation may be able to detect acetylene concentrations lower than 40ppm, the oxidized detector cannot do so with such precision. XPS studies suggest that this degradation is the result of sulfur on the metal surface oxidizing to sulfonate or sulfate in air [7]. Preventing this oxidation would create a highly sensitive acetylene detector.

4. Bimetallic Gate Metals for Surface Modification. Current work focuses on the use of bimetallic alloys in conjunction with alkane thiol self assembled monolayers to design more selective and robust sensors for hydrogen and acetylene detection. Sensors containing alloys of palladium with nickel, silver, gold, and copper are being prepared with varying weight percentages of palladium. These sensors will

be tested with and without SAM coatings to determine their response to gases such as hydrogen, acetylene, oxygen, hydrogen sulfide, carbon monoxide, and ethylene.

Significant work has been performed in the use of bimetallic alloys to catalyze reactions with greater rates and selectivity than their monometallic counterparts. We drew upon this body of literature to select nickel, silver, gold, and copper to alloy with palladium to create a more effective gate metal.

Density functional theory (DFT) studies have suggested that nickel may be more reactive to acetylene than palladium by more strongly binding hydrocarbons [8]. Recent studies have suggested via a density of states analysis that palladium-nickel alloys may lower the acetylene hydrogenation barrier and raise the ethylene hydrogenation barrier, improving selectivity [2]. Although this work modeled the metal surface as a 2x2 unit cell three layers deep, which will not account for any long range or relaxation effects, palladium-nickel is still an interesting candidate for a gate metal. Silver was selected for its common use as a promoter in industrial acetylene hydrogenation catalysts [4]. In addition, SAMs deposited on silver has been shown to have increased resistance to oxidation [7]. Palladium-gold bimetallic alloys have been shown to increase catalytic activity for reactions involving sulfur [9]. Finally, previous work has shown that palladium-copper alloys can resist restructuring effects from sulfide exposure [5] which may enable higher coverages and dosages of sulfides to tailor response. In addition, current work at the University of Colorado in thin film membranes and previous work with SAMs [7] suggests that sulfur binds tightly to copper and resists oxidation, which may lead to a more robust sensor.

5. Conclusions. MIS sensors represent a viable means of achieving gas sensing in mixed environments. These devices are inexpensive and easy to fabricate. Nevertheless, difficulties in tailoring selectivity and elucidating the sensing mechanism have limited their implementation. Through the work explained above, we hope to create a well-characterized, highly selective, and highly sensitive gas sensor for use in many environments. Sensors prepared at Sandia will be tested with a variety of techniques at the University of Colorado to gain greater understanding of the sensing mechanism and effect of these alloys.

REFERENCES

[1] A.E. Abom, E. Comini, G. Sberveglieri, L. Hultman, and M. Eriksson, *Thin oxide films as surface modifiers of mis field effect gas sensors*, Sensors and Actuators B, 85 (2002), p. 109.

[2] A.M. Goda, M.A. Barteau, and J.G. Chen, *Correlating electronic properties of bimetallic surfaces with reaction pathways of c2 hydrocarbons*, Journal of Physical Chemistry B, 110 (2006), p. 11823.

[3] M. Johansson, I. Lundstrom, and L.G. Ekedahl, *Bridging the pressure gap for palladium metal-insulator-semiconductor hydrogen sensors in oxygen containing environments*, Journal of Applied Physics, 84 (1998), p. 44.

[4] N.A. Khan, S. Shikhutdinov, and H.J. Freund, *Acetylene and ethylene hydrogenation on alumina supported pd-ag model catalysts*, Catalysis Letters, 108 (2006), p. 159.

[5] A. Kulprathipanja, G.O. Alptekin, J.L. Falconer, and J.D. Way, *Pd and pd-cu membranes: Inhibition of h2 permeation by h2s*, Journal of Membrane Science, 254 (2005), p. 49.

[6] D. Li, A.H. McDaniel, R. Bastasz, and J.W. Medlin, *Effects of a polyimide coating on the hydrogen selectivity of mis sensors*, Sensors and Actuators B, 115 (2006), p. 86.

[7] J.C. Love, D.B. Wolfe, R. Haasch, M.L. Chabinyc, K.E. Paul, and G.M. Whitesides, *Formation of structure of self-assembled monolayers of alkanethiolates on palladium*, Journal of the American Chemical Society, 125 (2003), p. 2597.

[8] J.W. Medlin and M.D. Allendorf, *Theoretical study of the adsorption of acetylene on the (111) surfaces of pd, pt, ni, and rh*, Journal of Physical Chemistry B, 107 (2003), p. 217.

[9] A.M. Venezia, V. La Parola, G. Deganello, B. Pawelec, and J.L.G. Fierro, *Synergetic effect of gold in au//pd catalysts during hydrodesulfurization reactions of model compounds*, Journal of Catalysis, 215 (2003), p. 317.

VISUAL SIMULATION OF BIOLOGICAL PATHWAYS

MATTHEW MEHNE* AND E.E. MAY†

Abstract. An important stratum among the many layers that make up the inner-workings of living cells and organisms is that of the metabolic network, a complex array of chemical reactions controlled by genes and the presence or absence of different enzymes, cofactors, and other regulatory components. It is common for biologists to visualize metabolic reaction networks graphically, representing them with metabolic maps-collections of flow charts showing interconnecting reactions and pathways. By simulating the dynamics of the metabolic pathways, we can add another dimension to the standard metabolic map by visually representing time dependant properties such as metabolite concentration, activation of genetic regulatory controls, and reaction rate. We model these dynamics using the Xyce electronic circuit simulator, equating the flow of metabolites in a biological system to the flow of charge in an electrical circuit. In order to illustrate the circuit model in a biological context, we have developed a visual interface to display the simulation data as an animated biological schematic. Our JAVA-based visualization tool draws a dynamic metabolic map and correlating gene network based on Xyce simulation data. The interface is designed to provide access to relevant and useful analysis tools that help integrate the simulation data quickly and efficiently.

1. Introduction. Biological systems are extremely complex and they can be described and analyzed at many different levels of abstraction. The inside of a living cell can be compared to a massive chemical factory where products are continuously imported and exported, and chemical procedures are managed rigorously by genes and their constituents. Having the right chemicals available at the right time is vital to the survival and function of all living cells. When we study the flow of chemicals that occurs from one reaction to the next, we find that more than a steady assembly line of processes, what we have is an intricately dynamic network of reaction pathways, where, based on the state of the cell, different processes are prioritized and at times they are shut on or off. We find that chemical functions shift based on metabolite availability and cellular supply and demand.

In order to begin to conceptually grasp the complex dynamics that manipulate the seemingly countless processes that occur in the cell, we must abstract their overall function from the intricacies of the chemical reactions that drive them. One simplification is to think of the flow of chemicals as running through a giant network of interconnected pipes of different shapes and sizes, regulated by a variety of valves, pumps, and other control devices. Continuing with this analogy, we can illustrate researchers in bioinformatics as sort of forensic plumbers; the research involves reconstructing, from laboratory evidence, how the networks are put together. More and more data is becoming available, including values on specific reaction rates as well as properties of different gene interactions in biological systems. Some networks have been studied so well that there is enough data for one to attempt to actually recreate the chemical flow and regulation that occur in those networks with a representative model made up of water pipes, pumps, and valves. The motivation to consider attempting such a feat would probably not be too different from what motivated Watson and Crick to experiment with models made from metal rods and balls to complement incomplete experimental data, a technique that lead them to discover the structure of DNA.

Similar to pipes and valves, but at a smaller and much more practical scale, we model the flow and control of biological networks as electric currents running

*University of Southern California.
†Sandia National Laboratories

through electronic circuits. This model provides a very useful parallelism that we use to recreate the dynamics of biological pathways. Using the Xyce electronic circuit simulator to replicate the dynamics of the reaction pathways, we are able to provide a useful glimpse in real time of how metabolic networks interact with constituent genes and the environment. It could be said that one disadvantage of using electronic circuits instead of plumbing components in our model is that we cannot actually see the flow of material or the opening and closing of valves as it occurs, instead we have immense tables of simulation data listing changes in voltage and current over time. To solve this, we develop a method to visually recreate the flow and dynamics of our models, using the Xyce output to animate the simulated network behavior over time. This provides a way, in a sense, of experiencing the simulated network–to get a feel for the dynamics and timing of the system. The hope is that this will help us gain a deeper understanding of why and how biological pathways function the way they do.

2. Goals. The primary goal of this project has been to facilitate the data analysis of biological pathway simulations run on the Xyce circuit simulator by implementing an interface that integrates the data into a descriptive visual representation, not only to further BioXyce research, but also to provide the data in a format that is accessible and useful to anyone studying biological pathways. Before this project, most analysis of the Xyce simulation data involved drawing time plots of the data to examine the concentration values of the simulated metabolites in the network. Plotted BioXyce data is often obscure to anyone without a thorough understanding of the network being represented. Even though an engineer with a solid background in electronic circuits could reconstruct the connectivity of the circuit design prescribed in the netlist file, knowledge of bioinformatics would still be vital to understanding the circuit's significance in the originally intended biological context. The layout and dependencies of the underlying biological pathway must be understood in order for the simulation data to be relevant to anyone other than the designer of the circuit. Therefore, we have found it useful when analyzing circuit simulations to have a method of portraying the corresponding biological pathway.

Part of the basis of our visualization approach was to adhere as much as possible to current trends in biological network drawings. Bioinformatics is still a relatively new field, and there is still no set standard on how to draw and lay out metabolic pathways and gene networks; however, there are some common patterns that can be followed to make the visualization more recognizable to the general academic community. Our goal has been to integrate the simulation data without deviating too much from the static layouts biologists use and are familiar with. We also aimed to keep the animation aspect of the visualization as simple and intuitive as possible so that even the untrained eye can get a fairly accurate feel for the concentration flow that is being represented.

3. Methods. In order to meet the visualization needs for the BioXyce project, a tool was developed in Java that automates the data representation starting from raw Xyce data, and then draws out the data in a biological context. The program is called XMAP[1], and being based in Java, it is supported by multiple platforms. It has been successfully run on both Windows and Macintosh machines.

XMAP was designed to try to make BioXyce simulation data as accessible as

[1]Xyce Metabolic Analysis Program; the X also represents the crossing between the electrical and biological circuit domains, while MAP is a reference both the visual representation of the biological maps as well as the process of mapping simulation data onto biological network data.

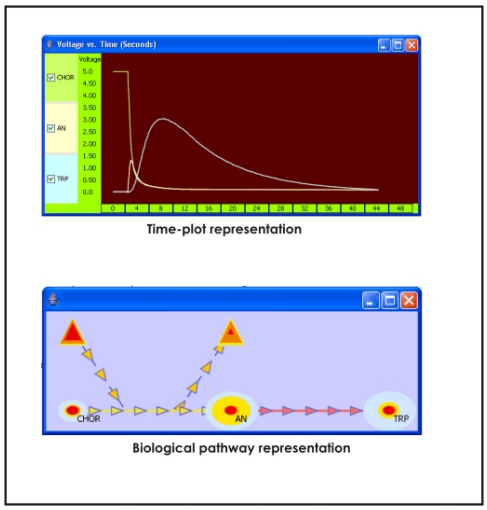


FIG. 3.1. *Time-plot and pathway visualization in XMAP*

possible. The most basic way to view simulation results in XMAP is through the automated graph plotter, which can instantly draw time plots for data points in the simulation output file (see Figure 3.1). Beyond drawing time plots, XMAP's main feature lies in its tools for drawing pathways that can then be automatically animated by XMAP based on loaded simulation data. Xyce runs simulations based on given time intervals; Xmap visually recreates the changes that occur during these time intervals either in real time or at a faster/slower pace as specified by the user. The hope is that dynamic aspects of the simulated networks can be assimilated more effectively by viewing the biological events as they occur in real time.

In order to represent the simulation data in a pathway context, we design animated network nodes that visually represent changes in concentration. We have found that changes in both size and color provide good visual queues to represent concentration changes. Growing and shrinking the metabolite nodes based on their concentration changes works best for mentally gauging the rate at which an elements concentration is increasing or decreasing. The average viewer can instantaneously recognize whether an animated object is growing or shrinking. Using color intensity to represent values has several advantages. In a static context, we seem to perceive colors faster than sizes; and visually, color changes tend to have a more intense psychological effect. When viewing pathways in XMAP with dozens of nodes, the color contrast becomes very useful, since it allows the viewer to asses the general state of the whole network at a glance. Color changes can be noted peripherally in a large network with much more ease than size changes, and patterns can be noticed by observing the overall color localized around different areas of the map.

The animated nodes in XMAP allow very small changes to be perceived since they have four layers of different colors that grow and shrink independently based on their concentration values. By identifying which sphere is growing, one can be completely certain of what intensity is being represented within 25%. An accustomed user can fairly easily refine their perception to the point where they can recognize a value within 10% or less. It should be mentioned that placing the mouse over any animated element allows the user to view the exact concentration value at any given

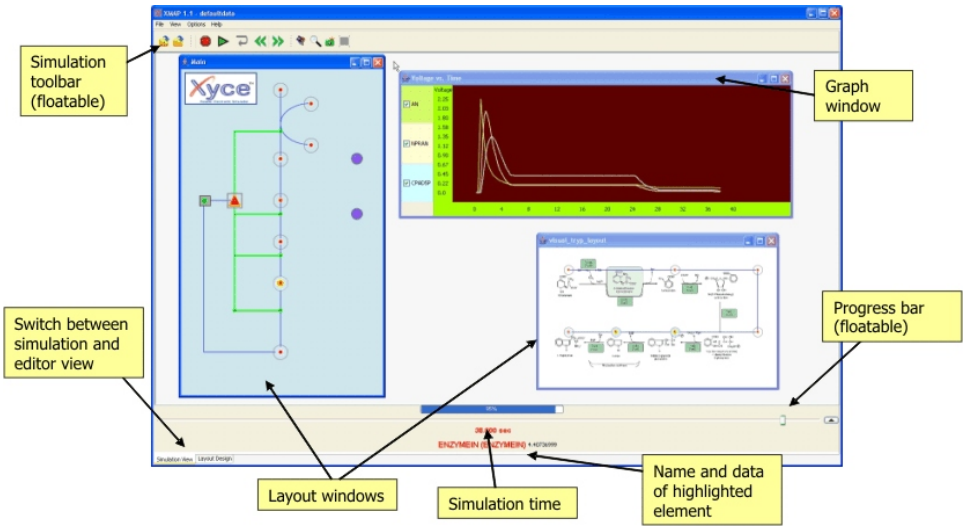


FIG. 3.2. *Simulation view in XMAP*

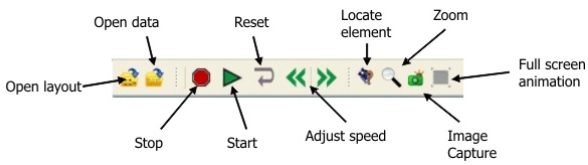


FIG. 3.3. *XMAP simulation mode toolbar*

time in the simulation.

XMAP has two views or user modes. The simulation view is where simulation data is analyzed; it is the context in which the animated visualization and time-plots are displayed. The layout design view is where biological layouts can be drawn and edited. Users can switch views by clicking on the tabs near the bottom of the XMAP window. The interface was designed to allow switching between the two views quickly and easily-which is useful, because often while running an animation based on a given biological layout, the user will want to change the layout to focus on different aspects or behaviors of the simulated data. For example, when observing a network with dozens of elements, one might need to focus momentarily on just two elements of interest. The user can proceed, while the simulation is running, to edit the layout as needed, perhaps erasing metabolites that don't need to be viewed at that moment, or dragging metabolites of interest so they are adjacent to one another. As soon as the layout is saved, changes are instantly reflected in the simulation view and are animated accordingly.

As shown in Figure 3.2, the simulation view allows multiple internal windows with different visual representation of the same simulation. Each window updates itself based on the current simulation time and loaded simulation data. The simulation time can be changed by dragging the progress bar, clicking on the plot, or by clicking the run button, which is on the toolbar represented by a green arrow (Figure 3.3).

The visualization can be run at different speeds and frame rates. If a speed or frame rate is selected that is not compatible with the simulation data (e.g., if a

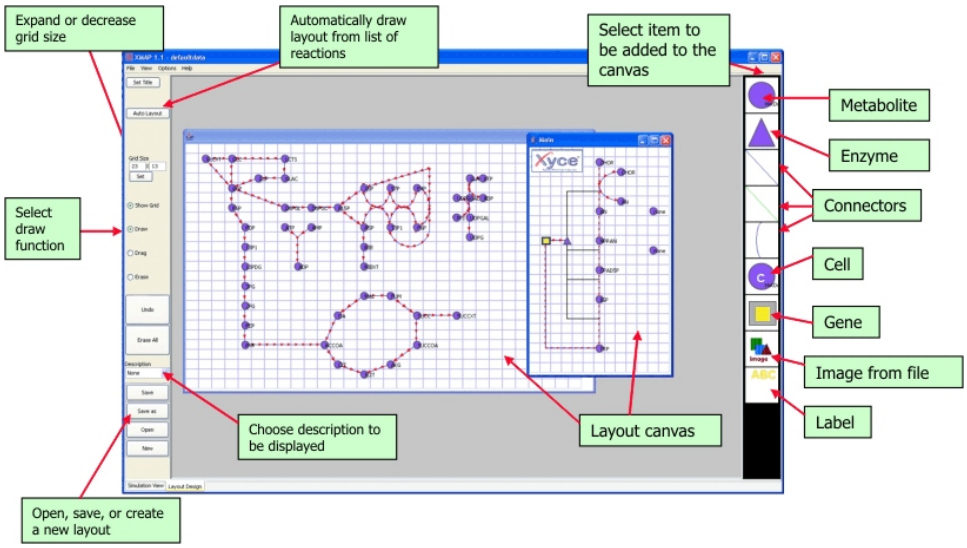


FIG. 3.4. *The layout design view in XMAP*

simulation is being run that only provides ten data points for every simulated second, and the frame rate is set to 24 frames per second) XMAP automatically interpolates the data. Also, for very large simulations (some data files can reach the gigabyte range), XMAP allows the user to skip lines of data or load subsets of the simulation for analysis.

Figure **??** displays a screenshot of XMAP's layout design view. The right panel contains the palette of objects that can be drawn on the layout canvas.After an object is drawn, it must be assigned a datapoint in the Xyce simulation data for it to be animated in simulation view.

Although some features were implemented to help automate the layout drawing process, the current version of XMAP relies mostly on the user to draw and edit the schematic representations of the simulated networks to be visualized. The most prominent advantage behind this is that the visual network layout can be customized to suit the user's preferences and analysis needs. As stated previously, there is no recognized standard in biological research prescribing how to draw metabolic networks, though users may want to draw the network in a way consistent with other schematics portrayed in academic literature. Manually drawing the network takes a small amount of time compared to the work involved in setting up a Xyce simulation. Once a network layout has been drawn, it can be saved and used with other simulation data sets. Our hope is to eventually establish a library of layouts that can be reused, edited, and shared among XMAP users.

As explained previously, XMAP is very good about allowing layout changes to be made on the spot. This is especially useful at moments when seeing the layout animated by simulation data provides an insight which inspires improvements to the layout design. Much more frequently the reverse will happen: seeing the visualized simulation data calls for a change in the Xyce simulation. What would be desirable is if the user could use the XMAP interface to make changes in the Xyce netlist file and then seamlessly rerun the simulation and load the data into XMAP. This has yet to be fully implemented, but it is now possible to load a simulation in XMAP,

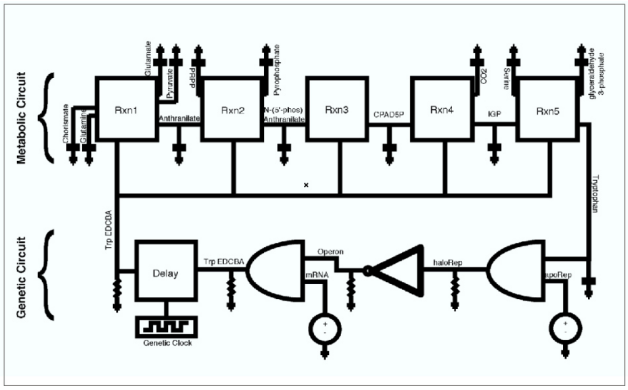


Fig. 4.1. *Tryptophan circuit representation*

and from the drawn layout, change the initial concentration values of metabolites and rerun the Xyce simulation from within XMAP. The user can Ctrl-Shift click on a metabolite to bring up a dialog with the current initial concentration value. The user can then set a new value; when the value is submitted, the Xyce circuit simulator is automatically initiated and run in the background as an external process. When the Xyce simulation finishes, XMAP reloads the data file and the visualization windows are refreshed to represent the new data.

4. Results. We ran various BioXyce simulations that were then visualized using XMAP. Each simulation focused on different aspects of XMAP's features and capabilities. Our simulation of tryptophan biosynthesis in the Escherichia coli bacteria provided an ideal model of a small hybrid metabolic/gene network. Another simulation, developed by Richard Schiek, of multicellular differentiation in drosophila embryos, was the initial basis for the implementation of multicellular visualization capabilities in XMAP.

4.1. Tryptophan Biosynthesis. The tryptophan biosynthesis pathway was simulated as a series of five primary reactions, starting with chorismate and the last one producing tryptophan (see Figure 4.1). Each reaction is ultimately regulated by a gene that is activated in the absence of tryptophan. The pathway is an example of a genetic feedback loop, where the absence of tryptophan promotes the production of more tryptophan and vice versa. The pathway functions in E. coli as a way of maintaining proper tryptophan levels in the cell.

Figure 4.2 shows the simulation in its initial state. Chorismate is at its maximum concentration while the rest of the metabolites are at their minimum. None of the main reactions are occurring since the genes regulating the pathways begin in a deactivated state and they experience an activation delay.

At just under three seconds (Figure 4.3) the genes have been activated and the flow of the reactions has begun. Note that more intense arrow colors represent faster reaction rates. The reaction flow has yet to reach the last two metabolites: indole-3-glycerol-phosphate and tryptophan.

Four seconds later (Figure 4.4), it can be observed that the reaction flow has shifted down the pathway. At this point chorismate is nearly depleted, and tryptophan is at its highest concentration and continues to degrade (degradation is symbolically labeled on the layout as consumption by other pathways).

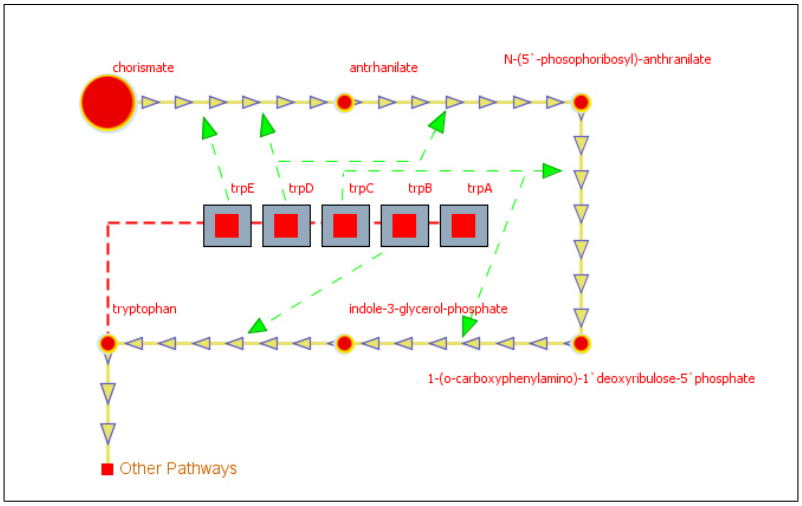


FIG. 4.2. *Tryptophan biosynthesis simulation at $T = 0$ seconds, visualized in XMAP*

Finally at the 14 second mark (Figure 4.5), the inhibitory influence of tryptophan on the genes has propagated enough to deactivate them. All of the reactions have stopped since the genes have stopped activating the production of the necessary enzymes to catalyze the reactions. The only flow left is the degradation of tryptophan.

The graph in Figure 4.6 shows chorismate sharply dropping while tryptophan increases slowly. After seeing the simulation data represented in the biological context, we understand this occurs because the metabolites between chorismate and tryptophan act like a buffer between the two metabolites.

4.2. Drosophila Multicellular Differentiation. One of the later capabilities added to XMAP was the ability to represent multi-cell systems. One of the primary goals driving the BioXyce project is to continue simulating larger and more complex networks, including entire cells. When simulating a multicellular scenario, there exist two network levels: the lower level is made up of the metabolic and gene pathways within the cells, and on top of that we have a network of multiple cells interacting with each other. In these scenarios, both internal and external processes affect the system as whole.

The first multiple cell simulation using BioXyce modeled multicellular differentiation occuring in drosophila embryos. Since we are simulating a 10 x 10 grid of cells, we would need to display 100 layouts to represent the inner network of each cell. Instead, XMAP displays a grid containing all the cells. The inner network of any particular cell can be displayed on demand by Ctrl-clicking a cell on the grid. A new internal window pops up showing the inner network. To help keep track of which inner network corresponds to what cell, XMAP provides the option of drawing anchor lines that map each cell to its inner layout (if open).

As seen in Figure 4.7, the inner networks of two drosophila cells on the grid are being displayed in new windows. The windows may be rearranged as needed. They remain easy to keep track of since they are visually anchored to the cell they represent. XMAP also has the ability to automatically generate custom cell grids such as the one in the figure.

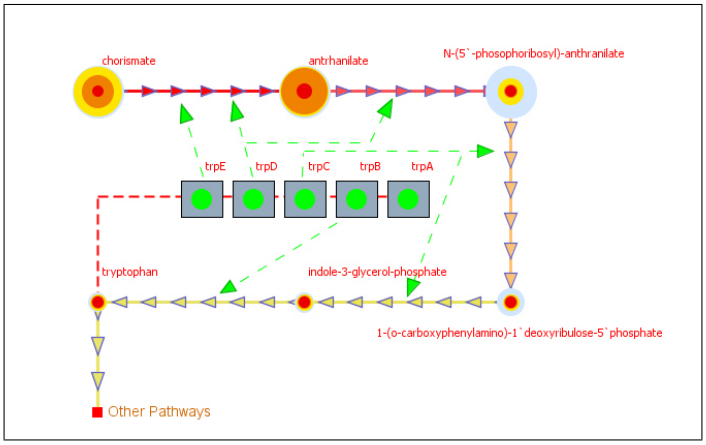


FIG. 4.3. *T = 2.75 seconds*

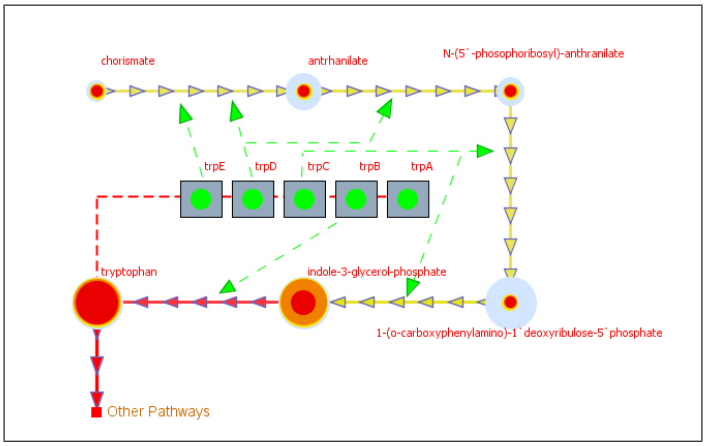


FIG. 4.4. *T = 7.75*

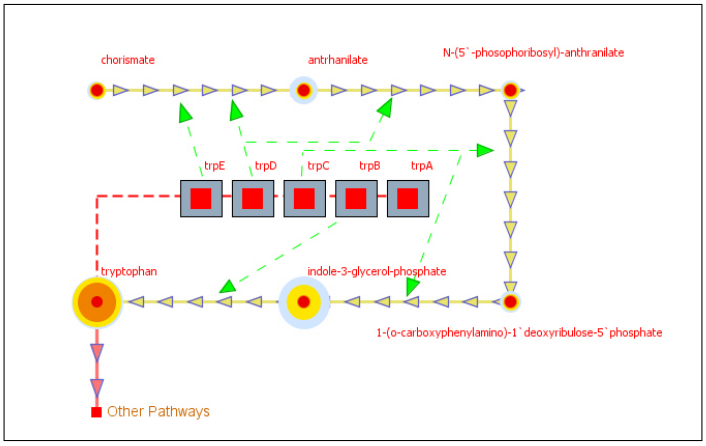


FIG. 4.5. *T = 14 seconds*

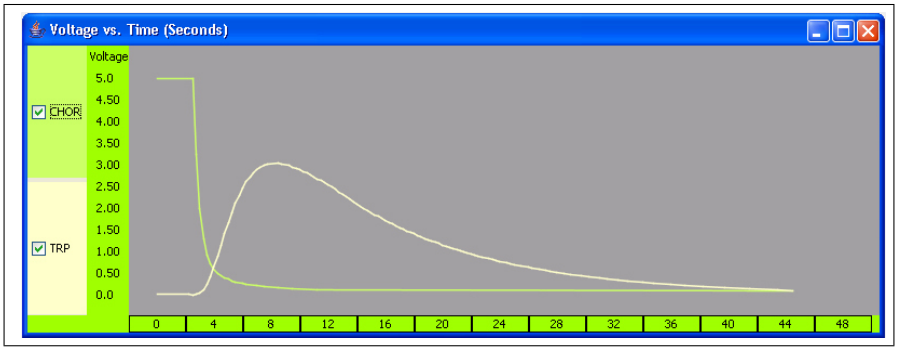


FIG. 4.6. *Chorismate and tryptophan concentration values compared against one another. The graph lists 'Voltage' as the Y domain because in Xyce, voltage values across capacitors represent metabolite concentrations.*

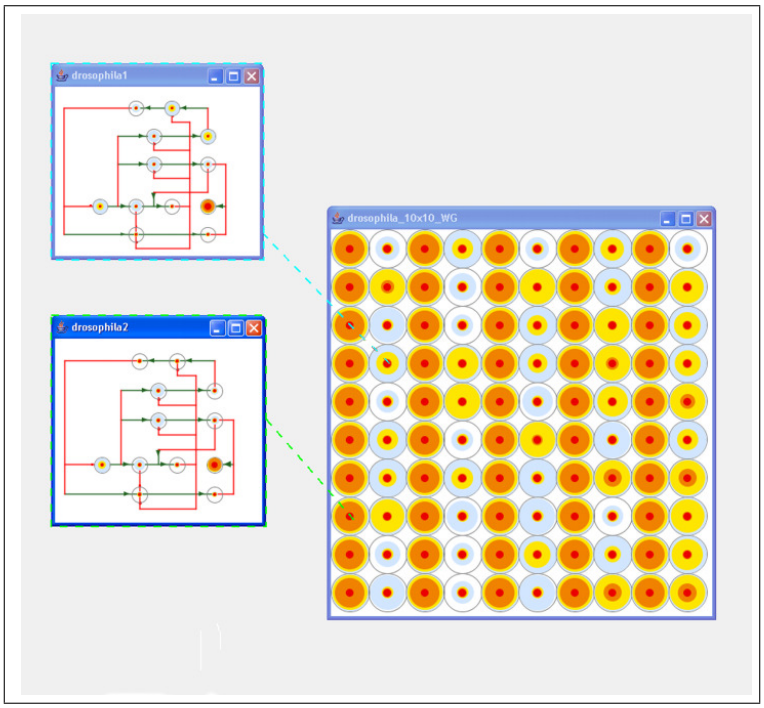


FIG. 4.7. *Multicellular visualization in XMAP of differentiation in embryonic drosophila cells*

5. Conclusions. Currently, XMAP provides a functional and versatile visualization interface for Xyce biological pathway simulations. The hope is that it will aid future BioXyce users by providing a quick and effective way to examine simulation output. It is also hoped that it will provide an effective means of presenting simulation results in an understandable manner to general academic audiences.

Ideally, in the future, XMAP can be fully expanded to function as a complete Xyce interface for simulating biological pathways. The idea is that, potentially, a person with no background at all in circuit simulation could still make use of Xyce by having XMAP automatically generate Xyce net-list files and run the simulation based on a schematic drawn in XMAP that has been annotated with reaction rate

information and metabolite concentrations. With this feature, any biologist wanting to visualize the dynamics of a pathway, or an entire network, could do so at the click of a button. Researchers in biology would be able to take advantage of the powerful capabilities that Xyce has to offer without having to know the difference between a capacitor and a resistor. The framework has been set for this, and it is a very feasible goal for the future of XMAP.

REFERENCES

[1] Richard L. Schiek and Elebeoba E. May., Xyce Parallel Electronic Simulator, Biological Pathway Modeling and Simulation. SAND REPORT, March, 2006.

[2] Xyce: Parallel Electronic Simulator Reference Guide. http://www.cs.sandia.gov/xyce, 2003.

[3] Purvi Saraiya, Chris North, and Karen Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. Information Visualization, 2005.

SUB-STRUCTURED BIOMOLECULAR DYNAMICS SIMULATIONS USING MULTIBODY DYNAMICS ALGORITHMS THROUGH LAMMPS POEMS COUPLING

RUDRANARAYAN M. MUKHERJEE† AND PAUL CROZIER‡

Abstract. In this paper we outline a method of reduced order modelling of biomolecular systems as sub-structured multi-rigid body articulated systems and the integration of a molecular dynamics software with a multibody dynamics software to facilitate this modelling effort. We use a recursive $O(n)$ method based on Kane's method for generating and solving the equations of motion. The methodology is verified by simulating several biomolecular systems. The method shows good energy conservation in NVE ensembles and preserves the essential dynamics of the system. We have developed an open-source computational tool by combining a classical molecular dynamics software LAMMPS and a multibody dynamics research code called POEMS. This tool is freely available to all researchers and gains on the complementary nature of the two codes by coupling the efficient force calculation algorithms in LAMMPS with the efficient algorithm in POEMS for generating and solving the coupled equations of motion.

1. Introduction. Molecular dynamics (MD) simulations provide the methodology for detailed fine scale modelling on the molecular level. MD in the most traditional sense can be viewed as a process by which one generates atomic trajectories of a system of particles by direct numerical integration of Newton's equations of motion for each particle, with the appropriate initial and boundary conditions. This type of procedure has the advantage that as the integration/simulation progresses, the simulation yields important information not only about the intermediate states and the mechanisms which produced them, but also the rates at which these processes occur. Additionally, not just these states, but the predicted rates serve as valuable means for validating the models. Unfortunately MD simulation using standard atomistic models quickly run into significant challenges for all but the most elementary systems. This is because classical molecular dynamics (MD) propagates the motion of molecular models by solving the equations of motion for all the atoms in the model. In the fully atomistic case, (i.e. the Newton's equations of motion are derived and solved for every atom of the system), we have the most direct application of the physics involved, with the associated implementation being in many regards conceptually the simplest and easiest to apply. Unfortunately, due to the nature of the molecular interactions, specifically the stiffness of the bonds and other interaction, solving these equations (though straight forward) requires very fine time steps in order to maintain temporal integrator stability (the highest frequency of the systems must be accurately captured). Depending on the temporal integration method used, at least one system wide force determination (to drive the equations of motion) must be performed and this determination is extremely expensive (often the most costly aspect) for large systems. Thus, though conceptually simple and easy to implement, such simplistic brute force methods grind to an effective halt under the burden of their sub-femtosecond ($< 10^{15}$ sec.) required time steps and associated expensive force determinations.

Due to these difficulties many approaches have been developed in an attempt accelerate the simulations. Many efforts have focused on overcoming the strict time step limits in MD simulations. If larger stable integration time steps can be taken, then fewer expensive force determinations (which generally dominate the overall cost) are needed. This accelerates the simulation because the CPU time required is roughly

†Rensselaer Polytechnic Institute, (mukher@rpi.edu).
‡Sandia National Laboratories

proportional to the number of systems level force determinations executed. These efforts have primarily focused on removing (or at least not considering) the high frequency components of the system. It is these highest frequency components which govern the required temporal integration step size [1]. Examples of these approaches include: Constrained Dynamics Through Explicit Constraints [2], Constrained Dynamics Through Implicit Constraints via Generalized Coordinates [3], Reduced Computational Order (Cost) Algorithms [4], Multirate Temporal Integration [5] [6], Eigenvector (Modal) Schemes [7], Implicit integration schemes [8], Reduced Cost Force Determination [9] and the use of Multibody Dynamics Algorithms [4] among others.

Each of these approaches offers it own advantages and real disadvantages. Outwardly these methods appear to have little in common other than the shared objective of performing accurate integration of the equations of motion in less time. Each in fact represents a form of model reduction. In this paper we outline a method of reduced order modelling of molecular systems as sub-structured multi-rigid body articulated systems and the integration of a molecular dynamics software with a multibody dynamics software to facilitate this modelling effort.

2. Modelling Approach. The fundamental idea behind this work is the coarse-graining of select spatial domains in a molecular dynamics simulation into uncoupled and coupled rigid body systems. This process can be viewed as aggregation where a large number of discreet particles such as atoms or molecules are constrained to move as either a single rigid body or a system of articulated rigid bodies connected by kinematic joints. This modelling approach is a valid if the in the spatial domain in consideration, the relative motion of these discreet particles is limited or localized.

2.1. Kinematic Model. The dynamics of a single rigid body as modelled using Newton-Euler equations of motion would include three degrees of translational motion and three degrees of orientational change. The translational degrees of freedom are easily those associated with the three Cartesian coordinates of the center of mass of the body. The time derivatives of these degrees of freedom give rise to the translational velocity and acceleration of the body. Similarly the time derivatives of the orientational degrees of freedom (or some combination of the same) may give rise to the angular velocity and angular acceleration of the body. However the choice of the orientational degrees of freedom can be tricky and result in numerical difficulties. The easiest way to model the orientation would correspond to the three Euler angles with each angle associated with a rotation about each of the Cartesian directions. Although commonly used, this choice of degrees of freedom can result in numerical singularities arising from dependency in the kinematic equations relating the orientation angles. To overcome this singularity, quaternion or Euler parameters are used in modelling the orientation. Euler parameters are a set of four parameters related to each other through a constraint equation. Use of these parameters result in robust kinematic equations which never suffer from numerical singularities. The time derivative of the Euler parameters are related to the angular velocity of the body. The angular velocity of the body is the time rate of change of orientation of the body and modelled as having a component about each of the Cartesian directions.

There are two different representations of these reduced order models. The first are single rigid bodies where there are no kinematic coupling between the dynamics of the bodies. The interactions between these rigid bodies are modelled explicitly through the use of force-fields such as all atom or unified atom interactions. The second type are coupled rigid bodies. Two rigid bodies are coupled when they share a common atom. This common atom is treated as a kinematic joint location. In these

models, along with the force-field interactions, there exist constraint inertial loads between bodies. This is because the bodies are coupled at the common atom location and a kinematic constraint exists between any two bodies in the coupled system. A set of coupled rigid bodies are hence forth referred to as chains. A system may have any number of chains in it. Different chains interact only through the force field interactions as there are no inertial coupling between different chains.

Another type of reduced order model which again reduces to an articulated chain topology is the axial bond constrained systems. The fully atomistic representation of these systems consist of atoms or beads connected to each other by stiff joints. In the reduced order model, the axial stiff spring is replaced by a constant length massless rigid link. Each link and the next bead it is attached to is treated as a single body with unit mass and negligible inertia. The successive bodies are connected to the base body by joints allowing only rotational degrees of freedom.

In our model, the chains are free floating. The base body is modelled as connected to the inertial reference frame by a six degree of freedom joint allowing relative translational and rotational degrees of freedom. Each joint is modelled using Euler parameters to avoid any singular configurations. Unlike the SHAKE or RATTLE formulations which require additional nonlinear equations to be solved to impose the constraints, our model does not require solving any additional equations for maintaining the constraints. Further while SHAKE and RATTLE impose the constraints iteratively and only to a specified tolerance, our modelling approach enforces the constraints non-iteratively and exactly with no constraint violation. The constraints are imposed implicitly through the use of relative (internal) coordinates which reduces the formulation to a minimum set of ordinary differential equations.

2.2. Generating the rigid body properties. The calculation of the the total mass, position of the center of mass and the velocity of the center of mass of any rigid body is a simple from the properties of all the atoms that are aggregated into that rigid body. However calculating the inertia matrix and the angular velocity of the bodies is more involved. The calculation of the inertia matrix of the body involves taking the second moment of the mass of each atom about the center of mass of the rigid body. As this inertia matrix would vary with the motion of the body, the temporally invariant principle moments of inertia are calculated by solving an eigen value problem from the calculated inertia matrix. The eigen value problem also produces the three principle directions associated with the principle moments. These directions are treated as the basis vectors of the body based reference frame thereby forming the transformation matrix from the body basis to the Newtonian basis. To calculate the angular velocity, the angular momentum of the system of atoms is calculated about the center of mass of the body by summing the first moment of momenta of each atom. It is then converted to the body basis. This now equals the product of the diagonal inertia matrix and the unknown angular velocity, which can now be easily calculated by a scalar division.

3. Algorithm Overview. In this section an overview of the recursive $O(n)$ algorithm is presented. This algorithm uses the projection method as promoted by Kane and others [10] and uses internal coordinates instead of the Cartesian coordinates to formulate and solve the equations of motion. The algorithm begins by generating a topology or connectivity map of the chains in terms of relative coordinates, joints and body fixed reference frames. Each body is associated with its own body fixed dextral set of unit vectors. The joint locations, inertia values and the generalized velocities are expressed with respect to the body fixed reference frames. One end of the chain

is chosen as the base body and it is connected to the inertial reference frame by a kinematic joint. Each successive body is connected by two kinematic joints, one to an inward body and the other to the outward body on the chain. The orientation and motion of a body is expressed in terms of the admissible degrees of freedom of the joints. The linear and angular velocities of a body are expressed as invertible linear combinations of the generalized speeds i.e. time derivatives of the relative degrees of freedom. The algorithm works in three recursive sweeps or traversals. The first sweep begins at the base body and moves outwards to the tip while recursively generating the kinematic preliminaries like partial velocities [10], inertias and applied forces from the known states of the system. The second traversal then begins at the tip and recursively moves inwards towards the base body. This is a triangulation traversal and it recursively generates the articulated compound inertias and forces. This traversal is equivalent to successively shifting the inertias and the forces inwards towards the base body. Since the boundary conditions are known at the base body, at the end of the triangulation sweep the equations of motion for the base body can be solved to generate the state derivatives of the base body. Starting at the base body the third traversal works outward while recursively solving for the state derivatives of each successive body. By the time the sweep ends at the tip, the state derivatives are all solved in an efficient $O(n)$ complexity.

3.1. Mathematical Preliminaries. To aid in the subsequent development, consider the notation associated with the description of an arbitrary set of interconnected rigid bodies shown in figure (3.1). For this system, proximal (parent) body $Pr[k]$ is connected to its child body k through joint-k, via joint points k^- and k^+ which reside in bodies $Pr[k]$ and k, respectively. Similarly, the *distal* (child) bodies of body k are given as members of the set of bodies $Dist[k]$. The position vector $\boldsymbol{s}^k$ locates joint-k relative to the mass center of body $Pr[k]$, while the position vector $\boldsymbol{r}^k$ locates the mass center of body k with respect to the outboard end of this same joint. It will also prove convenient to describe the position of child mass center k^* relative to proximal mass center $Pr[k]^*$ by the vector $\vec{\gamma}^k$.

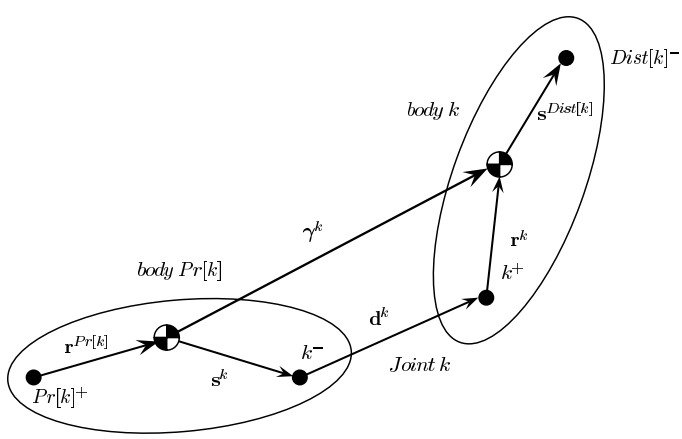


FIG. 3.1. *Notation Associated with an Arbitrary Set of Interconnected Rigid Bodies*

The angular velocity of any body k with respect to the Newtonian reference frame N, and velocity of its associated mass center k^* may always be written in terms of

the generalized speeds as

$$\boldsymbol{\omega}^k = \sum_{r=1}^{n} \boldsymbol{\omega}_r^k u_r + \boldsymbol{\omega}_t^k \tag{3.1}$$

and

$$\boldsymbol{v}^{k^*} = \sum_{r=1}^{n} \boldsymbol{v}_r^{k^*} u_r + \boldsymbol{v}_t^{k^*}. \tag{3.2}$$

In these expressions $\boldsymbol{\omega}_r^k$ and $\boldsymbol{v}_r^{k^*}$ are termed the rth *partial angular velocity of body* k and r^{th} *partial velocity of point* k^*, in N, respectively. These quantities may be thought of as basis vectors for the space of admissible system velocities and angular velocities, while the associated generalized speeds are the velocity space measure numbers. Additionally, the terms $\boldsymbol{\omega}_t^k$ and $\boldsymbol{v}_t^{k^*}$ appearing in equations (3.1)–(3.2), are referred to as the *angular velocity remainder term* of body k and *velocity remainder term* of point k^*, in N, respectively. These quantities are most often associated with specified/prescribed motion, and thus are not associated with the time derivative of a system degree of freedom.

When deriving this method it is often convenient to express quantities in a scalar matrix, as opposed to a tensor (vector and dyadic) form. For this purpose an arbitrary vector $\boldsymbol{\vartheta}^k$ will be represented in matrix form as $\underline{\vartheta}^k$, which is associated with the local dextral orthogonal unit vectors $\hat{k}_1$, $\hat{k}_2$, $\hat{k}_3$, fixed in body k. One may then define the *velocity*, *partial velocity*, and *velocity remainder term* matrices as

$$\underline{\mathcal{V}}^k = \begin{bmatrix} \underline{\omega}^k \\ \underline{v}^{k^*} \end{bmatrix}, \qquad \underline{\mathcal{P}}_r^k = \begin{bmatrix} \underline{\omega}_r^k \\ \underline{v}_r^{k^*} \end{bmatrix}, \quad \text{and} \quad \underline{\mathcal{V}}_t^k = \begin{bmatrix} \underline{\omega}_t^k \\ \underline{v}_t^{k^*} \end{bmatrix}. \tag{3.3}$$

With these matrices so defined, equations (3.1) and (3.2) may be expressed as

$$\underline{\mathcal{V}}^k = \overline{\underline{\mathcal{V}}}^k + \underline{\mathcal{V}}_t^k = \sum_{r=1}^{n} \underline{\mathcal{P}}_r^k u_r + \underline{\mathcal{V}}_t^k. \tag{3.4}$$

One can similarly represent the generalized acceleration matrix of an arbitrary body k as defined in previous works [11], [12], as

$$\underline{\mathcal{A}}^k = \begin{bmatrix} {}^N\underline{\alpha}^k \\ {}^N\underline{a}^{k^*} \end{bmatrix}, \tag{3.5}$$

where $\underline{\mathcal{A}}^k$ may also be divided into two portions. One is $\overline{\underline{\mathcal{A}}}^k$, which contains all terms which are explicit in the unknown state derivatives $\dot{\underline{u}}$ and the other is the *acceleration remainder term* $\underline{\mathcal{A}}_t^k$, which represents all of the other acceleration terms (and may be calculated directly from the system state), giving

$$\underline{\mathcal{A}}^k = \overline{\underline{\mathcal{A}}}^k + \underline{\mathcal{A}}_t^k. \tag{3.6}$$

4. $O(n)$ Forward Dynamics Analysis. The basic recursive $O(n)$ algorithm for performing forward dynamics simulation associated with tree-structure systems consists for three principal steps, or "sweeps". These steps are the *Kinematics Sweep*, the *Triangularization Sweep*, and the *Back Substitution Sweep*.

4.1. Recursive Kinematic Relationships. The Kinematic sweep starts at the base body and works outward to the tip while recursively using the kinematic relations to generate the partial velocities, transformation matrices, translational and rotational velocities and state dependent acceleration terms in the body basis. With the generalized velocity, generalized acceleration, and generalized acceleration remainder term matrices as represented above, it is possible to compactly represent the recursive relationships necessary for determining all system kinematic quantities. As has been demonstrated in [11] we have

$$\underline{\mathcal{V}}^k = \left[(\underline{\mathcal{S}}^k)^T \overline{\underline{\mathcal{V}}}^{Pr[k]} + \underline{\mathcal{P}}^k_k \underline{u}_k\right] + \underline{\mathcal{V}}^k_t, \tag{4.1}$$

and

$$\underline{\mathcal{A}}^k = \left[(\underline{\mathcal{S}}^k)^T \overline{\underline{\mathcal{A}}}^{Pr[k]} + \underline{\mathcal{P}}^k_k \dot{\underline{u}}_k\right] + \underline{\mathcal{A}}^k_t. \tag{4.2}$$

The quantity $\underline{\mathcal{S}}^k$ appearing in equations (4.1)–(4.2) is the basis consistent linear transformation matrix

$$\underline{\mathcal{S}}^k = \begin{bmatrix} \underline{\mathcal{C}}^k & \underline{\mathcal{C}}^k \underline{\gamma}^k_\times \\ \underline{0} & \underline{\mathcal{C}}^k \end{bmatrix}_{6\times 6}. \tag{4.3}$$

Within this expression $\underline{\mathcal{C}}^k \equiv {}^{Pr[k]}\underline{\mathcal{C}}^k$ is the direction cosine matrix which relates the body k basis vectors to those fixed in its parent body $Pr[k]$; $\underline{0}$ is a 3×3 zero matrix; and $\underline{\gamma}^k_\times$ is the skew symmetric matrix equivalent to the vector cross product operation $\boldsymbol{\gamma}^k\times$. The *shift matrix* transformation $\underline{\mathcal{S}}^k$ converts a system of forces and moments acting through the center of mass of k, to an equivalent force system, acting though a point of k which is instantaneously coincident with the center of mass of $Pr[k]$.

At this time, it is also convenient to define the body k *generalized inertia* $\underline{\mathcal{I}}^k$ and the body k *generalized force* $\underline{\mathcal{F}}^k$ matrices

$$\underline{\mathcal{I}}^k = \begin{bmatrix} \underline{I}^{k/k^*} & \underline{0} \\ \underline{0} & \underline{M}^k \end{bmatrix}_{6\times 6}, \tag{4.4}$$

$$\underline{\mathcal{F}}^k = \begin{bmatrix} \underline{T}^k - (\underline{I}^{k/k^*} \underline{\alpha}^k_t + \underline{\omega}^k_\times \underline{I}^{k/k^*} \underline{\omega}^k) \\ \underline{R}^k - \underline{M}^k \underline{a}^k_t \end{bmatrix}_{6\times 1}. \tag{4.5}$$

Within these expressions, $\underline{I}^{k/k^*}$ is the 3×3 central inertia matrix of body k, and $\underline{M}^k$ is the diagonal translational mass matrix of this same body. By comparison $\underline{T}^k$ and $\underline{R}^k$ represent the resultant force system of all moments and forces, respectively, acting on body k through its center of mass k^*.

4.2. Triangularization. The triangularization procedure works recursively inward to compute the articulated inertia mass matrix $\boldsymbol{\mathcal{I}}^k_3$ and the articulated generalized active force $\boldsymbol{\mathcal{F}}^k_3$ expressing as

$$\boldsymbol{\mathcal{I}}^k_3 = \boldsymbol{\mathcal{I}}^k_1 + \sum_{j\,\in\, Dist[k]} \boldsymbol{\mathcal{T}}^j \boldsymbol{\mathcal{I}}^j_3 \left(\boldsymbol{\mathcal{S}}^j\right)^T, \tag{4.6}$$

$$\boldsymbol{\mathcal{F}}^k_3 = \boldsymbol{\mathcal{F}}^k_1 + \sum_{j\,\in\, Dist[k]} \boldsymbol{\mathcal{T}}^j \boldsymbol{\mathcal{F}}^j_3, \tag{4.7}$$

where $Dist[k]$ is the *distal* (children) set associated with body k. The triangularization operator $\boldsymbol{\mathcal{T}}^j$ and the basis consistent shifting operator $\boldsymbol{\mathcal{S}}^j$ used in (4.6)-(4.7) are defined as

$$\boldsymbol{\mathcal{T}}^j = \left(\boldsymbol{\mathcal{S}}^j\right)^T \left[\boldsymbol{U} - \frac{1}{\mathcal{M}_j} \boldsymbol{\mathcal{I}}_3^j \boldsymbol{\mathcal{P}}_j^j \left(\boldsymbol{\mathcal{P}}_j^j\right)^T\right] , \tag{4.8}$$

$$\boldsymbol{\mathcal{S}}^j = \begin{bmatrix} \boldsymbol{\mathcal{C}}^j & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}}^j \end{bmatrix} \begin{bmatrix} \boldsymbol{U} & \boldsymbol{\gamma}_\times^j \\ \boldsymbol{0} & \boldsymbol{U} \end{bmatrix} . \tag{4.9}$$

In (4.8) and (4.9), matrix $\boldsymbol{\mathcal{C}}^j$ is the direction cosine matrix relating to local basis vector of body $Dist[k]$ to k, $\boldsymbol{U}$ is an identity matrix, and $\boldsymbol{\gamma}_\times^j$ is the matrix representation of the vector cross product. The quantities $\mathcal{M}_k$ is also given by

$$\mathcal{M}_k = \left(\boldsymbol{\mathcal{P}}_k^k\right)^T \boldsymbol{\mathcal{I}}_3^k \boldsymbol{\mathcal{P}}_k^k , \tag{4.10}$$

with $\boldsymbol{\mathcal{P}}_k^k$ defined as

$$\boldsymbol{\mathcal{P}}_k^k \equiv \begin{bmatrix} {}^N\boldsymbol{\omega}_k^k \\ {}^N\boldsymbol{v}_k^k \end{bmatrix} . \tag{4.11}$$

4.3. Back-Substitutions. At the base body, information associated with an entire set of outboard bodies has all been accumulated and is explicitly available such that the equation $\mathcal{M}_1 \dot{\boldsymbol{u}}_1 = \boldsymbol{\mathcal{P}}_1^1 \boldsymbol{\mathcal{F}}_3^1$ can be isolated and yields the solution of $\dot{\boldsymbol{u}}_1$. The solution for $\dot{\boldsymbol{u}}_1$ is then substituted into the next equation to solve for $\dot{\boldsymbol{u}}_2$. Proceeding in this manner, a generalized function expression for the solution of each generalized acceleration $\dot{\boldsymbol{u}}_k$ is given as follows

$$\dot{\boldsymbol{u}}_k = \frac{\left(\boldsymbol{\mathcal{P}}_k^k\right)^T}{\mathcal{M}_k} \left[\boldsymbol{\mathcal{F}}_3^k - \boldsymbol{\mathcal{I}}_3^k \left(\boldsymbol{\mathcal{S}}^k\right)^T \boldsymbol{\mathcal{A}}^{k-1}\right] , \tag{4.12}$$

with $\boldsymbol{\mathcal{A}}^k$ computed from

$$\boldsymbol{\mathcal{A}}^k = \left(\boldsymbol{\mathcal{S}}^k\right)^T \boldsymbol{\mathcal{A}}^{k-1} + \boldsymbol{\mathcal{P}}_k^k \dot{\boldsymbol{u}}_k . \tag{4.13}$$

5. Time Integration. Velocity-Verlet temporal integration scheme for temporally advancing a dynamics simulations has been extensively used for atomistic simulations. Velocity-Verlet algorithms are symplectic and gives very good energy conservation characteristics. However for reduced order models involving coupled bodies, the performance of the Velocity-Verlet is not as good as with atomistic simulations. This is because the velocity dependent inertial forces such that the gyroscopic and Coriolis forces come into play for the reduced order models. Further, the motion of individual bodies in the articulated system are coupled and the motion of one affects all others resulting in constraint forces acting on the bodies. Also, as compared with an atomistic simulation, the integrands are not the cartesian accelerations and velocities but the time derivatives of the generalized coordinates. For these reasons, the multibody dynamics equations of motions have been traditionally integrated by high order methods such as the Runge-Kutta 4-5 schemes. These schemes have traditionally been highly accurate and give good energy conservations for macroscopic problems like aerospace applications. However when applied to reduced order molecular dynamics simulations, these methods quickly become computationally expensive

as they require four expensive force calculations per integration step. For the systems studied in our work, the Lobatto III a-b partitioned Runge-Kutta integration scheme has been used. It is a second order method which iteratively calculates the velocities at the half step. The iteration quickly converges in one or two steps. The method efficiently accommodates for the velocity dependent inertial forces and requires only one force calculation per integration step.

6. Applications and Results. Discussed in this section are the test cases simulated to verify the validity of this development. The primary objective of simulating these test cases is to be able to reproduce the previously published results, and validate the stability of the simulations. Because of space constraints, the details of the results are not shown and can be found in an upcoming journal article.

6.1. Water Box. The fundamental test case to check the coupling of the two softwares is a box of waters consisting of 512 water molecules in NVE ensemble using different time steps of 0.5, 1 and 2 fs for a total simulation length of 100 ps. Each water molecule is treated as a single rigid body and the mass properties are calculated using the method outlined above. All atom force fields are used under periodic boundary conditions. The dynamics of the rigid bodies is simulated using POEMS and compared with the results obtained by imposing holonomic constant bond length constraints using SHAKE. While in SHAKE the Newton-Euler equations of motion are solved directly to generate the dynamics, in POEMS, Kane's [10] equations of motion are used. In either case each rigid body is modelled as having three degrees of translational freedom modelled using Cartesian displacements and three degrees of rotational freedom modelled using body based reference frames and Euler parameters. In SHAKE the temporal integration was carried out using Velocity Verlet while the Lobatto III A-B scheme was used in POEMS. The energy conservation in the simulations is calculated using the ratio of standard deviation in energy to the mean energy value as the comparison metric. The simulations at different time steps showed good energy conservation with smaller time steps giving better energy conservation. To analyze the simulation results we calculated the mean square displacement of the water molecules and the coefficient of diffusion using Einstein's equation. We also compared the thermodynamic properties using the POEMS and SHAKE approach. The results between the two sets of simulations were found to be in very good agreement.

6.2. Alanine Dipeptide. This molecule, CH3CONHCH(CH3)CONH, is small enough to avoid complexities of structure and yet has interesting dynamic behavior which makes it a prime candidate for initial simulation test to validate the method. The molecule has been sub-structured into two rigid bodies each corresponding to the -CONH unit while the remaining atoms are treated as bond constrained particles. All atom forces were calculated using LAMMPS while solving the multibody dynamics equations of motion of the sub-structured articulated system and the temporal integration were handled by POEMS. The simulation results, including energy conservation, structural properties and configurational parameters were found to match with those generated using atomistic simulations. Further, because of the coarse-graining, an improvement of an order of magnitude in the integration time step was achieved.

6.3. DNA simulations. In these simulations we modelled the bond constrained dynamics of tethered DNA strands of lengths of 16 and 32 atoms. The fully atomistic interactions consist of FENE bonds and truncated Lennard Jones and Coulomb interactions. The DNA strands are tethered to a membrane which is modelled using

a 9-3 Lennard Jones wall potential. In the reduced order model, the FENE bonds are constrained to fixed lengths modelled as massless rigid links. This reduces it to an articulated serial chain system modelled using POEMS. The simulation results were compared and validated against an atomistic simulation and an improvement in the integration time step of up to a factor of 6 was achieved.

6.4. Box of alkanes. We simulated a box of alkanes to validate the performance of the modelling scheme with an united atom potential. The systems under consideration were three boxes containing 216 chains of alkanes each of chain lengths 8,16 and 32 modelled under periodic boundary conditions in a NVE ensemble. The united atom Trepp3 force field was used in these simulations. The axial vibrations of the beads were constrained by modelling the stiff bonds as fixed length massless rigid links. This rendered the model as articulated chains of point masses connected by rigid links and kinematic joints. Three different time-steps of 1fs, 5fs and 10fs were used in the simulations. For these simulations too we used the energy metric discussed above to compare the simulations results with those obtained using SHAKE. The united atom force fields are smoother than the Lennard Jones potential and hence a better energy conservation at larger time steps were expected using the articulated rigid body representation. We were able to achieve stable simulations with good energy conservation with increase in time steps by an order of magnitude.

6.5. C-Terminal of Ribosomal. This is a larger problem which provides better understanding of the performance of the methodology for complex biological systems. The C-terminal fragment (1CTF) of the L7/L2 ribosomal protein from E.coli is used for this simulation. This system has been simulated in several references with different substructuring schemes. In this method the system is sub-structured into 31 small rigid bodies with hinges at the ϕ or ψ angles. The simulation was monitored for energy conservation and preservation of essential dynamics.

6.6. C-Terminal of Rubisco. We have built a model of the RuBisCO (Ribulose-1, 5-Bisphosphate Carboxylase/Oxygenase) enzyme for simulation using our coupled LAMMPS POEMS simulation software. The fully atomistic model of the C-terminal of RuBisCo consists of 510 atoms modelled using harmonic bond potentials, CHARMM angle and dihedral potentials and non-bonding CHARMM Lennard-Jones and Coulomb force fields with cut offs at 8 and 10. In the sub-structured model, the system consists of 11 rigid bodies connected together to form an articulated serial chain topology. All atom explicit force calculations are supported by this model. However intra-body interactions between the atoms that make up a rigid body are ignored as these would sum to zero. This is a modest sized problem which is a good example to validate the modelling approach. Using this model, NVE simulations were run for 1 nano-second. Different time steps were used to determine the drift in energy as a function of time-step. Ignoring the intra-body atomistic interactions give an immediate computational saving. Further, by using a rigid body model, an increase in the integration time step by an order of magnitude was observed.

6.7. Rhodopsin. Rhodopsin is a G-protein coupled receptor with a defined tertiary structure. It is a good example to study transduction and a large amount of experimental results are available about its structure and function. We have generated a sub-structured articulated rigid body model of the rhodopsin protein and have simulated the same with the LAMMPS-POEMS coupling. This is a fairly large system with the fully atomistic model consisting of about 5000 atoms. This atomistic model is sub-structured into 26 connected rigid bodies that form an articulated

serial chain topology. Similar to the RuBisCo model, the Rhodopsin model was simulated in NVE ensemble for 1 nanosecond at different time steps. The simulations showed good energy conservation at larger time steps. The coarse-graining provided significant computational savings in the calculations of the force interactions as the intra-body interactions were not calculated. Further using this model, we were able to obtain stable simulation with an increase in the time step by an order of magnitude.

7. Software Development. In this section the generation of the open source computational tool is discussed. The two research codes that are fundamental in this work are the LAMMPS: **L**arge-scale **A**tomic/**M**olecular **M**assively **P**arallel **S**imulator software from Sandia National Laboratories and the multibody dynamics software POEMS: **P**arallelizable **O**pen-source **E**fficient **M**ultibody **S**oftware. LAMMPS is an open source code, with a GPL type license. Under development by the primary author, Steve Plimpton, and others since the mid 1990s, LAMMPS is a general purpose classical molecular dynamics code [13]. The POEMS code [14] is also open source, with a BSD type license. This is a general purpose multibody dynamics research code being developed by Rudranarayan Mukherjee and other members of Prof. Anderson's research group at Rensselaer Polytechnic Institute. The two softwares have different functionalities which are complementary in nature. While LAMMPS is a classical molecular dynamics code with emphasis on atomistic simulations, POEMS is a multibody dynamics code with an emphasis on modelling dynamics of reduced order or coarse-grained models. A brief overview of both these softwares is presented in the next section.

LAMMPS is a classical molecular dynamics code that models an ensemble of particles in a liquid, solid, or gaseous state. LAMMPS can model atomic, polymeric, biological, metallic, or granular systems using a variety of force fields and boundary conditions. It runs efficiently on single-processor desktop or laptop machines, but is designed for parallel computers. In classical molecular dynamics, inter-particle force calculations are the most expensive part of MD simulations and hence have been carefully optimized and made more efficient as MD codes have matured. Among the many force fields (FF) currently available in LAMMPS are the commonly-used CHARMM [15] and AMBER [16] biomolecular FF. LAMMPS also has all of the capabilities commonly required in biomolecular simulation, including full long-range electrostatics capabilities using Ewald or particle-particle/particle-mesh (PPPM, similar to particle-mesh Ewald), SHAKE bond and angle constraints, rRESPA [17] hierarchical timestepping, and NVE, NVT, and NPT integrators. LAMMPS also has the capability to simulate hybrid bio/non-bio systems through the superimposing of force fields.

Though significant speedup can be gained from efficiently calculating the forces, which is the forte of LAMMPS, further substantial computational gains can be realized if the systems are coarse-grained by enforcing kinematic constraints. This is because imposing kinematic constraints can efficiently eliminate high frequency components thereby allowing larger temporal integration time-step. This results in a multiplying effect on improving simulation speed through combining larger integration step sized and reduce force calculation costs. However, generating and solving the equations of motion of reduced order models particularly those which represent coupled multibody systems can be challenging and unless some efficient formulation is resorted to, the computational cost can be as high as $O(n^3)$ where n is the number of degree of freedom in the system. LAMMPS integrates Newton's equations of motion for collections of atoms, molecules, or macroscopic particles and does not include any

efficient formulations for effectively formulating and solving the equations of motions of reduced order models.

This aspect has been addressed in POEMS. POEMS was written as a generic multibody simulation code which can efficiently handle the dynamics aspect of the reduced order models. POEMS is an object-oriented C++ research package for simulating the forward dynamics of multibody systems. Its emphasis was also placed on application to large ($n \gg 1$) systems, i.e. coupled systems involving many generalized coordinates. Majority of the effort in algorithm development has been oriented toward methods that perform well with applied to coupled systems involving many generalized coordinates. This code features libraries of different dynamics formulations for efficiently generating and solving the equations of motion of articulated system as well as different time integration schemes for advancing a simulation temporally. Commonly used kinematic and dynamic identities, organization of multibody topologies, and data structure with matrix manipulations which are generic to most multibody algorithms are built into the software using an object oriented design.

The software in its current form has three algorithms for solving equations of motion of articulated multi-rigid body systems in chain and tree topologies.

- *KaneSolver()* : The $O(n^3)$ complexity solver based on Kane's method [10].
- *OnSolver()* : The $O(n)$ complexity recursive solver based [12].
- *DCASolver()* : The Divide and Conquer method of $O(log(n))$ complexity [18].

It also contains an implementation of a generalized impulse momentum formulation for correct kinematic coarse-graining of the reduced order models. This is a novel feature that enforces the correct initial conditions required to preserve the essential dynamics of the systems. No other research package has a comparable formulation and it is a significant development as it allows smooth transition between models of different resolutions. Because of space constraints this algorithm is not discussed here and presented in another upcoming research paper.

Along with the solution of equations of motion, the software has temporal integration algorithms for temporal simulations. These include the following algorithms.

- Runge Kutta 4-5
- Predictor Corrector
- Verlet and Velocity Verlet
- Lobatto Patitioned III a-b Runge Kutta 4-5

By coupling together these two software we have created a synergistic simulation tool which is freely available to all researchers working in molecular dynamics. The inherent features of these code are complementary, with LAMMPS focussed more on the efficient generation of force field and potential calculations while POEMS is aimed at efficiently handling the dynamics aspect of the reduced order models. This two fold approach is instrumental in accelerating molecular dynamics simulations and be applicable to a wide variety of problems in biomolecular and materials modelling. POEMS is built into LAMMPS as an external library and is distributed along with LAMMPS.

8. Conclusions. A novel method based on sub-structured coarse grained models of molecular dynamics systems is developed, implemented and validated. This method uses efficient $O(n)$ complexity multibody dynamics algorithms for modelling the forward dynamics of these sub-structured models. A new computational tool is developed and released for public use under open source licensing. This computational tool culminates from coupling together the molecular dynamics code LAMMPS with the multibody dynamics software POEMS.

Acknowledgment. The authors thank Steve Plimpton and Kurt Anderson for their support and help in this endeavor.

REFERENCES

[1] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer, 2nd edition, 1993.

[2] H. C. Anderson. Rattle: A 'Velocity' version Shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 54:24–34, 1983.

[3] R. A. J. Abagyan and A.K. Mazur. New methodology for computer-aided modelling of biomolecular structure and dynamics. *Journal of Biomolecular Structure*, 6:833–845, 1989.

[4] H. M. Chun, C. E. Padilla, D. N. Chin, M. Watenabe, V. I. Karlov, H. E. Alper, K. Soosaar, K. B. Blair, O. M. Becker, L. S. D. Caves, R. Nagle, D. N. Haney, and B. L. Farmer. MBO(N)D: A multibody method for long-time molecular dynamics simulations. *Journal of Computational Chemistry*, 21(3):159–184, 2000.

[5] M. E. Tuckerman and B. J. Berne. Molecular dynamics in systems with multiple time-scales: Systems with stiff and soft degrees of freedom and with short and long range forces. *Joural of Computational Chemistry*, 95:8362–8364, 1992.

[6] M. Watanabe and M. Karplus. Dynamics of molecules with internal degrees of freedom by multiple time-step methods. *Journal of Chemical Physics*, 99:8063–8074, 1993.

[7] G. Zhang and T. Schlick. LIN: A new algorithm combining implicit integration and normal mode techniques for molecular dynamics. *Journal of Compuational Chemistry.*

[8] C. S. Peskin and T. Schlick. Molecular dynamics by the backward Euler's method. *Communications in Pure and Applied Math*, 42:1001–1031, 1989. in Press.

[9] L.Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 135(2):280–292, 1997.

[10] T. R. Kane and D. A. Levinson. *Dynamics: Theory and Application.* Mcgraw-Hill, NY, 1985.

[11] K. S. Anderson. An order-n formulation for motion simulation of general constrained multi-rigid-body systems. *Computers and Structures*, 43(3):565–572, 1992.

[12] K. S. Anderson. An order-n formulation for the motion simulation of general multi-rigid-body tree systems. *Computers and Structures*, 46(3):547–559, 1993.

[13] J.M. Haile. *Molecular Dynamics Simulation : Elementary Methods.* Wiley Interscience, New York, 1992.

[14] K.S. Anderson, R.Mukherjee, J.H. Critchley, J. L. Ziegler, and S.R. Lipton. Poems: Paralleizable open-source efficient multibody software. *Engineering with Computers*, 2005. In Review.

[15] M. Karplus. CHARMM (Chemistry at Harvard Macromolecular Mechanics). http://brooks.scripps.edu/, 2005. Brooks Group Computational BioPhysics and Chemistry.

[16] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. R. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Computer Physics Communications*, 91:1–41, 1995.

[17] S. J. Plimpton and M. Stevens. Particle–mesh ewald and rrespa for parallel molecular dynamics simulations. In *Proceedings Eighth SIAM Conference on Parallel Processing for Scientific Computing*, Minneapolis, MN, March 1997. DETC2005-85480.

[18] R. Featherstone. A divide-and-conquer articulated body algorithm for parallel $O(\log(n))$ calculation of rigid body dynamics. Part 1: Basic algorithm. *International Journal of Robotics Research*, 18(9):867–875, Sep. 1999.

BLOCK PRECONDITIONERS APPLIED TO INDUCED-CHARGE ELECTRO-OSMOSIS

R. SHUTTLEWORTH*, V. HOWLE†, K. LONG‡, J. TEMPLETON‡, AND R. TUMINARO‡

Abstract. Over the past several years, considerable effort has been placed on developing efficient solution algorithms for the incompressible Navier–Stokes equations. The effectiveness of these methods requires that the iterative solution techniques for certain linear subproblems exhibit robust and rapid convergence. We begin by introducing methods that have recently been developed for solving incompressible flow problems, which are based on an approximation of the Schur complement technique developed by Elman, Howle, Shadid, Shuttleworth, and Tuminaro [3], Kay, Loghin, and Wathen [7], and Silvester, Elman, Kay, and Wathen [12]. Then we describe how we apply these techniques to solving a flow over a diamond obstruction generated by the stablilized finite element code, MPSalsa, then expand this work to a fixed 'induced-charge electro-osmosis' (ICEO) microfluidic problem posed in the high level symbolic differentiation finite element code, Sundance.

1. Introduction. We consider flow problems where the microscale mixing of fluids must occur without the help of turbulence, but by molecular diffusion alone. From a modeling perspective, our flow problems have a length scale between 10-100 μm with a mixing time in the hundreds of seconds. Moreover, the fluid volume is very low (several nanoliters) and the Reynolds number is far less than 100 and normally less than 3. This results in laminar flow and can be commonly found in modeling blood samples, bacterial cell suspensions, or protein/antibody solutions.

In this model, the flow is pumped by either a pressure driven flow or by electrokinetic means. For pressure driven flow, the fluid being studied is pumped via positive displacement. We focus our efforts on electrokinetic displacement, where the walls of the microchannel are charged, generating a "double layer" of ions on the walls of the channel. Then, when an electric field is applied to this channel the double layer moves to the opposite polarity, thus creating motion of fluid near the walls. This motion is transferred to the bulk of the fluid creating mixing and movement of fluid in the channel.

We numerically model this phenomenon using the incompressible Navier-Stokes equations with an electric field equation used in the boundary conditions of the problem in question. Accurate, robust, and scalable solutions of the Navier-Stokes equations is an active and open research topic. Previous work in this area has developed solution methods that generate iteration counts independent of the size of the mesh. We hope to expand this recent work [3] for use in microfluidic applications.

Section 2 gives a brief background on our techniques for solving these problems. In Section 3, we detail the algebraic approach to our preconditioning operator. Section 4 provides a brief discussion of the implementation of the linear and nonlinear solvers. Details of the numerical experiments and the results of these experiments are discussed in Section 5. Concluding remarks are provided in Section 6.

2. Background. We can model the microfluidic behavior described above using the incompressible form of the Navier–Stokes equations

$$\begin{aligned} \alpha \mathbf{u}_t - \nu \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \mathrm{grad})\, \mathbf{u} + \mathrm{grad}\, p &= \mathbf{f} \\ -\mathrm{div}\, \mathbf{u} &= 0 \end{aligned} \tag{2.1}$$

*University of Maryland, College Park
†Sandia National Laboratories

in $\Omega \subset \mathbb{R}^d (d = 2 \text{ or } 3)$. Here the velocity, $\mathbf{u}$, satisfies suitable boundary conditions on $\partial\Omega$, p represents the hydrodynamic pressure and $\mathbf{f}$ the body forces. For the electrokinetic problem, which drives the flow, the boundary condition is modeled by Poisson's equation, i.e.

$$\nabla^2 \phi = f, \tag{2.2}$$

where ϕ is the electric potential to be determined from a given charge distribution, f.

We will be concerned with solution algorithms for the algebraic system of equations that result from Newton or Oseen linearization and discretization of these equations. The coefficient matrices have the form

$$A = \begin{pmatrix} F & B^T \\ \hat{B} & -C \end{pmatrix} \tag{2.3}$$

where F is the convection-diffusion-like operator, B^T the gradient operator, $\hat{B}$ is (a perturbation of) the divergence operator, and C is the operator that stabilizes the finite element discretization. The operator C can be zero or nonzero depending upon the stabilization and boundary conditions used in the discretization of the problem. The strategies we employ for solving (2.3) (which we will also refer to as the Jacobian or saddle point system) are derived from the LDU block factorization of this coefficient matrix,

$$A = \begin{pmatrix} I & 0 \\ \hat{B}F^{-1} & I \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -S \end{pmatrix} \begin{pmatrix} I & F^{-1}B^T \\ 0 & I \end{pmatrix}, \tag{2.4}$$

where

$$S = C + \hat{B}F^{-1}B^T \tag{2.5}$$

is the Schur complement (of F in A). They require methods for approximating the action of the inverse of the factors of (2.4) (the action of the inverse of S), which, in particular, requires approximation to the actions of F^{-1} and S^{-1}. Use of the exact Schur complement is not feasible, but replacement of S with carefully derived approximations leads to efficient *preconditioning* strategies for use with iterative solvers for a Krylov subspace method, such as GMRES.

3. Definition of Preconditioning Operator. We will develop our preconditioning strategy by lumping together the diagonal and upper triangular factors of (2.4). A discussion of the merits of choosing other combinations of factors can be found in [4]. The efficacy of choosing the diagonal and upper triangular factors as a preconditioner can be seen by analyzing the following generalized eigenvalue problem:

$$\begin{pmatrix} F & B^T \\ \hat{B} & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \lambda \begin{pmatrix} F & B^T \\ 0 & \hat{S} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}.$$

If $\hat{S}$ is the Schur complement, the eigenvalues of the preconditioned matrix are identically one. Further, this operator contains Jordan blocks of dimension at most 2, and consequently at most two iterations of a preconditioned GMRES iteration would be needed to solve the system [9].

To apply the preconditioner $\mathcal{Q}$ (i.e. the upper triangular and diagonal factors of (2.4)) in a Krylov subspace iteration, at each step the application of $\mathcal{Q}^{-1}$ to a vector is needed. By expressing this operation in factored form,

$$\begin{pmatrix} F & B^T \\ 0 & -S \end{pmatrix}^{-1} = \begin{pmatrix} F^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -S^{-1} \end{pmatrix} \tag{3.1}$$

the computational issues involved for the particular choice (3.1) can be seen. Two potentially difficult operations are required to apply $\mathcal{Q}^{-1}$ to a vector: S^{-1} must be applied to a vector in the discrete pressure space, and F^{-1} must be applied to a vector in the discrete velocity space. The application of F^{-1} can be performed relatively cheaply using an iterative technique, such as GMRES preconditioned with algebraic multigrid. However applying S^{-1} to a vector is too expensive. An effective preconditioner can be built by replacing this operation with an inexpensive approximation. Our approach will be called the pressure convection-diffusion (P-CD) strategy and will begin by defining a new operator, denoted F_p, used in approximating the Schur complement, using the following premise:

Suppose that we begin with a discrete version of a convection-diffusion operator derived by linearizing the Oseen version of the nonlinear term in (2.1),

$$(\nu\nabla^2 + (\mathbf{w} \cdot \text{grad})).$$

Here $\mathbf{w}$ is viewed as an approximation to the velocity from a previous nonlinear iteration. Suppose that there is an analogous operator defined on the pressure space,

$$(\nu\nabla^2 + (\mathbf{w} \cdot \text{grad}))_p.$$

Consider the commutator of these operators with the gradient:

$$\varepsilon = (\nu\nabla^2 + (\mathbf{w} \cdot \text{grad}))\nabla - \nabla(\nu\nabla^2 + \ (\mathbf{w} \cdot \text{grad}))_p. \tag{3.2}$$

Supposing that ε is small, and multiplying on both sides of (3.2) by the divergence operator and the inverse of the convection-diffusion operator, we get

$$\nabla^2(\nu\nabla^2 + (\mathbf{w} \cdot \text{grad}))_p^{-1} \approx \nabla \cdot (\nu\nabla^2 + \ (\mathbf{w} \cdot \text{grad}))^{-1}\nabla. \tag{3.3}$$

In discrete form, this becomes

$$\begin{aligned} (Q_p^{-1}A_p)(Q_p^{-1}F_p)^{-1} &\approx (Q_p^{-1}B)(Q_v^{-1}F)^{-1}(Q_v^{-1}B^T) \\ S = (BF^{-1}B^T) &\approx A_pF_p^{-1}Q_p \end{aligned}$$

where here F represents a discrete convection-diffusion operator on the velocity space, F_p is the discrete convection-diffusion operator on the pressure space, A_p is a discrete Laplacian operator, Q_v the velocity mass matrix, and Q_p is the pressure mass matrix. This suggests a suitable Schur complement approximation for a finite element discretization when $C = 0$. In the case of pressure stabilized finite element discretizations, the approximation is

$$C + BF^{-1}B^T \approx A_pF_p^{-1}Q_p. \tag{3.4}$$

A further discussion of this choice can be found in [2]. Applying the action of the inverse of $A_pF_p^{-1}Q_p$ to a vector requires solving a system of equations with a discrete

Laplacian operator, A_p, then multiplication by the matrix F_p, and solving a system of equations with the pressure mass matrix. Both the convection-diffusion-like system (with coefficient matrix F), and the Schur complement system (with coefficient matrix $A_p F_p^{-1} Q_p$), can be solved approximately using multigrid with little deterioration of effectiveness.

4. Implementation. Our implementation of the preconditioned Krylov subspace solution algorithm uses *Trilinos* [6], an effort at Sandia National Laboratories to develop parallel solution algorithms in an object-oriented collection of software packages for large-scale, parallel multiphysics simulations. One advantage of this is the capability to seamlessly use other Trilinos packages for core operations. We use the following components of Trilinos:

1. *Meros* - This is a preconditioning package that provides scalable block preconditioning for problems with coupled simultaneous solution variables. The pressure convection-diffusion preconditioner detailed in this study is implemented in this package. Meros uses the Epetra package for basic linear algebra functions and Thyra as an abstract interface to other Trilinos packages and for composed operations.
2. *Epetra* - This package provides the fundamental construction routines and operations needed for serial and parallel linear algebra libraries. Epetra also facilitates matrix construction on parallel distributed machines. Each processor constructs the subset of matrix rows assigned to it via the static domain decomposition partitioning generated by stand-alone library, CHACO [5], and a local matrix-vector product is defined. Epetra handles all the distributed parallel matrix details (e.g. local indices versus global indices, communication for matrix-vector products, etc.). Once the matrices F, B, $\hat{B}$, and C are defined, a global matrix-vector product is defined using the matrix-vector products for the individual systems. Construction of the preconditioner follows in a similiar fashion.
3. *AztecOO* - This package is a massively parallel iterative solver library for solving sparse linear systems. All of the Krylov methods (i.e. those for solving the Jacobian system, the F, and Schur complement approximation subsystems) are supplied by AztecOO [14].
4. *ML* - This is a multilevel algebraic multgrid preconditioning package. We use this package with AztecOO to solve the F and Schur complement approximation subsystems.
5. *NOX* - This is a package for solving nonlinear systems of equations. We use NOX for the inexact nonlinear Newton solver.

Once all of the matrices and matrix-vector products are defined, we can use Trilinos to solve the incompressible Navier–Stokes equations using our block preconditioner with specific choices of linear solvers for the Jacobian system, the convection–diffusion, and Schur complement approximation subproblems.

For solving the system with coefficient matrix F we use GMRES preconditioned with three levels of algebraic multigrid, and for the pressure Poisson problem, we use CG preconditioned with four levels of algebraic multigrid. For the convection-diffusion problem, a block Gauss Seidel smoother was used, and for the pressure Poisson problem, a multilevel smoother polynomial was used for the smoothing operations [1]. For the coarsest level in the multigrid scheme, a direct LU solve was employed. To solve the linear problem associated with each Newton iteration, we use GMRESR, a variation on GMRES proposed by van der Vorst and Vuik [15] allowing the preconditioner

to vary at each iteration. GMRESR is required because we used a multigrid preconditioned Krylov subspace method to generate approximate solutions in the subsidiary computations (pressure Poisson and convection-diffusion-like) of the preconditioner, so the preconditioner is not a fixed linear operator.

In our experiments, we compare methods from pressure correction and approximate commutator with a one-level Schwarz domain decomposition preconditioner [11]. This preconditioner does not vary from iteration to iteration (as the block preconditioners do), so GMRES can be used as the outer solver. Domain decomposition methods are based upon computing approximate solutions on subdomains. Robustness can be improved by increasing the coupling between processors, thus expanding the original subdomains to include unknowns outside of the processor's assigned nodes. Again, the original Jacobian system matrix is partitioned into subdomains using CHACO, whereas AztecOO is used to implement the one-level Schwarz method and automatically construct the overlapping submatrices. Instead of solving the submatrix systems exactly we use an incomplete factorization technique on each subdomain (processor).

We have tested the fluid methods discussed above using MPSalsa [10], a code that models chemically reactive, incompressible fluids, developed at Sandia National Laboratory. The discretization of the Navier-Stokes equations provided by MPSalsa is a pressure stabilized, streamwise upwinded Petrov Galerkin least squares finite element scheme [13] with Q_1-Q_1 elements. One advantage of equal order interpolants is that the velocity and pressure degrees of freedom are defined at the same grid points, so equal order interpolants for both velocity and pressure are used. For these experiments, we used an ILUT with no fill-in to compare with the approximate commutator methods. Therefore, the ILU factors have the same number of nonzeros as the original matrix with no entries dropped [11].

We have modeled the microfluidic simulations, using Sundance [8], a high-performance code for parallel finite-element solutions of partial differential equations. Sundance uses an autodifferentiation engine, so we can write out the symbolic objects we wish to differentiate in a high lever manner. This gives us flexibility in the formulation and discretization of our model problem.

5. Results. We will begin by giving some preliminary results in MPSalsa to show that this solution approach is valid for an incompressible flow problem. Then, we will show results for an incompressible microfluidic problem.

5.1. Backward Facing Step. We frist apply our method to a two-dimensional flow over a diamond obstruction. We consider a rectangular region with width of unit length and a channel length of seven units, where the fluid flows in one side of a channel, then around the obstruction and out the other end of the channel. Velocities are zero along the top and bottom of the channel, and along the obstruction. The flow is set with a parabolic inflow condition, i.e. $\mathbf{u}_x = 1 - y^2, \mathbf{u}_y = 0$, and a natural outflow condition, i.e. $\frac{\partial u_x}{\partial x} = p$ and $\frac{\partial u_y}{\partial x} = 0$.

For the diamond obstruction problem, we terminate the nonlinear iteration when the relative error in the residual is 10^{-3}, i.e.

$$\left\| \begin{pmatrix} \mathbf{f} - (F\mathbf{u} + B^T p) \\ g - (\hat{B}\mathbf{u} - Cp) \end{pmatrix} \right\| \leq 10^{-3} \left\| \begin{pmatrix} \mathbf{f} \\ g \end{pmatrix} \right\|. \tag{5.1}$$

The tolerance for the solve with the Jacobian system, is fixed at 10^{-3} with zero initial guess. For both problems, we employ inexact solves on the subsidiary pressure Poisson type and convection-diffusion subproblems. For solving the system with coefficient

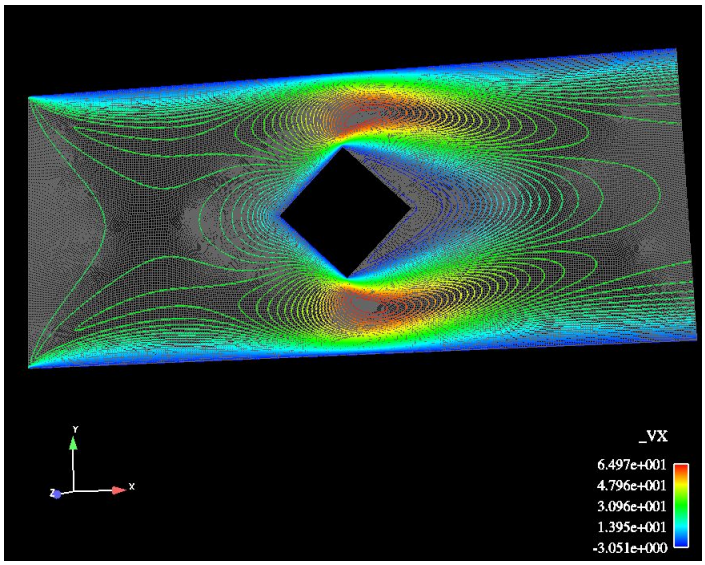


FIG. 5.1. *Model of a flow over a diamond obstruction.*

matrix A_p, we use six iterations of algebraic multigrid preconditioned CG and for the convection-diffusion-like subproblem, with coefficient matrix F, we fix a tolerance of 10^{-2}, i.e. this iteration is terminated when

$$\|(\mathbf{y} - F\mathbf{u})\| \leq 10^{-2}\|\mathbf{y}\|. \tag{5.2}$$

We compare this method to a one-level overlapping Schwarz domain decomposition preconditioner that uses GMRES to solve the Jacobian system at each step using the same tolerances. For both preconditioners, we use a Krylov subspace of 300 and a maximum number of iterations of 3000. The results were obtained in parallel on Sandia's Institutional Computing Cluster (ICC). Each of this cluster's compute nodes are dual Intel 3.6 GHz Xenon processors with 2GB of RAM.

We compare the pressure convection-diffusion preconditioner to the domain decomposition preconditioner on the flow over a diamond obstruction problem generated by MPSalsa. In the first column of Table 5.1, we list the Reynolds number followed by the number of unknowns for four problem sizes in column two. In columns three and four we list the total CPU time and the average number of outer linear iterations per Newton step for the pressure convection-diffusion and domain decomposition, respectively. We see iteration counts that are largely independent of mesh size for a given Reynolds number and an increase in the computational time as the mesh size is refined. The domain decomposition preconditioner does not display mesh independent convergence behavior as the mesh is refined. For Re 10 and Re 25, the pressure convection-diffusion preconditioner was faster in all cases. For Re 40, it was faster for all meshes except for the small problems with 62,000 unknowns run on one processor. Note that the GMRES solver preconditioned with domain decomposition stagnated before a solution was found for the problems with 4 million unknowns. The pressure convection-diffusion preconditioner converged without difficulty on this problem. On modest sized problems where both methods converged, the pressure convection-diffusion preconditioner ranged from 4 to 15 times faster than domain decomposition.

TABLE 5.1
Comparison of the iteration counts and CPU time for the pressure convection-diffusion and domain decomposition preconditioners for the flow over a diamond obstruction. NC stands for no convergence.

Re Number	Unknowns	p Convection-Diffusion		DD One-level		Procs
		iters	time	iters	time	
$Re = 10$	62K	20.5	138.8	110.8	186.6	1
	256K	22.5	266.2	284.6	1657.4	4
	1M	22.9	501.0	1329.0	7825.5	16
	4M	29.4	1841.7	NC	NC	64
$Re = 25$	62K	32.9	248.0	101.7	198.8	1
	256K	35.9	480.6	273.8	1583.1	4
	1M	38.3	956.9	1104.8	7631.5	16
	4M	52.0	4189.8	NC	NC	64
$Re = 40$	62K	54.6	565.8	70.4	267.2	1
	256K	70.1	1280.9	203.9	1420.7	4
	1M	65.4	2011.7	997.1	8188.2	16
	4M	79.8	9387.9	NC	NC	64

5.2. Induced-charge electro-osmosis (ICEO). For the two-dimensional induced-charge electro-osmosis (ICEO) problem, we consider an enclosed region with unit length width and length, with an obstruction in the center of the region. Velocities are zero along the top and bottom of the channel, and along the obstruction is set by solving (2.2). We have used Sundance to discretize the problem and to setup the necessary boundary conditions.

We terminate the nonlinear iteration when the relative error (5.1) is 10^{-3}. The tolerance for the solve with the Jacobian system, is fixed at 10^{-3} with zero initial guess. For both problems, we employ inexact solves on the subsidiary pressure Poisson type and convection-diffusion subproblems. For solving the system with coefficient matrix A_p, we use six iterations of algebraic multigrid preconditioned CG and for the convection-diffusion-like subproblem, with coefficient matrix F, we fix a tolerance of 10^{-2} (5.2). In 5.2, we give some preliminary iteration counts and timings on using the P-CD to solve this microfluidic problem. Further scaling studies and refinements are needed to show the efficacy of this method on this new class of problem.

TABLE 5.2
Iteration counts and CPU time for the pressure convection-diffusion preconditioners for the ICEO problem.

Re Number	Unknowns	Iters	Time
$Re = 10$	50K	20.5	138.8
$Re = 25$	50K	32.9	248.0

6. Conclusions and Future Work. We have shown how new preconditioning strategies based off of the approximation of the Schur complement generate iteration counts and CPU times that are better than other current strategies for solving the incompressible Navier-Stokes equations. We have shown that the iteration counts are independent of the mesh with competitive CPU times to other preconditioning strategies. We also show preliminary work on expanding these preconditioning strategies to

a fixed 'induced-charge electro-osmosis' (ICEO) microfluidic problem. Future work is still needed to show that these methods are truely applicable to microfluidic problems, but the preliminary results are promising.

REFERENCES

[1] M. Adams, Marian Brezina, Jonathan Hu, and R. Tuminaro, *Parallel multigrid smoothing: Polynomial versus Gauss-Seidel*, Journal of Computational Physics, 188 (2003), pp. 593–610.

[2] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, UK, 2005.

[3] H. C. Elman, Victoria E. Howle, John Shadid, Robert Shuttleworth, and Ray Tuminaro, *Block preconditioners based on approximate commutators*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1651–1668.

[4] ———, *A taxonomy and comparison of parallel block multilevel preconditioners for the incompressible Navier–Stokes equations*, tech. report, Sandia National Laboratories, 2006.

[5] B. Hendrickson and R. Leland, *A users guide to Chaco, version 1.0.*, Tech. Report SAND93-2339, Sandia National Laboratories, 1993.

[6] M. A. Heroux, *Trilinos/Petra: linear algebra services package*, Tech. Report SAND2001-1494W, Sandia National Laboratories, 2001.

[7] D. Kay, D. Loghin, and A. J. Wathen, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM Journal on Scientific Computing, 24 (2002), pp. 237–256.

[8] K. Long, *Sundance 2.0 tutorial*, tech. report, Sandia National Laboratories, 2004.

[9] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1969–1972.

[10] J. Shadid, A. Salinger, R. Schmidt, T. Smith, S. Hutchinson, G. Hennigan, K Devine, and H. Moffat., *MPSalsa version 1.5: A finite element computer program for reacting flow problems*, tech. report, Sandia National Laboratories, 1998.

[11] J. Shadid, R. Tuminaro, K. Devine, G. Hennigan, and P. Lin, *Performance of fully-coupled domain decomposition preconditioners for finite element transport/reaction simulations*, Journal of Computational Physics, 205 (2005), pp. 24–47.

[12] D. Silvester, H. Elman, D. Kay, and A. Wathen, *Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow*, J. Comp. Appl. Math., 128 (2001), pp. 261–279.

[13] T. E. Tezduyar, *Stabilized finite element formulations for incompressible flow computations*, Advances in Applied Mechanics, 28 (1991), pp. 1–44.

[14] R. Tuminaro, M. Heroux, S. Hutchinson, and J. Shadid, *Official Aztec user's guide: Version 2.1*, Tech. Report Sand99-8801J, Sandia National Laboratories, Albuquerque NM, 87185, Nov 1999.

[15] H. A. van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*, Numerical Linear Algebra with Applications, 1 (1994), pp. 369–386.

Enabling Computational Nanoscience

The Enabling Computational Nanoscience target area encompasses a wide variety of research spanning several disciplines. All of them share the goal removing obstacles to performing computational nanoscience in the future. The first several papers work towards improving computational design capabilities. Bashir *et al.* in a collaboration between SNL, MIT, and UT Austin, develop a novel method for generating reduced order models which is tailored to accelerating design calculations. Benavides *et al.* presents work on JAVA-based visualization tools for atomistic simulations so that materials experts can query the results of energy minimization calculations. Bichon *et al.* investigates a new approach to reliability assessment that uses global optimization techniques to look at multiple potential points of failure. The work of Crobak and Berry motivates the use of novel computational architectures for running graph algorithms with a diverse set of nanoscale applications. Johnson and Brewer present research into mesh quality optimization methods with a focus on improved scalability to very large meshes, which Parrish *et al.* work on algorithms for conformal mesh refinement. The final set of papers fall in the computer science arena. The work of Frost-Murphy *et al.* explores many issues in the area of reversible computing in support of the successful development of Quantum-dot Cellular Automata (QCA) microarchitectures. Rupnow and Underwood investigate the use of Reconfigurable Functional Units to accelerate a wide range Sandia application codes, including Molecular Dynamics and Electronic Circuit codes. Finally, Vance and Underwood look at the parallelization of massively multithreaded algorithms which will be needed for successful exploitation of nanoscale QCA computers. Well beyond the specific results of these papers, the full impact of these works will come from the propagation of the valuable research experiences represented by these papers into the careers of students who performed the work.

A.G. Salinger

October 30, 2006

REDUCED-ORDER MODEL CONSTRUCTION FOR HIGH-DIMENSIONAL SYSTEMS

O. BASHIR*, K. WILLCOX*, B. VAN BLOEMEN WAANDERS†, J. HILL*, AND O. GHATTAS‡

Abstract. Reduced-order models which are able to approximate full-scale outputs over a wide range of input parameters have an important role to play in making tractable large-scale optimal design, optimal control, and inverse problem applications. We pose the task of determining appropriate training sets for reduced-basis construction as a sequence of optimization problems and show that these problems have a closed-form solution under certain assumptions. We present an efficient model-construction algorithm that scales well to systems with initial conditions, i.e. states, of high dimension.

1. Introduction. Linear, time-invariant (LTI) systems are commonly used to predict the evolution of system state from a certain initial condition in many applications such as circuit simulation, micro- and nanoscale flow calculation, and image processing. As the number of state variables in a system grows, though, the computational cost of doing so increases non-linearly; depending on the application, the cost can become great enough such that solving large-scale problems in reasonable time becomes intractable. Model-order reduction methods have been devised which capture the dominant modes of large LTI systems with few state variables. The resulting reduced-order models can be used to quickly approximate full-order predictions.

Model-order reduction for prediction is straightforward if the set of possible initial conditions is both known and small. Given a single initial condition u_0, a full-scale LTI system can be simulated in time to determine the state $u(t)$. A common method of constructing a reduced-order model is to excite the full system with the initial condition u_0 and take "snapshots" of the state at different time instants. This data can be used to form a basis which transforms the full system into a reduced system, but the resulting reduced system will only be valid for initial conditions similar to u_0.

If the initial condition u_0 is not known *a priori*, construction of a basis for reduction is more difficult. Snapshots of the full solution must then be taken for some *set* of seed initial conditions, not just a single initial condition. For any system which has an initial condition with many degrees of freedom, this process is considerably more expensive than for systems with low-dimensional states. Simple systems with straightforward dynamics lend themselves to intuitive choices for seed initial conditions. In the abscence of this simplicity, though, the common way to construct a robust basis is to use many different combinations of seed initial conditions. For example, consider a small LTI system with only five state variables, each of which can take initial values between 0 and 100. With no understanding of the system dynamics, we might select 10 values between 0 and 100 as possible initial values for each state, and collect data for all possible combinations. This would correspond to a set of 10^5 seed initial conditions or forward simulations. Clearly, for large systems, or even for systems with a larger range of initial values, forming a reduced basis in this manner can be prohibitively expensive due to the curse of dimensionality.

To solve this problem, Grepl [2] introduced the greedy algorithm, which seeks to adaptively build a basis by finding the location in input parameter space where the

*Massachusetts Institute of Technology
†Sandia National Laboratories
‡University of Texas at Austin

error in the reduced model is maximal. This approach requires solution of a series of optimization problems: one per location in parameter space which is used to augment the basis. In this paper, we provide a closed-form solution to the greedy optimization problem which holds under certain reasonable assumptions. This solution eliminates the need for iteration in basis formation and thus dramatically decreases the offline cost of using the greedy approach to construct a reduced model.

Below, we present a method of constructing a basis for reduction which is appropriate for systems with initial conditions that have many degrees of freedom. First, we review reduced-order dynamical systems; we then present the theoretical approach leading to a basis-construction algorithm; finally, we use numerical experiments to demonstrate the utility of the algorithm.

2. Reduced-order Dynamical Systems. Consider the general linear, time-invariant (LTI) dynamical system

$$M\dot{u} + Ku = 0, \tag{2.1}$$

$$g = Gu, \tag{2.2}$$

with initial condition

$$u(0) = u_0, \tag{2.3}$$

where $u(t) \in \mathbb{R}^N$ is the system state, $\dot{u}(t)$ is the derivative of $u(t)$ with respect to time, and the vector u_0 contains the specified initial state. The matrix $G \in \mathbb{R}^{Q\times N}$ defines the Q outputs of interest, which are contained in the output vector $g(t)$. In general, we are interested in systems of the form (2.1) that result from spatial discretization of PDEs. In this case, the dimension of the system, N, is very large and the matrices $M \in \mathbb{R}^{N\times N}$ and $K \in \mathbb{R}^{N\times N}$ result from the chosen spatial discretization method.

A reduced-order model of (2.1)–(2.3) can be derived by assuming that the state $u(t)$ is represented as a linear combination of n basis vectors,

$$\hat{u} = \Phi u_r, \tag{2.4}$$

where $\hat{u}(t)$ is the reduced model approximation of the state $u(t)$ and $n \ll N$. The projection matrix $\Phi \in \mathbb{R}^{N\times n}$ contains as columns the basis vectors ϕ_i, i.e., $\Phi = [\phi_1\ \phi_2\ \cdots\ \phi_n]$, and the vector $u_r(t) \in \mathbb{R}^n$ contains the corresponding modal amplitudes. This yields the reduced-order model with state $u_r(t)$ and output $g_r(t)$

$$M_r\dot{u}_r + K_r u_r = 0, \tag{2.5}$$

$$g_r = G_r u_r, \tag{2.6}$$

$$M_r u_{r_0} = \Phi^T M u_0, \tag{2.7}$$

where $M_r = \Phi^T M\Phi$, $K_r = \Phi^T K\Phi$, $G_r = G\Phi$, and $u_{r_0} = u_r(0)$.

After discretization in time, the general LTI system (2.1) and (2.2) can be written as

$$A\mathbf{u} = Tu_0, \tag{2.8}$$

$$\mathbf{g} = C\mathbf{u}, \tag{2.9}$$

where $\mathbf{u} \in \mathbb{R}^{NT}$ contains the system state $u(t)$ discretized over T time instants. The vector $u_0 \in \mathbb{R}^N$ contains the specified initial state. The matrices $A \in \mathbb{R}^{NT\times NT}$ and

$T \in \mathbb{R}^{NT \times N}$ are functions of M and K (2.1) and result from the particular choice of temporal discretization scheme. The output vector $\mathbf{g} \in \mathbb{R}^{QT}$ contains the Q outputs in $g(t)$ at each of the T time instants, and is defined by the matrix $C \in \mathbb{R}^{QT \times NT}$, which has as block diagonal entries the matrix G in (2.2).

In discrete form, the reduced order model of (2.8) and (2.9) can be written as

$$A_r \mathbf{u}_r = T_r u_0, \tag{2.10}$$

$$\mathbf{g}_r = C_r \mathbf{u}_r, \tag{2.11}$$

where $A_r = V^T A V$, $T_r = V^T T$, and $C_r = CV$. The reduced-order discrete state is $\mathbf{u}_r \in \mathbb{R}^{nT}$ and the basis $V \in \mathbb{R}^{NT \times nT}$ contains T copies of Φ on its block diagonal entries.

3. Hessian-based Model Reduction. In this section, the proposed model reduction methodology is presented. The approach is motivated by the greedy algorithm of Grepl [2], which seeks to adaptively build a basis by finding the location in input parameter space where the error in the reduced model is maximal. The solution at that parameter location is then added to the basis, and the procedure is repeated. For the finite-time-horizon problem considered here, we will show that the optimization problem solved at each iteration of the greedy algorithm has a closed-form solution in the form of an eigenvalue problem.

3.1. Theoretical Approach. In applications such as source inversion, we require a reduced-order model that will provide accurate outputs for any initial condition contained in some set $\mathcal{U}_0$. Using the projection framework described above, the task therefore becomes one of choosing an appropriate basis so that the error between full-order and reduced-order outputs is small for all possible initial conditions. We define an optimal basis, V^*, to be one that minimizes the maximal error between the full-order and reduced-order outputs of the fully discrete system,

$$V^* = \arg\min_{V} \max_{u_0 \in \mathcal{U}_0} \frac{1}{2} \left(\mathbf{g} - \mathbf{g}_r\right)^T \left(\mathbf{g} - \mathbf{g}_r\right) \tag{3.1}$$

$$\begin{aligned} \text{s.t.} \qquad A\mathbf{u} &= T u_0, \\ \mathbf{g} &= C\mathbf{u}, \\ A_r \mathbf{u}_r &= T_r u_0, \\ \mathbf{g}_r &= C_r \mathbf{u}_r. \end{aligned}$$

For this formulation, the only restriction that we will place on the set $\mathcal{U}_0$ is that it contain vectors of unit length. This prevents unboundedness in the optimization problem, since otherwise the error in the reduced system could be made arbitrarily large. Because the system is linear, the basis V^* will still be valid for initial conditions of any finite norm.

A suboptimal but computationally efficient approach to solving (3.1) is inspired by the greedy algorithm of Grepl [2]. The greedy approach in this context is summarized as follows: given a temporary (non-optimal) reduced-order model defined by the matrices A_r, T_r and C_r, we propose to find the initial condition $u_0^* \in \mathcal{U}_0$ that maximizes the error in (3.1); that initial condition will then be used to augment the reduced basis with new snapshot data. These steps are performed at each iteration of the greedy algorithm to adaptively improve the reduced basis. The key step above is

finding the worst-case initial condition $u_0^* \in \mathcal{U}_0$, which is done by solving the modified optimization problem

$$u_0^* = \arg\max_{u_0 \in \mathcal{U}_0} \ \frac{1}{2} \left(\mathbf{g} - \mathbf{g}_r\right)^T \left(\mathbf{g} - \mathbf{g}_r\right) \tag{3.2}$$

$$\begin{aligned} \text{s.t.} \qquad A\mathbf{u} &= Tu_0, \\ \mathbf{g} &= C\mathbf{u}, \\ A_r\mathbf{u}_r &= T_r u_0, \\ \mathbf{g}_r &= C_r\mathbf{u}_r. \end{aligned}$$

After some assumptions and a series of steps beyond the scope of this paper, (3.2) becomes

$$u_0^* = \arg\max_{u_0 \in \mathcal{U}_0} \ \left(u_0^{\perp}\right)^T \frac{1}{2} \left(CA^{-1}T\right)^T \left(CA^{-1}T\right) \left(u_0^{\perp}\right), \tag{3.3}$$

where $u_0^{\perp}$ is the part of the initial condition not spanned by the temporary basis which exists at a given iteration of the greedy algorithm. By inspection it can be seen that, since the only restriction placed on $\mathcal{U}_0$ is that it contain vectors of unit length, the optimal solution to (3.3) is obtained by choosing u_0^* to be the dominant eigenvector of $H_0 = \left(CA^{-1}T\right)^T \left(CA^{-1}T\right)$, which happens to be the Hessian operator associated with (3.3) [1].

This result motivates the following efficient basis-construction approach for the initial condition problem. This method does not solve the optimization problems (3.1)

1. For the Hessian matrix, H_0, find the p eigenvectors $z_1, \ldots z_p$ with largest eigenvalues $\lambda_1, \ldots \lambda_p$.
2. For $i = 1, \ldots, p$, set $u_0 = z_i$ and compute the corresponding solution $\mathbf{u}^i$.
3. Form the reduced basis as the span of the solutions $\mathbf{u}^i$, $i = 1, \ldots, p$.

or (3.2) exactly, but provides a very good approximate solution to (3.2) under the reasonable conditions discussed above. Note that the eigenvectors are all computed from a single Hessian matrix. Algorithm 3.1 can be implemented very efficiently using a sparse eigenvalue solver. The Hessian matrix need not be formed explicitly; rather, the dominant eigenvectors can be computed via a set of solves of the system (2.8), (2.9).

4. Numerical Results. Above, we have provided the theoretical background for an algorithm which should be able to form a reduced model of a large, LTI system which effectively replicates outputs for any initial condition. We now set out to demonstrate the validity of the proposed method. In the process, we hope to gain an understanding of the effects of varying the number of eigenvectors of the Hessian operator used in the algorithm, as well as the effects of varying the size of the reduced basis. The general framework for numerical experiments in this paper is as follows:

1. Form a large LTI system with relatively few outputs and an initial condition with many degrees of freedom, i.e., many state variables.
2. Using Algorithm 3.1, form a reduced model of the full LTI system.
3. Randomly construct a set of initial conditions with which to test the reduced model. Use each of these initial conditions to simulate the full and reduced systems in time.

4. For each test initial condition, compare the full outputs and the reduced outputs to assess the quality of the reduced model.

These experiments were performed using MATLAB. Steps 1 and 2 above are detailed further in the following sections.

4.1. Formation of Large LTI Systems. A finite-element approximation of the convection-diffusion equation in a two-dimensional domain provides a suitable LTI system for experimentation. At $t = 0$, each node can have any real, non-negative concentration, so the initial condition is indeed of high-dimension.

Thus, we generate full systems for the experiments by using a finite-element discretization in a rectangular domain with a uniform and constant velocity field. The velocity is directed in the positive x-direction as defined by Figure 4.1. Below is the convection-diffusion equation along with the boundary condition used in the experiments.

$$\frac{\partial u}{\partial t} + \mathbf{U} \cdot \nabla u - \kappa \nabla^2 u = 0 \text{ in } \Omega \qquad (4.1)$$
$$u = 0 \text{ on } \Gamma_{in},$$

where $\mathbf{U}$ is the velocity vector field, κ is the diffusivity, and Γ_{in} is the inflow boundary, defined by the $x = 0$ segment in Figure 4.1. Homogeneous Neumann boundary conditions were applied on the remaining boundaries. The same spatial discretization, with triangular elements and $N = 1860$ unknown nodal values, was used for all experiments.

The mass matrix M and stiffness matrix K in (2.1) are dependent on the velocity field and Peclet number, which is defined as $Pe = \frac{v_c \ell_c}{\kappa}$. In this paper, the characteristic velocity was taken to be the maximum velocity magnitude in the domain, while the domain length was used as the characteristic length scale. The uniform velocity field decribed above was used in all experiments, but Pe was varied. Peclet numbers of 10, 100, 1000, and 10,000, in order of increasing convective nature, were used to generate different full-scale systems. Streamline Upwind Petrov-Galerkin (SUPG) stabilization was introduced to allow for higher Pe.

The matrix G which relates system states to outputs in (2.2) is dependent on which nodal values or combinations of values are chosen as the Q outputs of interest. Experiments were performed for both $Q = 2$ and $Q = 10$, and in both cases, each output was defined as the concentration at a certain nodal value in the domain.

4.2. Construction of Reduced-Order Models. We rewrite Algorithm 3.1 here in a form specific to the implementaion used in the experiments.

1. For the Hessian matrix, H_0, find the p eigenvectors $z_1, \ldots z_p$ with largest eigenvalues $\lambda_1, \ldots \lambda_p$.
2. For $i = 1, \ldots, p$, set the initial condition $u_0 = z_i$ and compute the corresponding solution $u^i(t)$ by simulating forward in time with timestep Δt from time $t = 0$ to $t = T_f$.
3. Aggregate all N-by-1 solution snapshots from all p forward solutions in a snapshot matrix $X \in \mathbb{R}^{N \times T_n p}$, where T_n is the number of timesteps from $t = 0$ to $t = T_f$.
4. Form the reduced basis as the span of the solutions by using proper orthogonal decomposition (POD): take the n most dominant eigenvectors of $\left(XX^T\right)$ to be the n reduced basis vectors. The basis can then be used to form the reduced model in (2.5), (2.6), and (2.7).

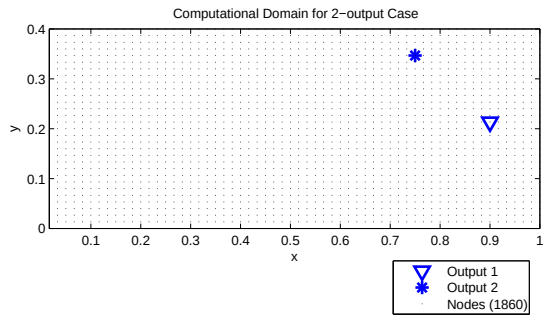


FIG. 4.1. *The computational domain and locations of output nodes for the two-output case.*

For all experiments, the timestep used was $\Delta t = 0.02$ and the time limit, set approximately by the maximum time of convection across the length of the domain, was $T_f = 1.4$. Each forward solution was computed using the implicit Crank-Nicolson method.

The only two parameters other than timestep and time limit that influence reduced model construction are p, the number of eigenvectors of H_0 used as seed initial conditions; and n, the number of reduced basis vectors. Values for p of 5, 10, and 20 were tested.

Unlike the selection of p, the determination of n was automated by choosing $\overline{\lambda} < \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{T_n p} \lambda_j}$. The fraction of the sum of the eigenvalues of the n POD basis vectors included in the reduced basis over the sum of the eigenvalues of all candidate POD basis vectors was bounded from below by $\overline{\lambda}$. Two different values, $\overline{\lambda} = 0.999$ and $\overline{\lambda} = 0.999999$, were chosen for the experiments. Use of the latter was expected to include more POD vectors in the reduced basis in order to meet the aforementioned criteria.

4.3. Results: Two-output Case. The set of results presented in this section was generated for the $Q = 2$ configuration pictured in Figure 4.1. The first output $g_1(t)$ was made to represent the concentration over time at the first output node. The second output follows a similar convention.

Before analyzing the performance of a reduced model construced by the algorithm above, we can examine the eigenvectors z_i of H_0 which serve as the seed initial conditions. Figure 4.2 shows only the dominant eigenvector for four different values of Pe. What we see is rather intuitive: the more convective the flow is, the greater the concentration is upstream of the outputs. The reduced model requires more information from nonzero upstream initial conditions if the flow is convective. If the flow is more diffusive, then initial conditions which are localized at the output nodes dominate the basis formation.

4.3.1. Representative Output Comparison. A reduced system constructed using the algorithm above should provide reduced outputs $g_{r,i}(t)$ which effectively approximate the full outputs $g_i(t)$ for any initial condition. To test this, each reduced

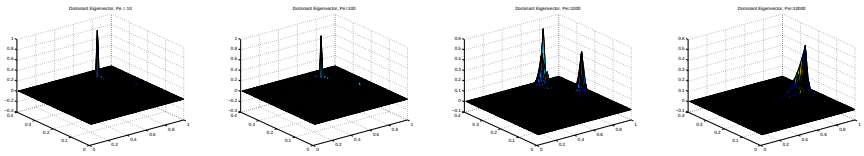


FIG. 4.2. *Dominant eigenvectors of H_0 for different Peclet numbers. Note the smoothing of the eigenvector upstream for more convective flows (higher Pe).*

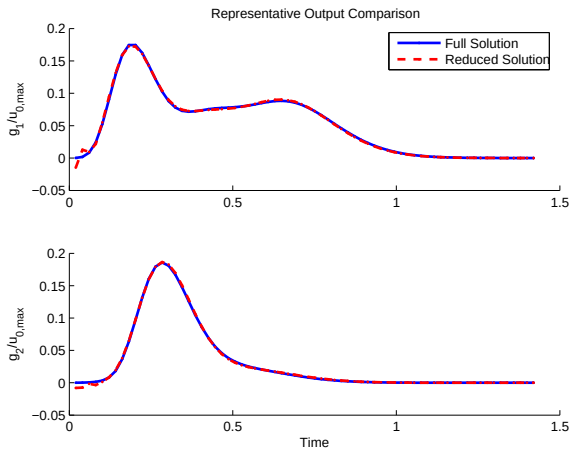


FIG. 4.3. *A typical comparison of full and reduced outputs. The initial condition was formed by superimposing 3 randomly chosen Gaussian distributions. $Pe = 100$, $p = 20$, $\overline{\lambda} = 0.999999$.*

model formed was excited with an initial condition involving a random combination of Gaussian distributions. Figure 4.3 shows a typical comparison of full and reduced outputs generated by these experiments. This result demonstrates that a reduced model of size $n = 118$ formed using Algorithm 3.1 can effectively replicate the outputs of this particular full-scale system of size $N = 1860$.

4.3.2. Effect of Variations in $\overline{\lambda}$. As discussed above, $\overline{\lambda}$ is the parameter which controls the number of POD vectors chosen for inclusion in the reduced basis. If it is too small, the reduced basis might not span the space of all initial conditions for which it is desired that the reduced model be valid. Figure 4.4 illustrates the effect of changing $\overline{\lambda}$. The curve corresponding to a greater value of $\overline{\lambda}$ shows a clear improvement over the other result. The pointwise sum of $|\overline{g}(t) - \overline{g}_r(t)|$, a measurement of the modeling error, is equal to 1.1507 for $\overline{\lambda} = 0.999$ and 0.1338 for $\overline{\lambda} = 0.999999$. This improvement comes with a price, though: the number of basis vectors, and therefore the size of the reduced model n, increases from 72 to 154.

4.3.3. Effect of Variations in p. Another way to alter the size and quality of the reduced model is to change p, the number of eigenvectors of H_0 which are used as seed initial conditions for basis creation. The effect of doing so is illustrated Figure 4.5. An increase in reduced model quality clearly accompanies an increase in p. For $p = 5$, the pointwise sum of the modeling error is 2.3968, while for $p = 20$ it is only 0.2348. The size of the reduced model n grows from 44 to 82 to 129 for p values of 5, 10, and 20 respectively. This is expected, since greater p implies more snapshot

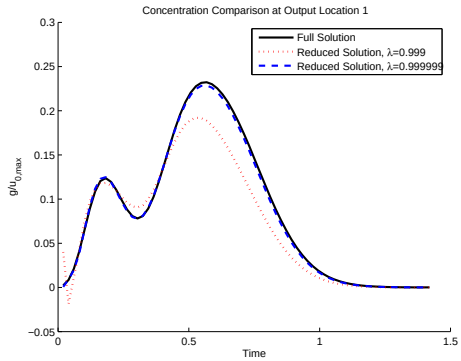


FIG. 4.4. *The plot above shows a comparison between full and reduced solutions at a certain location for two different values of* $\overline{\lambda}$. *The initial condition was formed by superimposing 7 randomly chosen Gaussian distributions.* $Pe = 100$, $p = 10$.

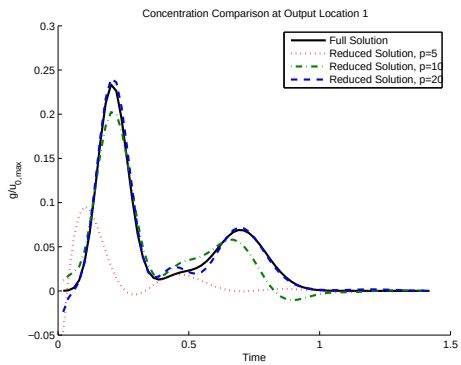


FIG. 4.5. *The above plot illustrates that increasing the number of Hessian eigenvector initial conditions* p *used in basis formation leads to more accurate reduced-order output. The test initial condition for this result was formed by superimposing 3 randomly chosen Gaussian distributions.* $Pe = 1000$, $\overline{\lambda} = 0.999$.

data with which to build the reduced basis, effectively uncovering more full system modes and decreasing the relative importance of the most dominant POD vectors (eigenvectors of XX^T). In general, for the same value of $\overline{\lambda}$, more POD vectors are included in the basis if p is larger.

Figure 4.6 shows a case which is difficult for the reduced order model to treat. Even though p and $\overline{\lambda}$ are relatively high in the first trial, the reduced model does not seem to accurately replicate the full output at the second output location. This is because the test initial condition for these trials was a single Gaussian distribution unlike any of the p seed eigenvectors. Despite this, the first plot ($p = 20$) shows that the maximum modeling error is only 2% of the maximum magnitude of the initial condition. Increasing p dramatically to 50 shows, in turn, a dramatic improvement in the reduced order output: in fact, the model even captures the small perturbation in the full output which, at its peak, has a magnitude of just 2×10^{-4} times the maximum magnitude of the initial condition. Whether increasing p so sharply is necessary depends on application demands, since doing so increases the size of the

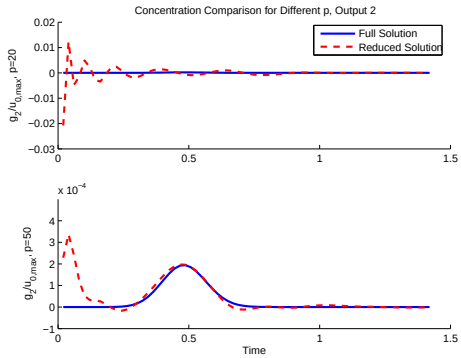


FIG. 4.6. *Another example of increased p leading to more accurate reduced-order output, even for initial conditions difficult for the reduced-order model to treat - in this case, a single Gaussian distribution centered far from the output locations.* $Pe = 1000$, $\overline{\lambda} = 0.999999$.

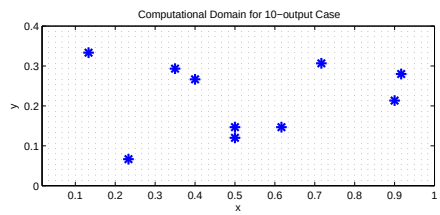


FIG. 4.7. *The computational domain and locations of output nodes for the ten-output case.*

reduced system n from 275 to 524.

4.4. Results: Ten-output Case. To understand how the proposed method scales with the number of outputs in the full LTI system, we repeat the experiments for systems with $Q = 10$ outputs. Figure 4.7 shows the locations of the nodes which correspond to each output. The locations were all randomly generated, except for one, which was intentionally placed at the same location as the first output node in the two-output case.

4.4.1. Representative Output Comparison. Figure 4.8 illustrates that the reduced basis formation method is effective for ten outputs as well. With a reduced model of size $n = 168$, all ten outputs are replicated closely. The reduced basis for the same configuration in the two-output case was of size $n = 118$, so the five-fold increase in number of outputs is not reflected in basis size.

The only noticeable case of modeling error can be seen in the fifth comparison plot, in which the $t = 0$ value of $g_5/u_{0,max}$ is close to 0.04 according to the reduced model, but only 0.01 as computed by the full model. Although the reduced model predicts a concentration four times too great, the error associated with this overprediction is only 3% of the maximum value of the initial condition in magnitude.

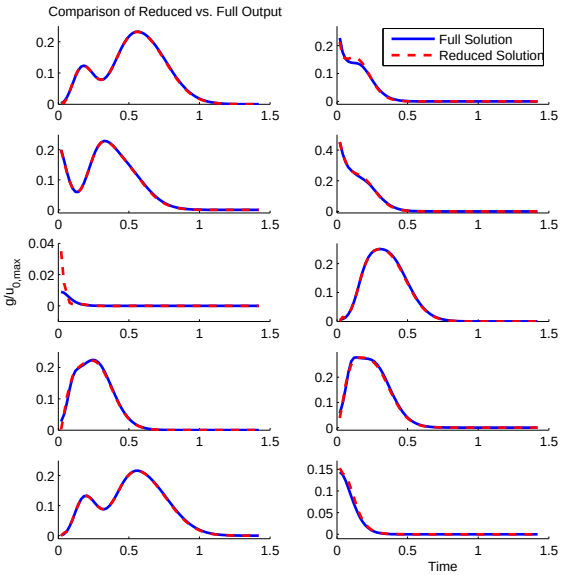


FIG. 4.8. *A comparison of all ten full and reduced outputs. The initial condition used to generate this data was created by superimposing 7 random Gaussian distributions.* $Pe = 100$, $p = 20$, $\overline{\lambda} = 0.999$.

TABLE 4.1
Comparison of modeling error for the same output node, two-output model versus ten-output model. The initial conditions were generated with 1, 3, and 7 random Gaussian distributions.

	$Q = 2$	$Q = 10$
1-Gaussian	0.4483	0.07519
3-Gaussian	0.2348	0.1254
7-Gaussian	0.1626	1.355

4.4.2. Comparison to Two-output Case. The first outputs $g_1(t)$ in both the two-output case and ten-output case were intentionally chosen to correspond to the same node in the computational domain. We can exploit this choice to examine the change in quality of the reduced approximation of $g_1(t)$ due solely to the inclusion of other outputs. To do this, we compare reduced models of two systems with identical physical properties (same M and K, but different G) built with the same values of p and $\overline{\lambda}$.

Table 4.1 shows the modeling error for two similar reduced models ($p = 20$, $\overline{\lambda} = 0.999$) of the same system ($Pe = 1000$), ignoring the difference in the G matrix in (2.2) related to the number of outputs Q. Each row of the table corresponds to a different randomly-generated initial condition. As seen above, the size of the reduced model $n = 224$ for the ten-output case is larger than that of the two-output model, $n = 129$. This is expected since the span of the reduced basis must be larger to capture the behavior at the increased number of output nodes. Interestingly, though, the addition of outputs increases the error for the 7-Gaussian initial condition but has the opposite effect for the other two test initial conditions. This suggests no

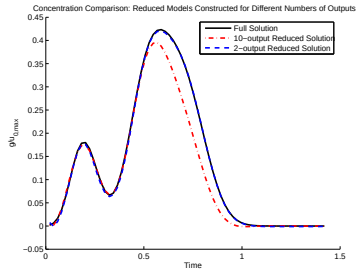


FIG. 4.9. *Certain initial conditions, such as that used to generate this data, are handled better by the two-output reduced model than by the ten-output reduced model.* $Pe = 1000$, $p = 20$, $\overline{\lambda} = 0.999$.

direct relationship between Q and the modeling accuracy for a given output: adding additional output nodes to the full system might actually improve accuracy for the previously existing outputs. The 7-Gaussian case, for which the error does indeed increase when more outputs are introduced, is shown in Figure 4.9.

4.5. Summary and Recommendations. The results above demonstrate that reduced models formed by the proposed method are indeed effective in replicating full-scale outputs. At this point, we can use the results to make recommendations about choosing p and $\overline{\lambda}$, the two parameters which control reduced-model construction.

In practice, one would like to choose these parameters such that both the reduced model size n and the modeling error for a variety of test initial conditions are minimal. The size of the reduced model is important because n is directly related to the online computational cost; that is, n determines the time needed to compute the reduced output approximations, which should be minimal for real-time applications. However, the offline cost of building the reduced model cannot be ignored.

Let us first consider $\overline{\lambda}$, the parameter which directly controls how many basis vectors n are chosen to represent the reduced basis. For implementations in which all POD basis vectors are calculated at once, as in MATLAB via singular value decomposition (SVD), there is no additional offline cost associated with increasing the value of $\overline{\lambda}$. The online cost, though, scales with n as mentioned above. Thus, increasing $\overline{\lambda}$ might appreciably improve modeling accuracy, but doing so can only increase the time needed to compute reduced output approximations.

Changes in p affect the offline cost more strongly. Each additional eigenvector of H_0 adds the cost of one additional forward solution of the full model to generate snapshot data. In addition, in some implementations, computing p eigenvectors of H_0 is more expensive than computing $p-1$ eigenvectors. Finally, the snapshot matrix X grows by T_n if p is incremented by one: performing POD becomes more expensive. If these increases in offline cost can be tolerated, though, the results suggest a clear improvement in reduced-model accuracy for a relatively small increase in online cost.

Figure 4.10 illustrates the details presented in the previous two paragraphs. For a single full LTI system with 10 outputs, six different reduced models were constructed with different combinations of p and $\overline{\lambda}$. Consider one of the three plots, which corresponds to a single test initial condition. The sum, over all 10 outputs, of the cumulative pointwise modeling error is plotted versus reduced-model size n. Ideally, a reduced model should have both small error and small n, so we examine those models whose points reside closest to the origin. Ignoring differences in offline model construction cost, increasing p should be favored over increasing $\overline{\lambda}$ if more accuracy is

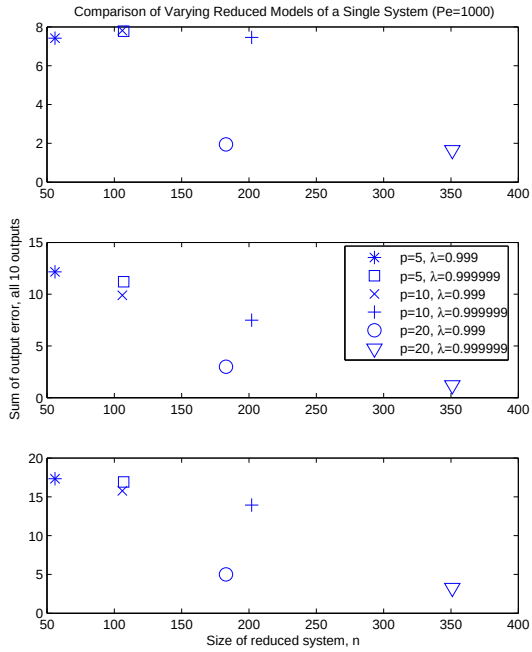


FIG. 4.10. *A measure of the error in six different reduced models of the same system plotted versus their sizes n. The initial conditions used to generate the data for each of the three plots were formed with 1, 3, and 7 random Gaussian distributions. The error shown is the sum, over all ten outputs, of the cumulative pointwise difference between the full and reduced models.* $Pe = 10000$*, 10 outputs.*

desired. This conclusion is reached by realizing that for a comparable level of error, reduced models constructed with larger p are much smaller.

In the case that forward solves of the full system are expensive, making basis construction with large p undesirable, there should be at least as many H_0 eigenvector initial conditions as the number of outputs Q. Figure 4.11 shows the H_0 eigenvalue spectra for the two-output and the ten-output cases. If we assume that the magnitude of each eigenvalue is related to the importance of including the corresponding eigenvector as a seed initial condition, then the sharp decrease near Q in each case supports the assertion that Q serves as a lower bound for p. The value of p should, in practice, be set considerably higher than Q to provide desirable results.

5. Conclusion. In this paper, we proposed a new method for constructing reduced-order models of LTI systems with initial conditions of high dimension. The models are intended to provide effective replication of full-scale, time-dependent outputs for any possible initial condition. The algorithm proposed leads to the construction of a reduced basis, which alone defines the reduced model via projection of the full system.

The method involves using eigenvectors of the Hessian matrix to directly solve an optimization problem inspired by the greedy algorithm. Each of these eigenvectors is used as a seed initial condition to train the reduced system. Above, we demonstrated the utility of the method by performing a series of experiments on a 2-D finite-element

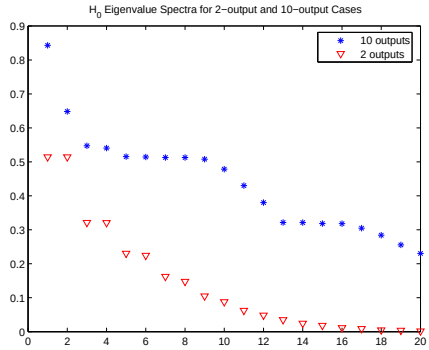


FIG. 4.11. *A comparison of the eigenvalue spectra of H_0 for the two- and ten-output cases.*

system; recommendations were made regarding the proper quantity of Hessian eigenvectors and POD basis vectors to obtain. Given its success, the method can be used to produce real-time output approximation if a reduced model is constructed beforehand.

Future work might involve a more rigorous means of finding the proper number of eigenvectors of the Hessian and basis vectors to use in the method. The method might also be tested on different types of LTI systems. Furthermore, a similar method might be applied to forced-response problems, in which an LTI system accepts inputs of high dimension. Finally, the proposed algorithm could be even more powerful if extended to large-scale, nonlinear systems.

REFERENCES

[1] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders. *Frontiers of Parallel Computing*, chapter Parallel algorithms for PDE-constrained optimization. SIAM, 2006.

[2] M.A. Grepl. *Reduced-basis approximation a posteriori error estimation for parabolic partial differential equations.* PhD thesis, Massachusetts Institute of Technology, 2005.

JAVA-BASED ATOMISTIC VISUALIZATION TOOLS

NICK BENAVIDES*, CYNTHIA A. PHILLIPS†, AND JEAN-PAUL WATSON†

Abstract. We consider visualization tools for comparing two conformations of the same small set of atoms. We motivate the application and document a set of useful tools and references for researchers wishing to learn Java and apply it to this application.

1. Introduction. In this paper we consider tools for visualizing differences between two conformations of the same set of atoms. Such tools will help researchers understand the space of conformations with nearly-minimum energy and help researchers understand why some conformations form local minima.

Given a placement of atoms in three dimensional space one can calculate the *Lennard-Jones* potential between each pair of atoms [10]. We define the energy of a configuration to be the sum of the pairwise Lennard-Jones potentials, taken over all pairs of atoms. We assume that in nature, a stable configuration will be a local minimum for this potential energy and it will be approximately optimal globally.

Researchers wish to better understand the nature of these local minima as well as the relationships between them. It may be difficult for a configuration to "escape" from a local minimum if there is a strong energy barrier. It may be almost impossible to transform directly from one local minimum to another. However, it can be possible to go from one minimum to another through a series of small steps, some of which will increase the energy. Specifically, there is a *path* of configurations $C_1, C_2, \ldots, C_k$, where configuration C_i can transform to configuration C_{i+1} reasonably easily. Frequently this consists of a series of small changes in energy to enable a configuration change with a large drop in potential energy.

A tool for visualizing differences between two configurations could help researchers explore the space of configurations with locally minimum, near-globally minimum potential energy. In this case, one could use pairwise comparisons as a starting point for the more complex comparisions among elements of the entire set. A pairwise comparison tool could be especially helpful for *path analysis*, studying the steps of small changes necessary to move from one local minimum to another. In this case there are specific configurations with clear pairwise relationships that one must understand. Even small problems with 20–40 identical atoms are relevant scientifically and potentially small enough for a human, with proper assistance, to visualize the whole ensemble.

We selected Java programming environment for the visualization tool. Java is a modern object-oriented language with powerful libraries to support graphics. Furthermore, we would like to interface this tool with the DAKOTA optimization toolkit (to obtain the configurations to visualize). Since DAKOTA uses Java, it is a natural choice. Looking ahead to other uses of the visualization technology, there are other optimization codes within the DOE complex and for exteral customers that will require a Java interface.

Learning Java does require some start up time. This paper is a guide to Java-based tools that might be helpful for this particular atom visualization project (how to visualize) and a guide to possible visualization methods (what to visualize).

*Albuquerque Academy & Santa Clara University
†Sandia National Laboratories

2. Representing Conformational Differences. There is considerable room in this application for artistic creativity. Tufte gives a number of compelling examples where showing people data in the right way can make critical data characteristics obvious [7–9]. These books are worth skimming for possible ideas.

A first step to visualizing differences, especially for one not familiar with Java, does not require complex spherical clusters or animiation. One can simply draw a matrix.

Consider the 2D matrix where each column and row represents an atom. Each matrix entry (i, j) is the Lennard-Jones potential between atom i and atom j. Each conformation has such a matrix. Of course, one need only consider half the matrix above the diagonal, since it is symmetric and presumably self-comparisons are not relevant. Given matrices from two configurations M_1 and M_2, we can compare the matrices via the difference matrix $M_d = M_1 - M_2$. We can display the matrix color coding each entry by magnitude and sign. Gray scale is a reasonable starting point.

3. Java-based tools and references. This section gives some guidance on references useful for getting started with Java for this application. We then describe some built-in Java types and tools relavant to the project. We conclude with some code with useful pieces that we hope will serve as a starting point for the next phase of development.

The the first several chapters of the book *Java In A Nutshell* [3] provides an overview of Java syntax syntax and how to compile and run a Java program. *Java 2D Graphics* [4] has a section on using grayscale and other color themes. Ultimately, one will likely need 3D graphics for this application. Chapter 14 of *Killer Game Programming in Java* [2] has a nice introduction to Java 3D. The single most useful book we found on Java 3D is *Java 3D Programming* [6].

Eventually the visualization tool will need a GUI (graphical user interface tool). Swing is a GUI toolkit for Java, capable of handling large software projects. A good reference for this tool is the book *Java Swing* [5]. We found no code fragments in this book useful for our immediate needs, but we expect a friendly user interface for experimenting with visualizations will become increasingly important as our tools mature.

Here are some tips for controlling color, particularly gray scale for a first development [4]. The `java.awt.color.ColorSpace` class encapsulates a given color space. A particularly useful color space type is;

```
public static final int TYPE_GRAY
```

According to Knudsen: "This constant represents a grayscale color space type. This color space type defines colors that are evenly spaced between black and white." There are other built-in color spaces, but gray scale is a simple first-pass method and it may even be completely sufficient for some tasks.

Any future 3D system may be able to use the Java Molecular Viewer (JMV) [1], developed by the Theoretical Biophysics group at the University of Illinois and the Beckman Institute. This code is freely available. Users can customize the way molecules appear. It's possible a comparison system could build upon this foundation. We were unable to run the latest version of JMV (0.85) through a browser on a solaris machine. The browser hung/froze. This problem occurred with mozilla and netscape. This could be a problem with the browser configuration. It's possible the standalone application would work better or perhaps one could use a browser on another platform.

Finally, here are two works of code with detailed examples. They are archived

here to provide a single start-up source for the next developer on this project. These are examples collected from various sections of [6] most relvant to the project.

```
/**
 * Render a 3D shape using a 3D rendering engine
 * that was written from scratch using AWT for
 * graphics operations.
*/

public class MyJava3D extends JFrame
{
 private static int    m_kWidth = 400;
 private static int    m_kHeight = 400;

 private RenderingEngine   renderingEngine = new AwtRenderingEngine();
 private GeometryUpdater   geometryUpdater = new RotatingGeometryUpdater();
 private RemderingSurface   renderingSurface;

 public MyJava3D( )
 {
     // load the object file
  Scene scene = null;
  Shape3D shape = null;

  // read in the geometry information from the data file
  ObjectFile objFileloader = new ObjectFile( ObjectFuke.RESIZE );

  try
  {
   scene = objFileloader.load( "hand1.obj" );
  }
  catch ( Exception e )
  {
   scene = null;
   System.err.println( e );
  }

  if( scene == null )
   System.exit( 1 );

  // retrieve the Shape3D object from the scene
  BranchGroup branchGroup = scene.getSceneGroup( );
  shape = (Shape3D) branchGroup.getChild( 0 );

   // add the geometry to the rendering engine...
   renderingEngine.addGeometry( (GeometryArray) shape.getGeometry() );

   // create a rendering surface and bind the rendering engine
   renderingSurface = new RenderingSurface( renderingEngine,
geometryUpdater );

   // start the rendering surface and add it to the content panel
   renderingSurface.start();
   getContentPane().add( renderingSurface );
```

```
   // disable automatic close support for Swing frame.
  setDefaultCloseOperation( WindowConstants.DO_NOTHING_ON_CLOSE );

  // add the window listener
  addWindowListener(

   new WindowAdapter()
   {
    // handle the system exit window message
    public void windowClosing( WindowEvent e )
    {
    System.exit( 0 );
                  }
   }
  };
 }
 public static void main( String[] args )
 {
  MyJava3D myJava3D = new MyJava3D();
           myJava3D.setTitle( "MyJava3D" );
           myJava3D.setSize( 300, 300 );
           myJava3D.setVisible( true );
 }
}

import java.applet.Applet;

import javax.media.j3d.*;
import javax.vecmath.*;

import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.image.TestureLoader;

/*
 * This example builds a simple Java 3D application using the
 * Sun utility classes: MainFrame and SimpleUniverse.
 * The example displays a moving sphere, in front of a
 * background image. It uses a texture image and one light
 * to increase the visual impact of the scene.
 */
public class SimpleTest extends Applet
{
 /*
  * Create a simple Java 3D environment containing:
  * a sphere (geometry), a light, background geometry
  * with an applied texture, and a behavior that will
  * move the sphere along the X-axis.
  */
 public SimpleTest()
 {
  // create the SimpleUniverse class that will
  // encapsulate the scene that we are building.
```

```
 // SimpleUniverse is a helper class (utility)
 // from SUN that is included with the core Java 3D
 // distribution.
 SimpleUniverse u = new SimpleUniverse();

 // create a BranchGroup. A BranchGroup is a node in
 // a Tree data structure that can have child nodes
 BranchGroup bgRoot = new BranchGroup();

 // create the Background node and add it to the SimpleUniverse
 u.addBranchGraph( createBackground() );

 // create the behaviors to move the geometry along the X-axis.
 // The behavior is added as a child of the bgRoot node.
 // Anything added as a child of the tg node will be affected by the
 // behavior (will move along the X-axis.)
 TransformGroup tg = createBehaviors( bgRoot );

 // add the Sphere geometry as a child of the tg
 // so that it will be moved along the X-axis.
 tg.addChild( createSceneGraph() );

 // because the sphere was added at the 0,0,0 coordinate
 // and by default the viewer is also located at 0,0,0
 // we have to move the viewer back a little so that
 // he/she can see the scene.
 u.getViewingPlatform().setNominalViewingTransform();

 // add a light to the root BranchGroup to illuminate the scene
 addLights( bgRoot );

 // finally wire everything together by adding the root
 // BranchGroup to the SimpleUniverse
 u.addBranchGraph( bgRoot );
}

/*
 * Create the geometry for the scene. In this case
 * we simply create a Sphere
 * (a built-in Java 3D primitive).
 */
public BranchGroup createSceneGraph()
{
 // create a parent BranchGroup node for the Sphere
 BranchGroup bg = new BranchGroup();

 // create an Appearance for the Sphere.
 // The Appearance object controls various rendering
 // options for the Sphere geometry.
 Appearance app = new Appearance();

 // assign a Material to the Apperance. For the Sphere
 // to respond to the light in the scene it must have a Material.
 // Assign some colors to the Material and a shininess setting
```

```
 // that controls how reflective the surface is to lighting.
 Color3f onjColor = new Color3f(0.8f, 0.2f, 1.0f);
 Color3f black = newColor3f(0.0f, 0.0f, 0.0f);
 app.setMaterial(new Material(objColor, black, objColor, black, 80.0f));

 // create a Sphere with a radius of 0.1
 // and associate the Appearance that we described.
 // the option GENERATE_NORMALS is required to ensure that the
 // Sphere responds correctly to lighting.
 Sphere sphere = new Sphere( 0.1f, Primitive.GENERATE_NORMALS, app );

 // add the sphere to the BranchGroup to wire
 // it to the scene.
 bg.addChild( sphere );
 return bg;
}

/*
 * Add a direction light to the BranchGroup.
 */
public void addLights( BranchGroup bg )
{
 // create the color for the light
 Color3f color = new Color3f( 1.0f,1.0f,0.0f );

 // create a vector that describes the direction that
 // the light is shining.
 Vector3f direction = new Vector3f( -1.0f,-1.0f,-1.0f );

 // create the directional light with the color and direction
 DirectionalLight light = new DirectionalLight( color, direction );

 // set the volume of influence of the light.
 // Only objects within the Influencing Bounds
 // will be illuminated.
 light.setInfluencingBounds( getBoundingSphere() );

 // add the light to the BranchGroup
 bg.addChild( light );
}

/*
 * Create some Background geometry to use as
 * a backdrop for the application. Here we create
 * a Sphere that will enclose the entire scene and
 * apply a texture image onto the inside of the Sphere
 * to serve as a graphical backdrop for the scene.
 */
public BranchGroup createBackground()
{
 // create a parent BranchGroup for the Background
 BranchGroup backgroundGroup = new BranchGroup();

 // create a new Background node
```

```
 Background back = new Background();

 // set the range of influence of the background
 back.setApplicationBounds( getBoundingSphere() );

 // create a BranchGroup that will hold
 // our Sphere geometry
 BranchGroup bgGeometry = new BranchGroup();

 // create an appearance for the Sphere
 Appearance app = new Apperance();

 // load a texture image using the Java 3D texture loader
 Texture tex = new TextureLoader( "back.jpg", this).getTexture();

 // apply tge texture to the Appearance
 app.setTexture( tex );

 // create the Sphere geometry with radius 1.0.
 // we tell the Sphere to generate texture coordinates
 // to enable the texture image to be rendered
 // and because we are *inside* the Sphere we have to generate
 // Normal coordinates inwards or the Sphere will not be visible.
 Sphere sphere = new Sphere( 1.0f,
         Primitive.GENERATE_TEXTURE_COORDS |
             Primitive.GENERATE_NORMALS_INWARD, app );

 // start wiring everything together,
 // add the Sphere to its parent BranchGroup.
 bgGeometry.addChild( sphere );

 // assign the BranchGroup to the Background as geometry.
 back.setGeometry( bgGeometry );

 // add the Background node to its parent BranchGroup.
 backgroundGroup.addChild( back );

 return backgroundGroup;
}

/*
 * Create a behavior to move child nodes along the X-axis.
 * The behavior is added to the BranchGroup bg, whereas
 * any nodes added to the returned TransformGroup will be
 * effected by the behavior.
 */
public TransformGroup createBehaviors( BranchGroup bg )
{
 // create a TransformGroup.
 //
 // A TransformGroup is a Group node (can have children)
 // and contains a Transform3D member.
 //
 // The Transform3D member contains a 4x4 transformation matrix
```

```
  // that is applied during rendering to all the TransformGroup's
  // child nodes. The 4x4 matrix can describe:
  // scaling, translation and rotation in one neat package!

  // enable the TRANSFORM_WRITE capability so that
  // our behavior code can modify it at runtime.
  TransformGroup objTrans = new TransformGroup();
  objTrans.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);

  // create a new Transform3D that will describe
  // the direction we want to move.
  Transform3D xAxis = new Transform3D();

  // create an Alpha object.
  // The Alpha object describes a function against time.
  // The Alpha will output a value that ranges between 0 and 1
  // using the time parameters (in milliseconds.)
  Alpha xAlpha = new Alpha( -1,
             Alpha.DECREASING_ENABLE |
             Alpha.INCREASING_ENABLE,
             1000,
             1000,
             5000,
             1000,
             1000,
             10000,
             2000,
             4000 );

  // create a PositionInterpolator.
  // The PositionInterpolator will modify the translation components
  // of a TransformGroup's Transform3D (objTrans) based on the output
  // from the Alpha. In this case the movement will range from
  // -0.8 along the X-axis with Alpha=0 to X=0.8 when Alpha=1.
  PositionInterpolator posInt = new PositionInterpolator( xAlpha,
objTrans, xAxis, -0.8f, 0.8f );

  // set the range of influence of the PositionInterpolator
  posInt.setSchedulingBounds( get BoundingSphere() );

  // wire the PositionInterpolator into its parent
  // TransformGroup. Just like rendering nodes behaviors
  // must be added to the scenegraph.
  objTrans.addChild( posInt );

  // add the TransformGroup to its parent BranchGroup
  bg. addChild( objTrans );

  // we return the TransformGroup with the
  // behavior attatched so that we can add nodes to it
  // (which will be affected by the PositionInterpolator.)
  return objTrans;
 }
```

```
 /*
  * Return a BoundingSphere that describes the
  * volume of the scene.
  */
 BoundingSphere getBoundingSphere()
 {
  return new BoundingSphere( new Point3d(0.0,0.0,0.0), 200.0 );
 }

 /*
  * main entry point for the Application.
  */
 public static void main(String[] args)
 {
  SimpleTest simpleTest = new SimpleTest();
 }
}
```

REFERENCES

[1] M. Bach, R. Brunner, J. Stone, and K. Vandivort, *Jmv*. Current user guide available at `http://www.ks.uiuc.edu/Research/jmv/current/doc/ug/ug.html`.

[2] Andrew Davison, *Killer Game Programming in Java*, O'Reilly Media, Inc, Sebastopol, CA, 2005.

[3] David Flanagan, *Java in a Nutshell*, O'Reilly Media, Inc, Sebastopol, CA, 2005.

[4] Jonathan Knudsen, *Java 2D Graphics*, O'Reilly Media, Inc, Sebastopol, CA, 1999.

[5] Mark Loy, Robert Eckstein, Dave Wood, James Elliot, and Brian Cole, *Java Swing*, O'Reilly Media, Inc, Sebastopol, CA, 2003.

[6] Daniel Selman, *Java 3D Programming*, O'Reilly Media, Inc, Sebastopol, CA, 2002.

[7] Edward Tufte, *Envisioning Information*, Graphics Press, Chelshire, CT, 1990.

[8] ———, *Visual Explanations*, Graphics Press, Chelshire, CT, 1997.

[9] ———, *The Visual Display of Quantitative Information*, Graphics Press, Chelshire, CT, 2001.

[10] David Wales, *Energy Landscapes with Applications to Clusters, Biomolecules, and Glasses*, Cambridge University Press, 2003.

MULTIMODAL RELIABILITY ASSESSMENT FOR COMPLEX ENGINEERING APPLICATIONS USING SEQUENTIAL KRIGING OPTIMIZATION

B.J. BICHON*, M.S. ELDRED†, AND L.P. SWILER†

Abstract. As engineering applications become increasingly complex, they are often characterized by implicit response functions that are both expensive to evaluate and nonlinear in their behavior. Reliability assessment given this type of response is difficult with available methods. Current reliability methods focus on the discovery of a single most probable point of failure, and then build a low-order approximation to the limit state at this point. This creates inaccuracies when applied to engineering applications for which the limit state has a higher degree of nonlinearity or is multimodal. This paper describes the application of sequential kriging optimization to reliability assessment through an efficient global search for multiple most probable points. By locating multiple most probable points of failure, more complex limit states can be modeled, leading to more accurate probability integration. Several possible formulations of this method will be explored and applied to a collection of example problems that currently available methods have difficulty solving either accurately or efficiently.

1. Introduction. Accurate reliability assessment is a problem of great importance to the engineering community. Poor solutions lead to designs that are either unreliable or overly expensive. However, the ability to accurately quantify the uncertainty in a design becomes increasingly difficult as the analysis of the design becomes more expensive and its behavior more nonlinear.

Current methods of reliability assessment solve an optimization problem to locate the most probable point of failure (MPP), and then quantify the reliability based on its location and an approximation to the shape of the limit state at this point. Typically, gradient-based solvers are used to solve this optimization problem, which may converge to sub-optimal solutions for response functions that possess multiple local optima. Convergence to sub-optimal MPPs and limit state approximations that may be inaccurate, make MPP search methods unreliable in practice. Engineers are then forced to revert to sampling methods, which are impractical when evaluation of the response function is expensive.

A reliability assessment method that is both efficient when applied to expensive response functions and accurate when the response function is highly nonlinear is needed. This paper investigates the application of a global optimization tool known as sequential kriging optimization to the search for multiple MPPs. By locating multiple points on the limit state, more accurate approximations to nonlinear limit states can be built, resulting in a more accurate assessment of the reliability.

Sequential kriging optimization [12] (SKO) is an adaptation of efficient global optimization [13] (EGO), which was developed to facilitate the optimization of expensive implicit response functions. These methods build an initial kriging model as a global surrogate for the response function, then intelligently select additional samples to be added for the next kriging model. The new samples are selected based on how much they are expected to improve the current best solution to the optimization problem. When this expected improvement is acceptably small, the optimal solution has been found. Because SKO is a global optimization method, it is able to locate multiple local optima, which will allow for the discovery of multiple MPPs.

*Vanderbilt University
†Sandia National Laboratories

Section 2 describes the reliability assessment problem and traditional methods of solving it. Section 3 gives an overview of SKO and outlines some ideas on how it might be modified for application to MPP search. Section 4 details a preliminary investigation into some of the ideas posed in Section 3. Finally, Section 5 provides concluding remarks and some initial impressions on the promise of this method.

2. Reliability Assessment. The goal of reliability assessment is to determine the probability that an engineered device, component, system, etc. will fail in service given that its behavior is dependent in part on random inputs. This behavior is defined by a response function $g(\mathbf{x})$, where $\mathbf{x}$ represents the vector of random variables defined by known probability distributions. Failure is then defined by that response function exceeding (or failing to exceed) some threshold value $\overline{z}$. The probability of failure, p_f, is then be defined by

$$p_f = \int_{g > \overline{z}} \cdots \int f_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x} \tag{2.1}$$

where $f_{\mathbf{x}}$ is the joint probability density function of the random variables $\mathbf{x}$, and the integration is performed over the failure region where $g > \overline{z}$. In general, $f_{\mathbf{x}}$ is impossible to obtain, and even if it is available, evaluating the multiple integral is impractical. [9] Because of these complications, methods of approximating this integral are used in practice.

2.1. MPP Search Methods. These methods involve solving a nonlinear optimization problem to locate the point on the limit state (the contour on the response function where $g = \overline{z}$) that has the greatest probability of occurring. This point is known as the most probable point or MPP. An approximation to the limit state is then formed at this point to facilitate the integration required to compute the probability of failure.

The MPP search is performed in uncorrelated standard normal space ("u-space") because it simplifies the probability integration; in this space, the distance from the origin to the MPP is equivalent to the number of input standard deviations from the mean response at which the limit state lies. This distance is known as the reliability index and is denoted by β. The transformation from correlated non-normal distributions (x-space) to uncorrelated standard normal distributions (u-space) is nonlinear in general, and possible approaches include the Rosenblatt [15], Nataf [5], and Box-Cox [1] transformations. The nonlinear transformations may also be linearized, and common approaches for this included the Rackwitz-Fiessler [14] two-parameter equivalent normal and the Chen-Lind [3] and Wu-Wirsching [18] three-parameter equivalent normals. This work employs the Nataf nonlinear transformation, which occurs in the following two steps. To transform between the original correlated x-space variables and correlated standard normals ("z-space"), the CDF matching condition is used:

$$\Phi(z_i) = F(x_i) \tag{2.2}$$

where $F()$ is the cumulative distribution function of the original probability distribution. Then, to transform between correlated z-space variables and uncorrelated u-space variables, the Cholesky factor $\mathbf{L}$ of a modified correlation matrix is used:

$$\mathbf{z} = \mathbf{L}\mathbf{u} \tag{2.3}$$

where the original correlation matrix for non-normals in x-space has been modified for z-space [5].

The forward reliability analysis algorithm for computing the probability/reliability level that corresponds to a specified response level is called the reliability index approach (RIA), and the inverse reliability analysis algorithm for computing the response level that corresponds to a specified probability/reliability level is called the performance measure approach (PMA) [17]. The differences between the RIA and PMA formulations appear in the objective function and equality constraint formulations in the MPP searches. For RIA, the MPP search for achieving the specified response level $\overline{z}$ is formulated as

$$\begin{aligned} \text{minimize} \quad & \mathbf{u}^T\mathbf{u} \\ \text{subject to} \quad & G(\mathbf{u}) = \overline{z} \end{aligned} \tag{2.4}$$

and for PMA, the MPP search for achieving the specified probability/reliability level $\overline{p}, \overline{\beta}$ is formulated as

$$\begin{aligned} \text{minimize} \quad & \pm\, G(\mathbf{u}) \\ \text{subject to} \quad & \mathbf{u}^T\mathbf{u} = \overline{\beta}^2 \end{aligned} \tag{2.5}$$

where $\mathbf{u}$ is a vector centered at the origin in u-space and $g(\mathbf{x}) \equiv G(\mathbf{u})$ by definition. In the RIA case, the optimal MPP solution $\mathbf{u}^*$ defines the reliability index from $\beta = \pm\|\mathbf{u}^*\|_2$, which in turn defines the probability of failure through the probability integration.

Recent research has focused on the use of local and multipoint surrogate models to reduce the expense of the MPP search [7, 8]. All of these MPP search methods employ local optimization techniques and converge to a single MPP. But the limit state of a complex engineering application may be multimodal and possess multiple significantly probable points of failure. These points are commonly referred to as multiple most probable points (multiple MPPs) despite the misnomer. The SKO method for uncertainty quantification proposed here uses a global surrogate model and global optimization methods to reduce expense and locate multiple MPPs.

2.2. Probability Integration. For an RIA formulation, after the MPP is found and the reliability index β is known, the next step is to integrate over the failure region to calculate the probability of failure. This can be greatly simplified from Eqn. 2.1 by approximating the shape of the limit state with one over which it is easier to integrate. The simplest approximation is the first-order reliability method (FORM), which approximates the limit state as a linear function. Because β represents the distance from the mean response to the MPP in standard normal space, the probability integration simplifies to:

$$p_f = \Phi(-\beta) \tag{2.6}$$

where $\Phi()$ is the standard normal cumulative distribution function. Another alternative is the second-order reliability method (SORM), which incorporates some curvature in the limit state approximation [2, 10, 11]. Breitung applies a correction based on asymptotic analysis [2]:

$$p_f = \Phi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1+\beta\kappa_i}} \tag{2.7}$$

where κ_i are the principal curvatures of the limit state (the eigenvalues of an orthonormal transformation of $\nabla^2_{\mathbf{u}} G$, taken positive for a convex limit state). This method

essentially uses a parabolic approximation to the limit state and is more accurate for large values of β because it collapses to first-order integration at $\beta = 0$. An alternative correction in Ref. [10] is consistent with Breitung's correction in the asymptotic regime ($\beta \to \infty$) but does not approach first-order integration as $\beta \to 0$:

$$p_f = \Phi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \psi(-\beta)\kappa_i}} \tag{2.8}$$

where $\psi() = \frac{\phi()}{\Phi()}$ and $\phi()$ is the standard normal density function. Ref. [11] applies further corrections to Eqn. 2.8 based on point concentration methods.

Each of these methods makes some approximation to the shape of the limit state, making them inaccurate if the approximation is poor. If multiple MPPs are present, the true shape of the limit state cannot be linear and is likely not parabolic, so more advanced methods are needed. One possibility is to borrow concepts from system reliability where the total probability of failure is calculated as the probability of the union of multiple distinct modes of failure. If each of the n multiple MPPs on a single limit state are treated as single MPPs on n multiple limit states (with each limit state considered a distinct failure event E), then the probability of failure could be computed as:

$$p_f = P(E_1 \cup E_2 \cup \cdots \cup E_{n-1} \cup E_n) \tag{2.9}$$

Assuming the individual failure modes are independent, DeMorgan's Rule can be used to simplify this to:

$$p_f = 1 - \prod_{i=1}^{n} P(\overline{E_i}) = 1 - \prod_{i=1}^{n} [1 - P(E_i)] \tag{2.10}$$

If the reliability index for each MPP i is denoted by β_i and FORM is used for the probability integration at each MPP, the total probability of failure could be calculated by:

$$p_f = 1 - \prod_{i=1}^{n} [1 - \Phi(-\beta_i)] \tag{2.11}$$

and a similar formulation could be used for SORM methods.

Another possibility is to perform the probability integration numerically by sampling the response function. Sampling methods do not rely on a simplifying approximation to the shape of the limit state, so they can be more accurate than FORM and SORM, but they can also be prohibitively expensive because they generally require a large number of response function evaluations. The simplest sampling method is Basic Monte Carlo. Samples of the input variables are randomly generated based on their distributions and the response function is then evaluated at these input values. The value of the response function at this sample point is then compared to the limit state to determine if the observed response is a success or failure. The probability of failure is then calculated as simply the ratio of observed failures to the total number of observations. One major drawback to this method is that it will generate a large number of samples in the high-probability region of the response function, but because engineers are typically concerned with high-reliability problems, this region is of little

interest as the limit state most likely lies in a much less-probable region of the design space.

Importance sampling methods avoid this problem by centering the sampling density function at the MPP. This ensures the samples will lie in a more interesting region of the design space and is a much more efficient sampling method than Basic Monte Carlo. Adaptive importance sampling (AIS) further improves the efficiency by adaptively updating the sampling density function. Multimodal adaptive importance sampling [6, 19] is a variation of AIS that allows for the use of multiple sampling densities making it better suited for cases where multiple sections of the limit state are highly probable.

Note that Importance Sampling methods require the location of at least one MPP be known because it is used to center the initial sampling density. However, current gradient-based, local search methods used in MPP search may fail to converge or may converge to poor solutions, thus making these methods inapplicable. Because SKO is a global optimization method that does not depend on the availability of accurate gradient information, convergence to the MPP should be more reliable. Moreover, SKO has the ability to locate multiple MPPs, which would provide multiple starting points and a true multimodal sampling density for the initial steps of multimodal AIS. An additional advantage of the SKO method proposed is that a byproduct of the MPP search process is a kriging model that is accurate in the vicinity of the limit state, thereby providing an inexpensive surrogate that can be used to provide response function samples. Using SKO to locate multiple MPPs, and then using the resulting kriging model to provide function evaluations in multimodal AIS for the probability integration, could result in an accurate and efficient reliability assessment tool.

3. Sequential Kriging Optimization. Sequential kriging optimization is a global optimization method put forth by Huang et al. [12], but has its roots in the efficient global optimization (EGO) method from Jones et al. [13]. The main difference between SKO and EGO is the formulation of what is known as the expected improvement function (EIF), which is the feature that sets all EGO-type methods apart from other global optimization methods. The EIF is used to select the location at which a new training point should be added to the kriging model by maximizing the amount of improvement in the objective function that can be expected by adding that point. The general procedure of these EGO-type methods is:

1. Build an initial kriging model of the objective function
2. Use cross validation to ensure that the kriging model is satisfactory.
3. Find the point that maximizes the EIF. If the EIF value at this point is sufficiently small, stop.
4. Evaluate the objective function at the point where the EIF is maximized. Update the kriging model using this new point. Go to Step 3.

The following sections discuss the construction of the kriging model used, the form of the EIF, and then some ideas on how that EIF could be altered for SKO's application to MPP search.

3.1. Kriging Model. Kriging models are set apart from most other surrogate models by the fact that they provide not only a prediction of the modeled function value at a point, but also an indication of the uncertainty of that prediction. This uncertainty is a result of the construction of the covariance function, which is based on the idea that when input points are near one another, the correlation between their corresponding outputs will be high. As a result, the uncertainty associated with the

model's predictions will be small for input points which are near the points used to train the model, and will increase as one moves further from the training points.

It is assumed that the true function being modeled $Y()$ contains some random noise (modeled as independent identically distributed normal deviates [12]), and can be described by: [4]

$$Y(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}) + \varepsilon \tag{3.1}$$

where $\mathbf{h}()$ is the trend of the model, $\boldsymbol{\beta}$ is the vector of trend coefficients, $Z()$ is a stationary Gaussian process with zero mean and covariance defined below that describes the departure of the model from its underlying trend, and ε is the noise in the true function. The trend of the model can be assumed to be any function, but taking it to be a constant value is generally sufficient. [16] The covariance between outputs of the Gaussian process $Z()$ at points $\mathbf{a}$ and $\mathbf{b}$ is defined as:

$$Cov\left[Z(\mathbf{a}), Z(\mathbf{b})\right] = \sigma_Z^2 R_Z(\mathbf{a}, \mathbf{b}) \tag{3.2}$$

where σ_Z^2 is the process variance and $R_Z()$ is the correlation function. There are several options for the correlation function, but the squared-exponential function is common, and is used here for $R_Z()$:

$$R_Z(\mathbf{a}, \mathbf{b}) = \exp\left[-\sum_{i=1}^{d} \theta_i (a_i - b_i)^2\right] \tag{3.3}$$

where d represents the dimensionality of the problem, and θ_i is a scale parameter that indicates the correlation between the points within dimension i. A large θ_i indicates a low correlation.

The expected value $\hat{Y}()$ and variance $s^2()$ of the kriging model prediction at point $\hat{\mathbf{x}}$ are:

$$\hat{Y}(\hat{\mathbf{x}}) = \mathbf{h}(\hat{\mathbf{x}})^T \boldsymbol{\beta} + \mathbf{v}(\hat{\mathbf{x}})^T \mathbf{V}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \tag{3.4}$$

$$s^2(\hat{\mathbf{x}}) = \sigma_Z^2 - \begin{bmatrix} \mathbf{h}(\hat{\mathbf{x}})^T & \mathbf{v}(\hat{\mathbf{x}})^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{V} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{h}(\hat{\mathbf{x}}) \\ \mathbf{v}(\hat{\mathbf{x}}) \end{bmatrix} \tag{3.5}$$

where $\mathbf{v}()$ is a vector containing the covariance between $\hat{\mathbf{x}}$ and each of the n training points (defined by Eqn. 3.2), $\mathbf{V}$ is an $n \times n$ matrix containing the covariance between each pair of training points with the variance of the noise in the true function evaluations σ_ε^2 added to the diagonal terms, $\mathbf{y}$ is the vector of response outputs at each of the training points, and $\mathbf{F}$ is an $n \times q$ matrix with rows $\mathbf{h}(\mathbf{x}_i)^T$ (the trend function for training point i containing q terms; for a constant trend $q = 1$). This form of the variance accounts for the uncertainty in the trend coefficients $\boldsymbol{\beta}$, but assumes that the parameters governing the covariance function (σ_Z^2, σ_ε^2 and $\boldsymbol{\theta}$) have known values.

The parameters σ_Z^2, σ_ε^2, and $\boldsymbol{\theta}$ are determined through maximum likelihood estimation. Denote by $\mathbf{R}$ the correlation between the outputs at the training points. $\mathbf{R}$ is then equivalent to $\mathbf{V}/(\sigma_Z^2 + \sigma_\varepsilon^2)$, and the ij^{th} component of $\mathbf{R}$ is:

$$R_{ij} = \begin{cases} 1 & \text{if } i = j \\ gR_Z(\mathbf{x}_i, \mathbf{x}_j) & \text{if } i \neq j \end{cases} \tag{3.6}$$

where $g = \sigma_Z^2/(\sigma_Z^2 + \sigma_\varepsilon^2)$. The log likelihood can then be written as: [16]

$$\log\left[p(\mathbf{y}|\mathbf{R})\right] = -\frac{1}{n}\log|\mathbf{R}| - \log(\hat{\sigma}^2) \tag{3.7}$$

where $|\mathbf{R}|$ indicates the determinant of $\mathbf{R}$, and $\hat{\sigma}^2$ is the optimal value of the variance given $\boldsymbol{\theta}$ and is defined by:

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{3.8}$$

where $\hat{\boldsymbol{\beta}}$ is the generalized least squares estimate of $\boldsymbol{\beta}$ from:

$$\hat{\boldsymbol{\beta}} = \left[\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}\right]^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{y} \tag{3.9}$$

Maximizing Eqn. 3.7 gives estimates of $\boldsymbol{\theta}$ and g; recognizing that $\hat{\sigma}^2 = \sigma_Z^2 + \sigma_\varepsilon^2$, estimates of σ_Z^2 and σ_ε^2 can also be obtained.

3.2. Expected Improvement Function. The expected improvement function is used to select the location at which a new training point should be added. It does this by calculating the probability that any point in the design space will provide a better solution than the current best solution based on the predictions and uncertainties of the kriging model. An important feature of the EIF is that it provides a balance between exploiting areas of the design space where good solutions have been found, and exploring areas of the design space where the uncertainty is high. First, recognize that at any point in the design space, the kriging predictor $Y_P()$ can be described by a normal distribution:

$$Y_P(\mathbf{x}) \sim N\left[\hat{Y}(\mathbf{x}), s(\mathbf{x})\right] \tag{3.10}$$

where the expected value $\hat{Y}()$ and the variance $s^2()$ were defined in Eqns. 3.4 and 3.5, respectively. The EIF used in SKO is then defined as: [12]

$$EI(\mathbf{x}) \equiv E\left[\max\left(\hat{Y}(\mathbf{x}^*) - Y_P(\mathbf{x}), 0\right)\right]\left(1 - \frac{\sigma_\varepsilon}{\sqrt{s^2(\mathbf{x}) + \sigma_\varepsilon^2}}\right) \tag{3.11}$$

where $\mathbf{x}^*$ is the current effective best solution and is defined below. The expectation term in Eqn. 3.11 can then be computed by integrating over the distribution of $Y_P(\mathbf{x})$ with $\hat{Y}(\mathbf{x}^*)$ held constant, and can be expressed analytically as: [12, 13]

$$\begin{aligned} E\left[\max\left(\hat{Y}(\mathbf{x}^*) - Y_P(\mathbf{x}), 0\right)\right] = \left[\hat{Y}(\mathbf{x}^*) - \hat{Y}(\mathbf{x})\right] \Phi\left[\frac{\hat{Y}(\mathbf{x}^*) - \hat{Y}(\mathbf{x})}{s(\mathbf{x})}\right] \\ + \; s(\mathbf{x})\phi\left[\frac{\hat{Y}(\mathbf{x}^*) - \hat{Y}(\mathbf{x})}{s(\mathbf{x})}\right] \end{aligned} \tag{3.12}$$

The effective best solution $\mathbf{x}^*$ is determined through a utility function $\upsilon()$ to account for the uncertainty in the evaluation of the true function (due to the noise ε). This function is defined as: [12]

$$\upsilon(\mathbf{x}) \equiv -\hat{Y}(\mathbf{x}) - \eta s(\mathbf{x}) \tag{3.13}$$

where η is a tunable constant that indicates the desired level of risk aversion. For instance, if $\eta = 1$, it implies a willingness to trade 1 unit of the predicted value for 1 unit of the standard deviation of prediction uncertainty. [12] Because unobserved points in the design space will likely have greater uncertainty than the observed training

points, the utility function is only maximized over the n training points for the sake of computational efficiency [12]:

$$\mathbf{x}^* = \max\left[\upsilon(\mathbf{x}_1), \upsilon(\mathbf{x}_2), \ldots, \upsilon(\mathbf{x}_{n-1}), \upsilon(\mathbf{x}_n)\right] \tag{3.14}$$

Note that if the function being modeled is deterministic ($\sigma_\varepsilon = 0$), then Eqn. 3.11 collapses to Eqn. 3.12, and the training points are directly interpolated by the kriging model (therefore $s(\mathbf{x}) = 0$ at the training points), so the utility function becomes simply $\upsilon(\mathbf{x}) = -\hat{Y}(\mathbf{x}) = -Y(\mathbf{x})$.

The point at which the EIF is maximized is selected as an additional training point. In Ref. [12] this maximization is performed using a Nelder-Mead simplex approach, which is a local optimization method. Because the EIF is quite often multimodal (particularly in the early stages of the SKO process) it is expected that Nelder-Mead may occasionally fail to converge to the true global optimum, thus global optimization techniques for finding the point that maximizes the EIF will be investigated. With the new training point added, a new kriging model is built and then used to construct another EIF, which is then used to choose another new training point, and so on, until the value of the EIF at its maximized point is below some specified tolerance.

It is important to understand how the use of this EIF leads to optimal solutions. This particular EIF is tailored to the problem of bound-constrained minimization. Eqn. 3.12 indicates how much the objective function value at $\mathbf{x}$ is expected to be less than the predicted value at the current effective best solution. It contains a balance between exploitation of regions of the design space where good solutions have been discovered, and exploration of regions that have not been well explored and thus have greater uncertainty. Because the kriging model provides a random distribution at each predicted point, points with good predicted values and even a small variance will have a significant probability of producing a better solution (exploitation), but so will points that have relatively poor predicted values and greater variance (exploration). The problem of MPP search, however, is made more complicated due to its inclusion of equality constraints. For this problem, the EIF must be reformulated to give it knowledge of these constraints.

3.3. Incorporation of Equality Constraints for MPP Search. In this work, the equality constraints will be enforced through the use of a merit function. Possible merit functions include a penalty function P, a Lagrangian function L, and an augmented Lagrangian function L_A, the forms of which are:

$$P(\mathbf{x}, r_p) = f(\mathbf{x}) + r_p \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x}) \tag{3.15}$$

$$L(\mathbf{x}, \boldsymbol{\lambda}_c) = f(\mathbf{x}) + \boldsymbol{\lambda}_c^T \mathbf{c}(\mathbf{x}) \tag{3.16}$$

$$L_A(\mathbf{x}, \boldsymbol{\lambda}_c, r_p) = f(\mathbf{x}) + \boldsymbol{\lambda}_c^T \mathbf{c}(\mathbf{x}) + r_p \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x}) \tag{3.17}$$

where $f()$ is the objective function, $\mathbf{c}()$ is the vector of equality constraint functions (converted to a standard form such that for feasible point $\mathbf{c}(\mathbf{x}) = 0$), and $\boldsymbol{\lambda}_c$ is the vector of Lagrange multipliers for the equality constraints. The proper form of the penalty schedule r_p has not yet been fully investigated for this work, but one possibility may be:

$$r_p = \exp\left[\frac{k + \text{offset}}{10}\right] \tag{3.18}$$

where k is the current iteration number of the SKO process. Simple zeroth-order updates for the Lagrange multipliers are used:

$$\boldsymbol{\lambda}_c^{k+1} = \boldsymbol{\lambda}_c^k + 2r_p\mathbf{c}(\mathbf{x}) \tag{3.19}$$

The Lagrange multipliers are applied whenever a new iterate is accepted, whereas the penalty schedule is updated for any iteration that fails to reduce the constraint violation.

4. Computational Experiments. Preliminary work has investigated the use of the penalty function in Eqn. 3.15. The application of this method to the PMA case is fairly straightforward because the objective function involves the minimization of the response function that the kriging model represents, which is the initial intent of the SKO method. Furthermore, the equality constraint for PMA is only applied to the location of the optimal solution, which is deterministic. In this case, the constraint function can be written as $c(\mathbf{u}) = \|\mathbf{u}\| - \overline{\beta}$. This is then added to the expected value of the kriging model at $\mathbf{u}$, to produce the penalty function:

$$\hat{P}(\mathbf{u}) = \hat{G}(\mathbf{u}) + r_p\left(\|\mathbf{u}\| - \overline{\beta}\right)^2 \tag{4.1}$$

which is written in terms of $G()$ and $\mathbf{u}$ because the kriging model is a surrogate for the response function in u-space. Note that the penalty function is denoted as $\hat{P}$ to emphasize that only the expected value of the kriging model is being penalized rather than the full distribution at $\mathbf{u}$; the variance $s(\mathbf{u})$ remains the same.

Application to the RIA case is less clear because the objective function involves the minimization of a deterministic function. Moreover, the constraint is stochastic because it involves the kriging model, so feasibility can only be determined in a probabilistic sense. The constraint function could be formulated as $c(\mathbf{u}) = \hat{G}(\mathbf{u}) - \overline{z}$, and added to the deterministic function being minimized to produce the penalty function:

$$\hat{P}(\mathbf{u}) = \|\mathbf{u}\| + r_p\left(\hat{G}(\mathbf{u}) - \overline{z}\right)^2 \tag{4.2}$$

For both formulations, the resulting expected improvement and utility functions would be as follows:

$$EI(\mathbf{u}) = \left\{\left[\hat{P}(\mathbf{u}^*) - \hat{P}(\mathbf{u})\right]\Phi\left[\frac{\hat{P}(\mathbf{u}^*) - \hat{P}(\mathbf{u})}{s(\mathbf{u})}\right] + s(\mathbf{u})\phi\left[\frac{\hat{P}(\mathbf{u}^*) - \hat{P}(\mathbf{u})}{s(\mathbf{u})}\right]\right\} \cdot \left(1 - \frac{\sigma_\varepsilon}{\sqrt{s^2(\mathbf{u}) + \sigma_\varepsilon^2}}\right) \tag{4.3}$$

$$\upsilon(\mathbf{u}) = -\hat{P}(\mathbf{u}) - \eta s(\mathbf{u}) \tag{4.4}$$

Preliminary results with a static r_p have shown that the penalty function provides poor scaling between the objective function and the constraint violation; one or the other simply overwhelms the EIF. The Lagrangian methods should help provide the scaling needed.

Another idea is to add the penalty (for either the RIA or PMA formulation) to the true function evaluations that are used to build the kriging model. The EIF could

then be used in the form of Section 3.3.2 with no modifications. The advantage to this is that the entire kriging model is built with some knowledge of the constraints, whereas in Eqns. 4.1-4.4 only the expected value of the kriging model is modified by the penalty, while the variance remains the same. The disadvantage to this method is that the kriging model that results as a byproduct of the optimization will not be a model of the true function, but rather of the penalized function. However, after the MPPs are found, the final set of training points could be evaluated without the penalty and used to build a kriging model that could then be used in the probability integration. This adds slightly to the expense, but this method may provide a better handling of the constraints.

5. Conclusions. The prevalence of engineering problems defined by expensive, nonlinear response functions has necessitated the development of reliability assessment methods that are both efficient and accurate. To that end, this paper has presented an idea based on the application of sequential kriging optimization to MPP search. Because this is a global optimization method, it is capable of locating multiple MPPs, which allows for more accurate probability integration. In order to apply SKO to MPP search, a method of properly incorporating the equality constraints must be developed. One possibility is through the use of a merit function, several forms of which are being investigated. The formulation of a reliability assessment tool based on sequential kriging optimization is still in its early stages, but initial results have shown promise.

REFERENCES

[1] Box, G.E.P. and Cox, D.R., An Analysis of Transformations, *J. Royal Stat. Soc.*, Series B, Vol. 26, 1964, pp. 211-252.

[2] Breitung, K., Asymptotic Approximation for Multinormal Integrals, *J. Eng. Mech., ASCE*, Vol. 110, No. 3, 1984, pp. 357-366.

[3] Chen, X. and Lind, N.C., Fast Probability Integration by Three-Parameter Normal Tail Approximation, *Struct. Saf.*, Vol. 1, 1983, pp. 269-276.

[4] Cressie, N.A.C., *Statistics for Spatial Data*, revised edition, 1993 (Wiley: New York).

[5] Der Kiureghian, A. and Liu, P.L., Structural Reliability Under Incomplete Probability Information, *J. Eng. Mech., ASCE*, Vol. 112, No. 1, 1986, pp. 85-104.

[6] Dey, A. and Mahadevan, S., Ductile Structural System Reliability Analysis using Adaptive Importance Sampling, *Struct. Saf.*, Vol. 20, 1998, pp. 137-154.

[7] Eldred, M.S. and Bichon, B.J., New Second-Order Formulations for Reliability Analysis and Design, *AIAA J.*, in preparation.

[8] Eldred, M.S., Agarwal, H., Perez, V.M., Wojtkiewicz, S.F., Jr., and Renaud, J.E., Investigation of Reliability Method Formulations in DAKOTA/UQ, (to appear) *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, Taylor & Francis Group.

[9] Haldar, A. and Mahadevan, S., *Probability, Reliability, and Statistical Methods in Engineering Design*, 2000 (Wiley: New York).

[10] Hohenbichler, M. and Rackwitz, R., Improvement of Second-Order Reliability Estimates by Importance Sampling, *J. Eng. Mech., ASCE*, Vol. 114, No. 12, 1988, pp. 2195-2199.

[11] Hong, H.P., Simple Approximations for Improving Second-Order Reliability Estimates, *J. Eng. Mech., ASCE*, Vol. 125, No. 5, 1999, pp. 592-595.

[12] Huang, D., Allen, T.T., Notz, W.I., and Zeng, N., Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models, *J. Global Opt.*, Vol. 34, 2006, pp. 441-466.

[13] Jones, D., Shonlau, M., and Welch, W., Efficient Global Optimization of Expensive Black-Box Functions, *INFORMS J. Comp.*, Vol. 12, 1998, pp. 272.283

[14] Rackwitz, R. and Fiessler, B., Structural Reliability under Combined Random Load Sequences, *Comput. Struct.*, Vol. 9, 1978, pp. 489-494.

[15] Rosenblatt, M., Remarks on a Multivariate Transformation, *Ann. Math. Stat.*, Vol. 23, No. 3, 1952, pp. 470-472.

[16] Sacks, J., Schiller, S.B., and Welch, W., Design for computer experiments, *Technometrics*, Vol. 31, 1989, pp. 41-47.

[17] Tu, J., Choi, K.K., and Park, Y.H., A New Study on Reliability-Based Design Optimization, *J. Mech. Design*, Vol. 121, 1999, pp. 557-564.

[18] Wu, Y.-T. and Wirsching, P.H., A New Algorithm for Structural Reliability Estimation, *J. Eng. Mech., ASCE*, Vol. 113, 1987, pp. 1319-1336.

[19] Zou, T., Mourelatos, Z., Mahadevan, S., and Tu, J., Reliability Analysis of Automotive Body-Door Subsystem, *Rel. Eng. and Sys. Saf.*, Vol. 78, 2002, pp. 315-324.

TOWARD MASSIVELY MULTITHREADED SHORTEST PATH ALGORITHMS: APPLICATIONS AT THE NANOSCALE

JOSEPH CROBAK* AND JOHN BERRY†

Abstract. The single source shortest path (SSSP) problem often arises at the nanoscale. In this paper, we briefly detail the SSSP problem and three nanoscale problems involving the SSSP problem. Next, we argue that the Cray MTA-2 is the most suitable architecture for solving the SSSP problem on graphs at the nanoscale.

1. Introduction. The single source shortest path (SSSP) problem is a fundamental problem in graph theory. Historically, the problem arises in road and computer network routing. At the nanoscale, the SSSP problem arises in several applications, such as path planning for nano-robotics and the design of a new computational device based upon Quantum-dot Cellular Automata (QCA). At such a small scale, the number of particles in a system is very large (Aluminum contains 6×10^{22} atoms per cm^3 [1]). Thus, graphs representing molecules or other nanoscale particles can contain billions or more vertices.

In the case where non-negative edge weights are not allowed, the SSSP problem can be solved in $O(m + n \log n)$ time using Dijkstra's classical algorithm [4], where n is the number of vertices and m is the number of edges. State of the art algorithms (e.g. [5, 6, 8, 10]) can solve the SSSP in time nearly linear with respect to $n + m$. For modestly sized graphs (up to tens of millions of vertices), a modern personal computer can solve the problem using one of these modern approaches in a few seconds.

Graphs with more than a few million vertices, such as graphs of particles at the nanoscale, do not fit into the main memory of a personal computers. Because memory accesses by a graph algorithm are irregular, cache is ineffective Traversing the graph leads to numerous cache misses as vertices and edges are accessed. Graph algorithms generally examine the entire graph, which leads to a large number of memory operations. These memory operations are the bottleneck when running a large graph algorithm on a traditional microprocessor.

In order to process larger graphs, a more powerful architecture is required. Modern distributed memory supercomputers are not ideal because the graph must be partitioned across the cluster. For unstructured classes of graphs such as social networks, this partitioning is ineffective. Graphs arising from nano-scale problems do have some geometric structure. However, even assuming an effective partitioning, it remains to be seen whether the cache of traditional processors can be utilized during the run of a graph algorithm on a giant instance. The Cray MTA-2 architecture we outline below uses the strategy of latency tolerance rather than the traditional strategy of latency mitigation via cache.

Rather than distributing memory over the nodes in a cluster and forcing the programmer to manage communication, a shared memory machine provides a global address space to all of the processors or cores. The most popular implementation of shared memory is Symmetric MultiProcessing (SMP). A SMP machine still suffers from some of the performance problems such as the effects of caching and the limitations of RAM.

*Lafayette College

†Sandia National Laboratories

[1] Atom Density = (Avogardo's Number) × (Density) / (Atomic Mass Units) where Density of Aluminum = 2.7 g/cm^3 and AMU = 26.98g/mol.

Because of the limitations of these hardwares, we have studied graph algorithms on the Cray MultiThreaded Architecture (MTA). As we describe in section 3 this architecture is an appealing alternative to all of the above architectures when implenting graph algorithms for very large graphs.

In this paper, we shows a significant link between problems in nano-technology and the SSSP. In addition, we argue that the Cray MTA-2 architecture is the designed well to solve this problem on graphs of particles at the nano-scale. The outline of the paper is as follows. First, we discuss several different problems at the nano-scale to which a SSSP solver could be applied. Second, we discuss the architecture of the Cray MTA-2 the benefits of this architecture for nano-scale problems. Finally, we summarize the benefits of the Cray MTA-2 on the problems.

2. Applications. The Single Source Shortest Path (SSSP) problem arises in several applications at the nano-scale. These range from nano-robotic planning to Quantum-Dot Cellular Automata (QCA)- a technology that may soon replace silicon-based computers. Nano-scale problems that rely upon shortest path algorithms are described in the following sections 2.1 through 2.3.

2.1. The Curvature-Constrained Shortest-path Problem. The curvature constraint shortest-path problem is a planning problem with applications in nano-robotics. For example, this problem models nanorobotic navagation through the heart in the presence of obstacles (e.g. red blood cells). The problem is:

> *Given a robot R and a set $P = \{p_1, p_2, ..., p_m\}$ of m polygons, compute the shortest path from position s to position t while avoiding the polygons. Further, this path must be computed within a certain curvature (thus ensuring that R can maneuver through the path).*

In three-dimensions, this problem is at least NP-hard, and it is NP-hard in many cases in two-dimensions. For a complete overview of the complexity of the problem and its variants, see [9]. Regardless, approximation algorithms exist to solve the problem. For example, the algorithm of Wang and Agarwal [12] computes a path whose length is at most $(1 + \varepsilon)$ times the length of an optimal curvature-constrained path in $O((n^2/\varepsilon^2) \log n)$-time, where n is the number of vertices describing the positions of polygons in P. This algorithm reduces the problem to the Single Source Shortest Path (SSSP) problem.

2.2. Nanoparticle Self-Assembly. The nanoparticle self-assembly problem is very similar to the Curvature-constraint Shortest Path problem. This problem has the added goal of assembling (or moving) particles into certain areas. Formally, the problem is:

> *Given a robot R, a set $P = \{p_1, p_2, ..., p_m\}$ of m polygons, a set $O = \{o_1, o_2, ..., o_n\}$ of n objects, and a set $D = \{d_1, d_2, ..., d_n\}$ of n object destinations, plan a shortest path for R such that an object from O is placed at each $d \in D$. The robot pushes the particles into place and must avoid all polygons in P.*

A solution to this problem is suggested by Makaliwe and Requicha in [7]. Their planner operates in three phases. They are:

- *Assignment* All objects are identical, so the planner must decide which object will be placed at which destination.
- *Path find* Next, the planner discovers paths that the robot can take to move objects to destinations. Because a path might not exist for a given assignment, a new assignment might be needed.

FIG. 2.1. *(a) QCA cell representing 1, (b) QCA cell representing 0, and (c) QCA cells in a wire. The empty circles are electron "wells" and the black circles represent electrons.*

- *Ordering of paths* Finally, the planner must decide in which order to move each object. It attempts to minimize the distance traveled.

During the *Path find* phase of the planner, the authors solve the SSSP problem using Dijkstra's algorithm.

2.3. Quantum-Dot Cellular Automata. Quantum-Dot Cellular Automata were first suggested in the early 1990s [11]. A QCA is made up of *cells* that represent either a '0' or a '1' depending on the configuration of the electrons within the cell. When cells are placed next two each other, they arrive at the same configuration as the electrons in adjacent cells repel one another. For an example, see Figure 2.1.

QCA cells can be arranged to simulate digital logic gates (such as AND, NOT, and OR gates). They can also be arranged to create more complex structures, such as adders and shifters. As complexity of structure increases, problems arise. For example, it is not completely evident how to make two QCA wires cross. One solution to this problem is to eliminate QCA wire crossings. Chaudhary, et al. [3] reduce this problem to the node-duplication based crossing-elimination (NDCE) problem. Their algorithm to solve the problem runs in linear-time, but is not guaranteed to find the optimal solution (i.e. there may be more node-duplications than needed). Ultimately, they reduce the problem to the shortest path problem.

3. The Cray MTA-2. The Cray MTA-2 is a shared memory multiprocessor architecture. Each processor in the MTA-2 system has 128 hardware threads. On each clock cycle, a context switch occurs and a different hardware thread takes over execution. Furthermore, the instruction pipeline is 21 stages deep, and each thread can have only one instruction in the pipeline at any time. With sufficient parallelism, therefore, each thread can afford to wait 21 cycles before expecting a load or store to have occurred. Since each thread can have up to 8 memory references in flight and the memory cycle time is about 150 cycles, a sufficiently parallel application can keep the machine almost 100% utilized.

This model is ideal for graph algorithms since they are latency-dominated. Furthermore, MTA-2 processors incur no overhead from caching, which would be ineffective anyway. The shared memory archictecture of the MTA-2 also has the advantage of fine-grain synchronization, which efficiently allows mutual exclusion at the vertex or edge level in graph algorithms.

As demonstrated in [1, 2], the MTA-2 runs search kernel graph algorithms much more efficiently than parallel computers based on tradional microprocessors. Furthermore, in work done during the summer of 2006, we have implemented Thorup's SSSP algorithm [10] and demonstrated scalable performance on the MTA-2. Instances of the three problems mentioned in Section 2 are potentially very large and computationally intense. All three problems either reduce to the shortest path problem or solve the

shortest path problem in sub-problems. Thus, SSSP algorithms on the MTA-2 should be very good for solving these problems.

4. Summary. We have given a brief overview of three different problems at the nanoscale that relate to the shortest path problem. The curvature-constrained shortest path problem and the nanoparticle self-assembly problems are two important problems in planning. At this scale, the input to these problems is large, and the communication pattern for shortest path computations is irregular. Thus the best architecture for solving these problems is the Cray MTA-2. Likewise, the number of QCA cells can grow very large when trying to emulate a modern microprocessor.

REFERENCES

[1] D.A. Bader and K. Madduri, *Designing multithreaded algorithms for breadth-first search and st-connectivity on the cray mta-2*, in The 35th International Conference on Parallel Processing (ICPP), Columbus, OH, August 2006.

[2] Jonathan Berry and Bruce Hendrickson, *Graph software development and performance on the mta-2 and eldorado*, in Gray User Group 2006, 2006.

[3] A. Chaudhary, D. Z. Chen, X.S. Hu, K. Whitton, M. Niemier, and R. Ravichandran, *Eliminating wire crossings for molecular quantum-dot cellular automata implementation*, in ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design, Washington, DC, USA, 2005, IEEE Computer Society, pp. 565–571.

[4] E.W. Dijkstra, *A note on two problems in connection with graphs*, Numerical Mathematics, 1 (1959), pp. 269–271.

[5] Andrew V. Goldberg, *A simple shortest path algorithm with linear average time*, in Lecture Notes in Computer Science, vol. 2161, 2001.

[6] Torben Hagerup, *Improved shortest paths on the word ram*, in ICALP '00: Proceedings of the 27th International Colloquium on Automata, Languages and Programming, London, UK, 2000, Springer-Verlag, pp. 61–72.

[7] J.H. Makaliwe and A.A.G. Requicha, *Automatic planning of nanoparticle assembly tasks*, in Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2001, 2001, pp. 288–293.

[8] S. Pettie and V. Ramachandran, *Computing shortest paths with comparisons and additions*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02), SIAM, 6–8 2002.

[9] John Reif and Zheng Sun, *Nano-robotics motion planning and its applications in nanotechnology and biomolecular computing*, in NSF Design and Manufacturing Grantees Conference, Jan 1999.

[10] Mikkel Thorup, *Undirected single-source shortest paths with positive integer weights in linear time*, Journal of the ACM, 46 (1999), pp. 362–394.

[11] P.D. Tougaw and C.S. Lent, *Logical devices implemented using quantum cellular automata*, Journal of Applied Physics, 75 (1994), pp. 1818–1825.

[12] Hongyan Wang and Pankaj K. Agarwal, *Approximation algorithms for curvature-constrained shortest paths*, in SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, 1996, Society for Industrial and Applied Mathematics, pp. 409–418.

OPTIMIZING MESH OPTIMIZATION

ERICK JOHNSON† AND MICHAEL BREWER‡

Abstract. Research has been focused on creating a better initial mesh and optimizing a mesh in order to correct any errors. Mesh optimization is used to both untangle and smooth a mesh to minimize skewed elements. Each is critical as most analysis software cannot compensate for malformed elements. The time required to optimize a mesh also becomes an issue and finding the line between time and quality needs to be addressed in order to provide the best solution. One idea to address quality and time is to cull (hide) the elements or nodes which meet certain criteria, thus time and effort no longer need be spent on improving something that is "good enough." Culling would allow the available computational power to be spent improving sections of a mesh which are "worse." An important issue being addressed is ensuring the quality of a mesh is the same or better with culling than it would have been without.

1. Introduction. There is a significant drive to perform simulations and experiments on computers. This is because time and money can be saved in the design process if an actual prototype does not need to be produced and tested every time a change is made. Using computers also has the potential to increase the number of computational experiments that can be performed as a direct result of the savings of time and money. This is specially true in the micro/nano sciences where current limitations in manufacturability significantly impacts the number of or even feasibility of certain experiments. In order to produce viable finite element or finite volume solutions though, a high quality mesh of the model or system needs to be created.

A mesh is the discretization of a computational model, which is needed to perform an analysis on the same model. To easily move from the model to the solution a suitable mesh is required. An improper mesh (too coarse, tangled, etc) is not necessarily going to allow a correct analysis, while a mesh that is extremely fine may take an inordinate amount of time to solve. The way analyses are performed necessitates a mesh of high quality. Another issue to consider is the amount of time it takes to create a high quality mesh. For various reasons in the design through analysis process the largest portion of time is spent on meshing and its related procedures. Depending on the application, the line between quality and time changes and needs to be redetermined, which in turn influences the operations on the geometry to be meshed. For example, there may be features on the geometry that are smaller than the size of the mesh elements. These features may adversely influence the mesh and need to be removed [4].

One solution to these problems is mesh optimization. Mesh optimization can produce a higher quality mesh[1], which is easier to analyze; thus saving time and yielding a more correct solution. This is done by one of two methods: a topology change or node movement. Node movement is a more intuitive approach because it maintains the connectivity between nodes (it doesn't change which neighbors a node is connected to). Node movement optimization can be solved by either a global or local scheme. The global scheme takes the entire system and solves for a direction to move each individual node for an improved node location. It then moves all the nodes at once. The local scheme moves through the mesh on a node-by-node basis and solves for the optimal location of each node based on its neighboring nodes or elements.

†Carnegie Mellon University
‡Sandia National Laboratory

[1]It needs to be noted the optimization is based on some predetermined idea of what "good" is. This depends on the desired result within a certain quality metric.

However, in both cases the *entire* mesh is iterated over until the final solution is achieved. This paper looks at a method to decrease the time and computational cost of mesh optimization while maintaining a high quality mesh as the final product. The method is applied with node movement in a local scheme. The algorithm is implemented in Mesquite [1], the mesh optimization library used by CUBIT [6]. A more rigorous comparison of mesh optimization techniques can be found in [2].

2. Culling: A "Good Enough" Approach. A question should be asked, "Why spend time and effort improving a node's position when its neighboring elements are 'good enough'?" Based on some definition of "good" that all the elements should be in the most optimal case, there is some value nearby that can be defined as "good enough." If an element falls in between these then the time spent improving it could be better spent improving another element. In other words, while improving a mesh comprised of a list of nodes, instead of improving the whole list on each iteration, why not hide select nodes so the list decreases in size over a series of passes. The idea of "good enough" does not lead to the conclusion the mesh could be better, because the improvements made to a culled node beyond this point are negligible in terms of the entire mesh.

This method has been seen in many implementations of the Laplacian algorithm (e.g. [6] [1]). The Laplacian smoother alone moves a node to the center of its neighbors; however, in these implementations it hides the nodes that are known to be close to their optimal positions[2]. The research for this paper stems from these methods.

2.1. The Algorithm. This idea of culling by itself seems fairly logical; however there is a very important difference between how it was described and how it is presented in Algorithm 1. As seen in line 7, there is a means to make the node and its neighbors visible. This addendum to the culling algorithm is what ensures a high quality mesh. Otherwise the culling algorithm potentially hides a node which could be improved more. As an example, let's say a "good" position for a node is the geometric center of its neighbors. If one node does not meet the criteria, it is moved and hidden with no trouble. However assume more nodes don't meet the criteria. In this case the optimal position will change as the neighboring nodes move. Since nodes are only hidden in the list there can be no increase in improvement when a node is hidden early. The addition of 1.7 allows for a node previously culled to be pulled back into the list in order to ensure it's at the best location.

Algorithm 1 Current Culling Algorithm

```
1: while a node exists do                          ▷ at least one node is not hidden
2:     while termination criterion has not been met do
3:         move the node(s) in this patch
4:         if the node is now 'good enough' then
5:             hide the node(s) in this patch
6:         else
7:             make the node(s) and its neighbor(s) visible
8:         end if
9:     end while
10: end while
```

[2] That is, a node is hidden as long as the nodes around it have not moved significantly since the last time the node's position was updated

Another item requiring a moment is the termination criterion, the mechanism for stopping the iterations of the optimization procedure. This is not just a reiteration of the culling criterion (what has currently been defined as "good enough"), though it is an important part. Additional termination criteria will provide multiple ways to stop when implemented in a program. It may not seem evident at first, but in some cases a node cannot be moved to a position satisfying the criterion (or criteria). As such the algorithm would be stuck in an infinite loop with no way out. Another case is where an optimizing function (e.g. conjugate gradient) may require multiple iterations before arriving at the best location when optimizing the position of a single node. These extra criterion allow the algorithm to move through the whole mesh hopefully making enough changes in the previous pass to successfully cull a troublesome node.

2.2. Benefits. There are obvious benefits to using the culling algorithm. Removing the nodes that are no longer needed significantly decreases the time spent optimizing a large mesh when compared to the global scheme[3]. Even if nodes are periodically made visible, the overall trend should be a decrease in the number of nodes. Also, since the algorithm is applied to the local scheme there should also be a large savings on memory when compared to the global scheme. In each case this is because the algorithm will be manipulating a small patch as opposed to a matrix of the entire system.

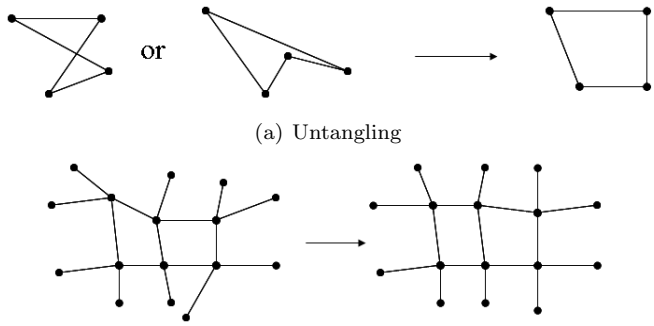
3. Untangling With The Culling Algorithm. Mesh optimization can both untangle and smooth a mesh to increase its quality. To untangle a mesh nodes are moved until all the elements are convex shapes (Fig 3.1(a)). Smoothing is a little more qualitative in that the desire is to make the mesh a more continuous pattern (Fig 3.1(b)); this is obviously more subjective than whether an element has a negative area[4] or not. For this reason the paper focuses on untangling; however, it can be easily applied to smoothing algorithms as well. All of this optimization is done to aid the finite element analysis in reaching the correct solution as quickly as possible. Poor element shapes significantly reduce the accuracy of a solution, and many solvers will not accept tangled elements. Even if a solver does allow them, it impacts the accuracy of the solution. Though a lot of smoothing methods inherently incorporate some untangling of the mesh (trying to make smoother elements will push all elements toward a better shape), applying only a smoother to a tangled mesh does not guarantee it will become untangled. Therefore, ensuring an untangled mesh prior to smoothing becomes an important issue as it allows us to then use a shape improvement algorithm that does not guarantee non-inverted elements.

If the quality metric to define untangling is equation (3.1) as defined by [5], where α is the element corner area and β is some value defined by the user and maintains $0 < \beta < \overline{\alpha}$, then untangled elements will result in $\mu = 0$, while tangled elements are a positive value capable of being minimized. If β were not included the most optimal value for a tangled element would be when its area was zero, thus β pushes an element toward a positive area. There are other methods to untangle a mesh (e.g. [3]; however equation (3.1) is already implemented in Mesquite can be easily formulated to a smooth optimization scheme and has proven to be a reliable method.

$$\mu = |\alpha - \beta| - (\alpha - \beta), \tag{3.1}$$

[3]This is the trend that has been observed for a small selection of tangled meshes; however, there is no hard evidence to prove it is always the case

[4]volume can be substituted throughout



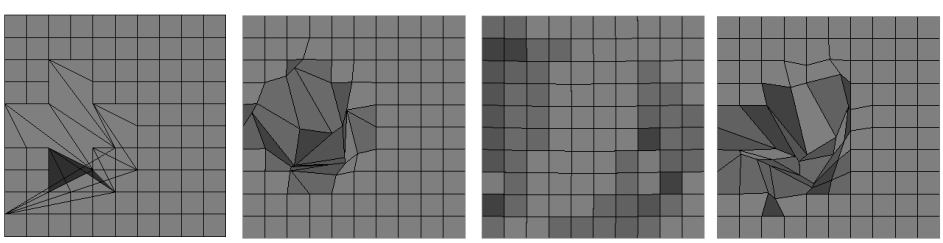
(a) Untangling

(b) Smoothing

FIG. 3.1. *Direction of mesh optimization*

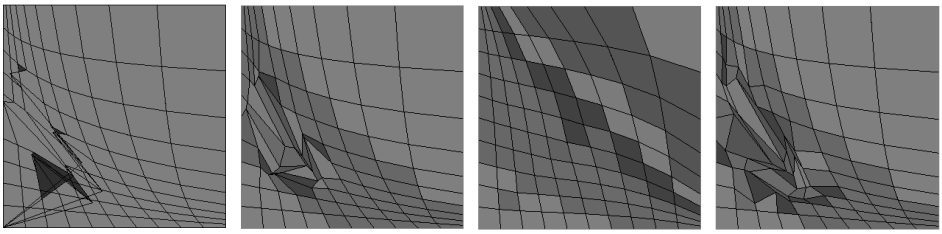
Both "good" and "good enough" can be defined based on the quality of a node's neighbors, μ. However, a different definition may yield a better termination. If the culling criterion is changed to the gradient of the untangling quality metric, its minimum will be a node's most optimal position and may be easier to define than a direct relation to quality.

The following sets of figures show a comparison of untangling between the global scheme, the smart Laplacian smoother (a Laplacian smoother in Mesquite), and the local scheme with a gradient culling. The smart Laplacian smoother is used frequently because of its speed and simplicity; however, the comparison is unfair because it is not explicitly formulated to untangle a mesh. The goal is to show two points about the local untangler with culling. First, the culling algorithm influences a small region surrounding the tangled area. This buffer zone increases the chance a mesh can be untangled (compare 3.4(b) to 3.4(d)). Second, because the local untangler limits its working space it does not influence the entire mesh (compare 3.3(c) to the rest). This is very advantageous, as the biased mesh shows, because changing the whole mesh effects the solution and any previous work done to alter the mesh. Though none of the methods were able to successfully untangle 3.4(a), the culling algorithm was able to produce an untanlged mesh in two cases: 1) running the algorithm a second time, and 2) in one run after changing the criterion from the quality gradient of the element to the quality of the element. The global scheme did not succeed when changed similarly, and the smart Laplacian method is unable to untangle the mesh due to the the concave nature of the hole.



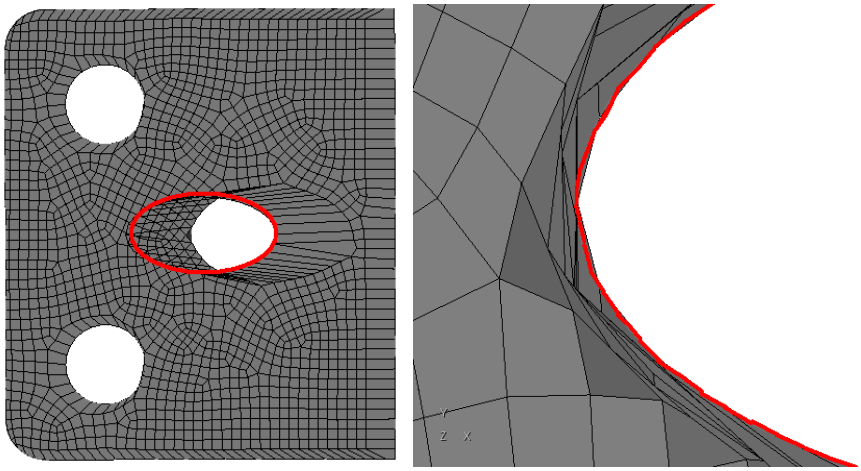
(a) Original Mesh (b) Global (c) Smart Laplacian (d) Local with Culling

FIG. 3.2. *Untangling a 10x10 block. All methods are successful.*

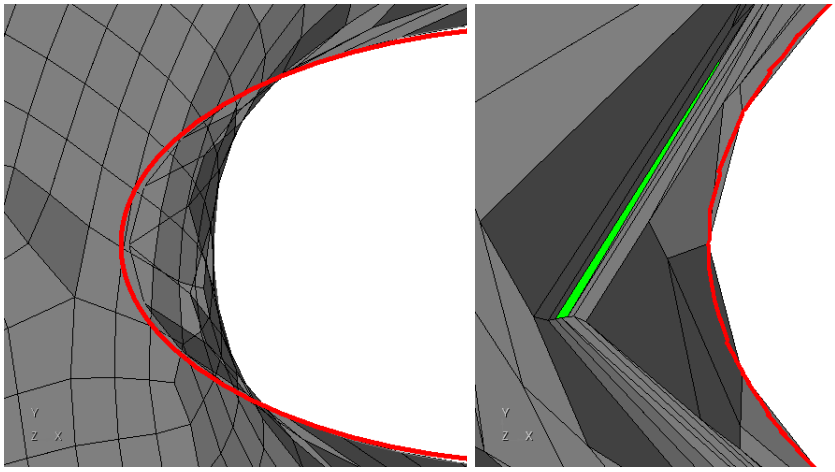


(a) Original Mesh (b) Global (c) Smart Laplacian (d) Local with Culling

FIG. 3.3. *Untangling a 10x10 block with a curve bias. All methods are successful.*



(a) Original Mesh (b) Global



(c) Smart Laplacian (d) Local with Culling

FIG. 3.4. *Untangling a surface where the mesh is outside the geometry. (b) & (c) fail with multiple tangled elements remaining, (c) also does not constrain itself to the geometry, (d) fails with 1 element remaining*

4. Conclusion & Future Work. A culling algorithm appears to be a good approach to improving mesh quality more quickly. It cannot guarantee a mesh will be untangled, but seems to provide at least the same consistency as the global scheme with the untangling metric from [5]. The primary focus of future work should be determining the relationship between the culling criterion and the quality metric being applied. In a lot of cases the quality metrics being used are scale dependent and influence the definitions of "good" and "good enough." Determining these relationships will allow a comparison of the resultant quality to ensure culling is a viable means to successfully improve a mesh. A rigorous approach needs to be accomplished as well to compare time and memory usage of the local and global schemes. Though the current algorithm appears to perform well trying alternative algorithms may provide a more robust method (one potential alternative is Algorithm 2). This example may require more time to complete the optimization; however it should prevent removing *some* of the nodes too early.

Algorithm 2 Alternative Culling Algorithm

```
while a node exists do                          ▷ at least one node is not hidden
    while termination criterion has not been met do
        move the node(s) in this patch
        make the node(s) and its neighbor(s) visible
    end while
    if the node is now 'good enough' then
        hide the node(s) in this patch
    end if
end while
```

REFERENCES

[1] M. Brewer, L. Diachin, P. Knupp, T. Leurent, and D. Melander, *The mesquite mesh quality improvement toolkit*, in Proceedings, 12th International Meshing Roundtable, pp. 239–250.

[2] L. Freitag, P. Knupp, T. Munson, and S. Shontz, *A comparison of inexact newton and coordinate descent mesh optimization techniques*, in Proceedings, 13th International Meshing Roundtable, pp. 19–22.

[3] L. Freitag and P. Plassmann, *Local optimization-based untangling algorithms for quadrilateral meshes*, in Proceedings, 10th International Meshing Roundtable, pp. 397–406.

[4] K. Inoue, T. Itoh, A. Yamada, T. Furuhata, and K. Shimada, *Face clustering of a large-scale cad model for surface mesh generation*, Computer Aided Design, 33 (2001), pp. 251–261.

[5] P. Knupp, *Hexahedral and tetrahedral mesh untangling*, Engineering with Computers, 17 (2001), pp. 261–268.

[6] Sandia National Laboratory, *Cubit: Geometry and mesh generation toolkit*, `http://cubit.sandia.gov/`.

TOWARD A REVERSIBLE QUANTUM-DOT CELLULAR AUTOMATA MICROARCHITECTURE

S. FROST-MURPHY[¶], E. DEBENEDICTIS[‖], AND P. KOGGE[**]

Abstract. Quantum-dot cellular automata (QCA) is a promising emerging technology with device characteristics very different from transistors with the potential for very low power operation. Coupling QCA with the reversible computing paradigm may lead to a realm where the energy dissipation due to the deletion of bits dominates the power equation. This work explores several problems on the path toward a reversible QCA microarchitecture including a survey of high level reversible languages, a floorplan for reversible general purpose computing, a reversible planar crossover design, and a new look at the set of 3-input, 3-output universal reversible gates.

1. Introduction. There are many indicators that a radical change in computing technology will be required soon. How "soon" is defined is a hotly debated question, but the ITRS roadmap is filled with "red blocks" signifying problems that it is unknown how the problem will be solved. These red blocks start showing up as soon as 2008 and dominate by 2014. This seems to indicate that the problem is approaching sooner rather than later. In addition to the oft predicted demise of exponentially scaling CMOS, there are application domains where even end of the roadmap CMOS (could it be realized) would be unable to provide the required computing power [4]. These are two factors driving research into novel nano-scale devices and architectures to make use of them.

There are computing paradigms that have the potential to beat end-of-the-roadmap CMOS and supply the computing power needed by these types of applications. The solution seems to be combining reversible computing with one of the promising emerging devices. This work explores a particular device called quantum-dot cellular automata (QCA), and is focused toward the design of a reversible QCA microarchitecture that will harness the strengths of the reversible computing paradigm with the strengths of QCA.

Towards this end, several thrusts have been simultaneously explored including: a floorplan for general purpose reversible computing; a reversible coplanar QCA wire crossover design; three-input, three-output universal reversible gate amenable to QCA circuit design.

1.1. Reversible Computing. Reversible computing builds on a well-established thermodynamics history starting with Maxwell's Demon. In 1887, Maxwell's thought-experiment that showed that destroying, or erasing, information results in heat dissipation, specifically at least $kT\ln(2)$ where k is Boltzman's constant and T is the temperature of the system.

In traditional computing based on CMOS technology, the energy dissipated by the device and clock independent of the function being performed dominated any energy dissipation due to irreversibility. However, non-traditional technologies such as quantum-dot cellular automata (QCA) offer a new opportunity to experimentally verify the connection between physical devices and information.

The key insight of reversible computing is that information does not need to be destroyed during computation. There is a fundamental connection between logical

[¶]University of Notre Dame
[‖]Sandia National Laboratories
[**]University of Notre Dame

TABLE 1.1
Truth Table of AND Operation

Irreversible AND			Reversible AND				
A	B	A and B	A	B	A	B	A and B
0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0
1	0	0	1	0	1	0	0
1	1	1	1	1	1	1	1

reversibility and physical reversibility, and if a logically reversible system is implemented by physically reversible devices, there need not be any power dissipation due to information erasure.

To be reversible, a function needs to be one-to-one. Any function can be made to be one-to-one by saving the inputs. For instance, it is clear from examining the truth table of the AND operation (Table 1.1) that AND is naturally irreversible since there are three zeros in the output making it impossible to determine what the inputs were from just the output. However, by copying the inputs to the output the function becomes one-to-one. In this way, any irreversible function can be made to be reversible at the expense of carrying additional information, or garbage data, forward through the computation.

Copying the inputs to the outputs is an awkward way that reversibility can be forced on any irreversible function. However, there is a set of naturally reversible gates that do not incur this garbage penalty. There are seven classes of 3-input, 3-output reversible gates that are universal, meaning that any function can be computed using only the gate being considered. These will be discussed later.

To take full advantage of reversible computing, the physical implementation of the logic must be physically reversible. Traditional CMOS is not physically reversible. There are many emerging nanodevices that are being explored for extending Moore's law beyond the end of CMOS. One such device is quantum-dot cellular automata (QCA). The basic device operates much differently from traditional CMOS and has the potential for very low power operation to the point that energy dissipated due to information destruction will be a significant if not dominant factor of the overall heat dissipation of the system. In addition to being very low power, QCA has the potential to be clocked in the tens of terahertz, and molecular implementations of QCA cells are being explored which would lead to very high device densities [9] [11]. In short, QCA has great potential for very fast, low power, dense computing, making it an attractive device with the potential to beat end of the roadmap CMOS and extend computing beyond what is possible with CMOS.

1.2. QCA Basics. The basic device is a QCA cell that consists of four quantum dots arranged in a square. When two excess electrons are introduced into the cell, they can tunnel between the dots in the cell and naturally form two stable states that correspond to a logical zero and a logical one (figure 1.1a) [10].

If several QCA cells are placed in a row, and the first cell has a fixed polarization, the value will ripple down the cells like dominoes forming a wire (figure 1.1b).

The basic gate is a majority gate which is a three-input, one-output gate (figure 1.1c). The output is the majority of the inputs (i.e. if the input contains two or more zeros, the output will be a zero; if the input contains two or more ones, the output will be a one). By fixing one of the inputs to zero, an AND gate is formed. Similarly,

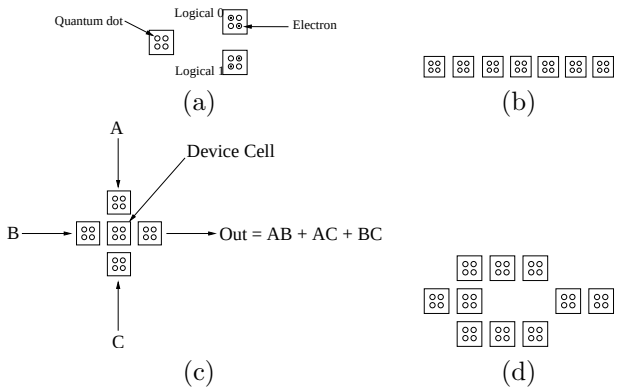


FIG. 1.1. *QCA Device Basics: (a) QCA Cell schematic, (b) QCA Wire, (c)Majority Gate, (d) Inverter*

fixing one of the inputs to one forms an OR gate. An inverter can also be formed using QCA cells (figure 1.1d) which is the final piece for a logically complete set.

To be able to design a computer, there must be a way for wires to cross. In traditional technology, this is accomplished by vias and multiple metal layers. QCA will most likely have only one layer of devices. There are several proposals to allow planar crossovers that will require very little area but may be difficult to fabricate. There is also a planar logical crossover that requires more area, but will not require extremely precise fabrication (figure 1.2).

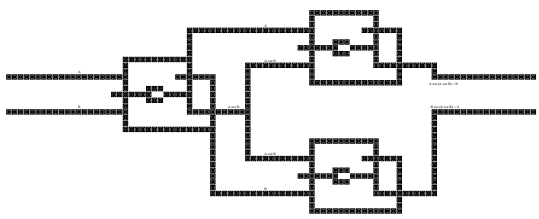


FIG. 1.2. *A planar crossover circuit using 3 XOR gates with no internal physical crossovers.*

The final, and perhaps most important, piece of the QCA puzzle is the clock. Unlike a traditional CMOS clock signal, the QCA clock is a different kind of phenomenon than the QCA signal. The clock is an electric field that controls the tunneling of the electrons between the dots of the cells. When the electric field is high, the electrons will not be able to tunnel, fixing the state of the cell and allowing it to act as a driver cell. Where the field is low, the cells will have no value and will not effect the computation of neighbor cells. While the field is rising, the cells are assuming new values that will be latched when the field is high.

There are two clocking strategies being proposed in the context of reversible computing. The first is Landauer clocking in which a sinusoidal wave moves across the QCA circuit in one direction. The other is Bennett style clocking in which the leading edge of the clocking wave sweeps across the QCA circuit staying high behind the leading edge. At the edge of the circuit being clocked, the edge is retracted, leaving the clock low behind the edge (figure 1.3). Landauer clocking supports reversible computing with fully reversible gates. Bennett clocking allows reversible computing

to be done with irreversible parts [8].

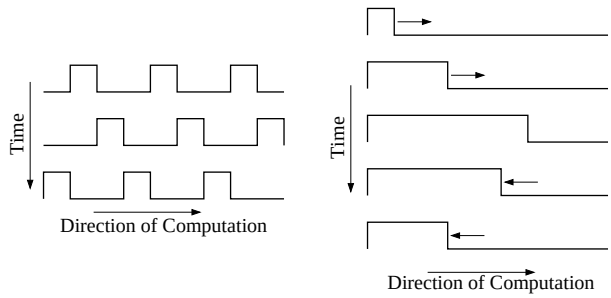


FIG. 1.3. *Two clocking schemes: (a) Landauer clocking wave, (b) Bennett clocking*

The actual implementation of the QCA cells and the clock are matters of ongoing research by electrical engineers and materials scientists. Proof-of-concept QCA cells have been fabricated using metal-dots that operate at cryogenic temperatures [3] [19]. Exploration continues into molecular implementations and at size scales between the two extremes.

QCA has a solid foundation in irreversible architecture research as well. A processor and memory have been designed showing that general purpose computing can be done with QCA [13] [7], and the fundamental properties of the QCA from an architectural viewpoint have been explored [18] [17] [16] [15].

The rest of this paper describes ongoing work toward designing a general purpose reversible QCA architecture including a proposed floorplan for implementing Bennett's algorithm in "hardware", a discussion of a reversible logical crossover circuit, and a discussion of the sets of universal reversible gates that are targets to be implemented in QCA.

2. Reversible Language Survey. Examining high level reversible languages reveals the fundamental operations that a reversible microarchitecture will need to support. In addition, revealing the reversibility of the microarchitecture to higher levels creates opportunities for applications and algorithms to exploit that reversibility.

The high-level languages that exist for more than theoretical analysis (i.e. Turing machines) are Janus, Psi-Lisp, and R. Janus was a 1982 side project by two Caltech undergraduates - Chris Lutz and Howard Derby [12]. A letter from Lutz to Landauer is all that remains of this imperative language. It is a limited language, and while R was developed independently, many of the same constructs are used, and both guarantee reversibility in similar ways. Janus will not be discussed further. Psi-Lisp was developed by Henry Baker in 1992 [1]. It is a reversible version of a linear lisp he developed previously to manage garbage collection in lisp. It is a functional language. R was developed by Mike Frank as a toy language in which to write a few simple reversible benchmarks for his doctoral research [6]. R is a C-like procedural language with some lisp-like features.

All of these languages share the following characteristics:

- All primitive operations are reversible
- There is no assignment (so no copying), only variable binding
- All branches and conditionals must maintain the original test value

2.1. Psi-Lisp. Beyond this, Psi-Lisp is a linear language. This corresponds to a sequent calculus in which the structural rules of weakening and contraction are disal-

lowed. In practice, this means that making and destroying copies of information is not allowed. This restriction leads to an odd way of implementing functions and recursions. Rather than copies of the arguments being passed to the functions or recursive call, the values of the variable and the arguments are swapped. (All arguments begin with a null value.) By the end of the function or recursive call, all values must be swapped back to their original position so that the variable has a value.

2.2. R. R does not have the same argument swapping scheme, but it does require that any variables bound in a let statement must have the same value at the end of executing the let body as it has at the beginning of the let body. On the face of it, this is not as stringent a requirement as in Psi-Lisp, but the ultimate effect is similar.

In addition, much of the uncomputing is hidden from the programmer. For instance, decomputation of expressions (which do not change the stored value of a variable) is specified by the compiler with no mechanism for the programmer to manage that decomputation.

Michael Frank rejects the sort of linearity discussed above as a requirement for reversibility. However, despite this rejection, R itself appears to be linear. As future work, it will be interesting to determine if linearity is necessary for reversibility or if, given reversible primitives, it is merely sufficient for reversibility.

3. Collapsed Bennett Model. At a high level of abstraction, reversibility can be forced onto any algorithm or circuit by saving the inputs and all intermediate results. However, depending on the function, this can lead to an exponential explosion in the amount of data that needs to be stored. Charles Bennett proposed an algorithm that minimizes the amount of data that needs to be stored at the cost of execution time [2]. Using Bennett's algorithm, any irreversible algorithm can be divided into segments that will be calculated, have the intermediate result latched so it can be used by the next segment, and then when it is no longer needed, uncalculate the intermediate result so it does not have to be stored. Bennett's algorithm is an optimal ordering for the computing and uncomputing the segments. Figure 3.1 shows an example of the order of computations and uncomputations for an algorithm broken into eight segments.

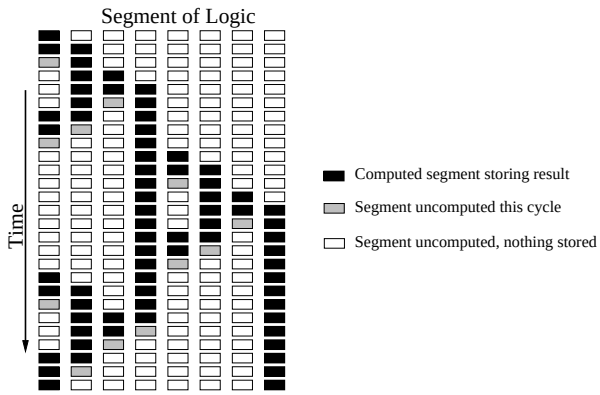


FIG. 3.1. *Bennett's algorithm divides an algorithm into stages (8 stages in this example) and selectively computes and decomputes them to store the least amount of data necessary to maintain the reversibility of an irreversible algorithm.*

In previous work, DeBenedictis (not published) introduced the beginnings of a potentially implementable model that collapses Bennett's reversible algorithm tree

into a single level of logic with a stack at either end of the combinational logic and a shifting mechanism to pop the top of one stack and push it onto the other stack (i.e. shift left or shift right). In this work, we further developed this model and designed a simple ripple-carry adder to demonstrate the proposed operation.

A schematic of the components can be seen in figure 3.2. It consists of a left stack, an area of combinational logic, a shifter unit, a shift-disable area, and a right stack. In addition, the logic unit as a whole can be disabled by adjusting the voltage bias across the area, and separately the shifter can be similarly disabled.

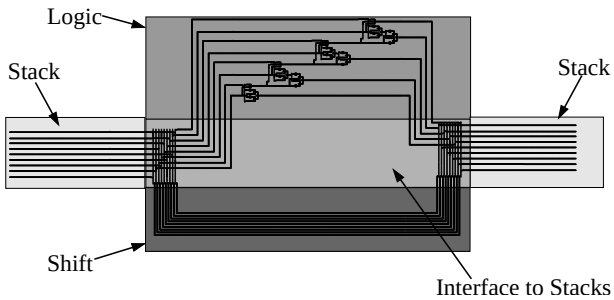


FIG. 3.2. *The regions of the collapsed Bennett layout include two stacks, a logic or computational area, a shift area that allows data to be transferred between the stacks, and an interface between the stacks and the logic and shift regions.*

The collapsed Bennett model operates as follows:

1. Initial input begins at the top of the left stack
2. A cascade shaped Bennett clock moves across the logic section (starting at the left, the clock goes high and stays high as the clock front travels to the right).
3. Result is latched at the top of the right stack
4. The cascade is retracted, decomputing the logic (leaving the results latched)
5. A set of results is shifted to prepare for the next stage
   a) Right stack is popped and the value is pushed onto the left stack (via the shifter) and is ready to be the input in the next cycle
   b) Left stack is popped and the value is pushed onto the right stack (via the shifter) and is ready to be decomputed in the next cycle
6. This process is repeated with the shifting determined by Bennett's 1989 algorithm [2]

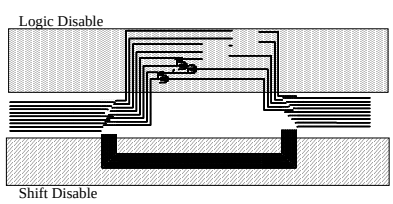


FIG. 3.3. *There are two disable sections in this layout. The top area disables the logic, while the bottom area disables the shift. While disabled, the QCA cells have no value and do not contribute to the computation of any nearby cells.*

A simple adder is shown in this section to illustrate the operation of the layout and demonstrate the required interfaces between the computation and storage. One

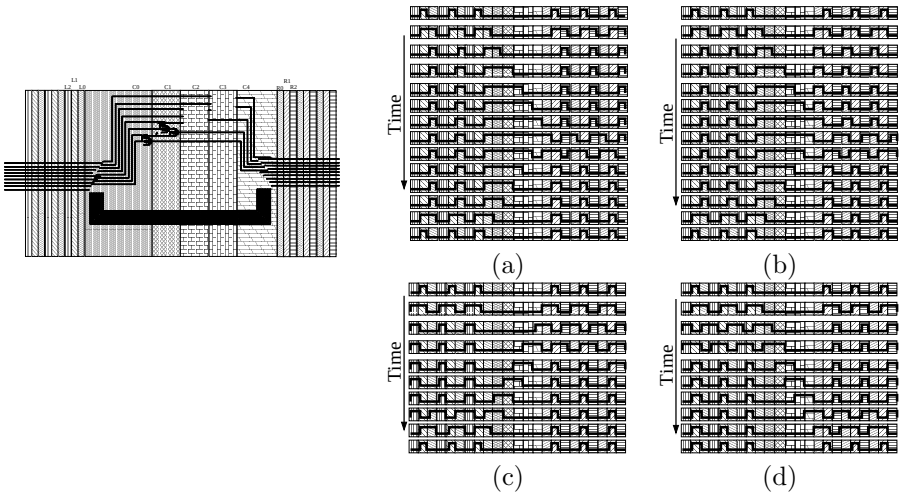


FIG. 3.4. *Clocking signals required for four modes of operation of collapsed Bennett clocking layout: (a) Compute, (b) UnCompute, (c) Shift Left, (d) Shift Right*

can imagine sandwiching an entire processor between two stacks in this manner for a general purpose reversible processor.

4. Reversible Crossover. The traditional method for decomputing a circuit is to make a mirror circuit which exactly undoes the computation done in the original circuit. The desired output is copied out before the circuit is decomputed. One of the challenges with QCA is that physical crossovers will most likely be very difficult. However, if a majority gate, even one with one held input, is to be decomputed by a separate mirror circuit, there must be a physical crossover (figure 4.1).

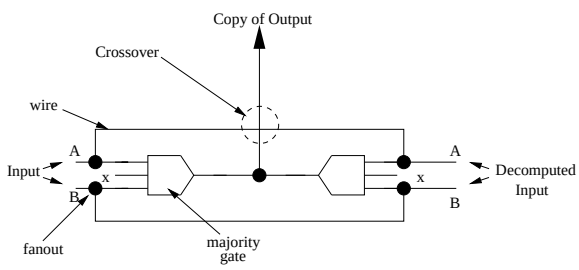


FIG. 4.1. *A physical crossover is unavoidable if reversibility is to be done with wave style clocking in which the clock travels in one direction only. x is a held input.*

There are several logical crossover circuits [14]. The one discussed in this paper has fewer majority gates than others. If done irreversibly, the crossover circuit will dissipate 21 bits (figure 4.2)

This circuit can be embedded in a Bennett clocking zone and be executed reversibly (figure 4). However, at the end of the Bennett clocking period, there will be 4 bits stored – two input bits, and two output bits. The reversible crossover circuit can be used as a component in a larger reversible architecture, and it will be an architectural decision whether to dissipate those 4 bits for the sake of throughput, or decompute those 4 bits at the expense of throughput.

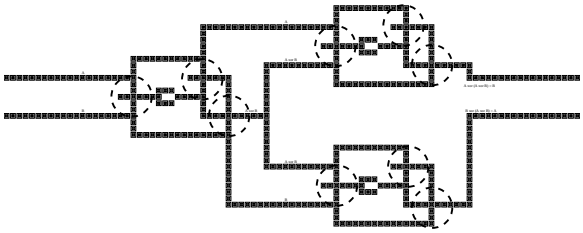


FIG. 4.2. *The logical crossover circuit will dissipate three bits at each majority gate, and two bits at each gate with a held input for a total of 21 bits for this design.*

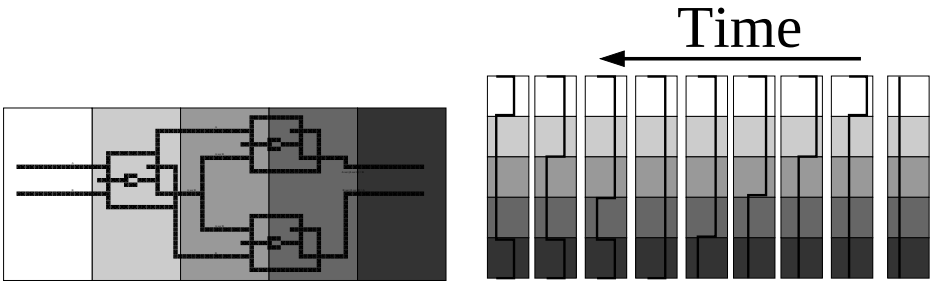


FIG. 4.3. *The Bennett clock reversible crossover circuit and the clock signals needed to execute the function reversibly.*

Identifying the necessity of physical crossovers for reversibility with Landauer clocking is a significant result. It has a substantial impact on defining the design space and illuminating the set of tradeoffs that need to be made.

5. Reversible Gates. The Collapsed Bennett layout shows a method for doing reversible computing using irreversible building blocks. Another method for reversible computing is using reversible building blocks with the Landauer clocking scheme. In the Landauer clocking scheme, a sinusoidal clock signal travels across the QCA circuit in one direction only. If the circuit itself is reversible (i.e. destroys no information) it can operate in this clocking scheme without dissipating any information. This section examines what reversible building blocks should be examined for QCA implementation. More specifically, this section examines the set of universal 3-input, 3-output reversible gates.

5.1. Traditional Description. The most general description of a classical three bit gate can be written as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \oplus M \begin{pmatrix} x \\ y \\ z \end{pmatrix} \oplus N \begin{pmatrix} yz \\ xz \\ xy \end{pmatrix} \oplus \begin{pmatrix} p \\ q \\ r \end{pmatrix} xyz \qquad (5.1)$$

where M and N are three by three matrices [5]. We are interested in reversible gates only. To be reversible, the gate must meet three basic criteria. First, the gate must have the same number of inputs as outputs. Second, the columns of the M matrix above must be linearly independent. Third, the final vector in equation 5.1 must contain all zeros. In addition, the first vector in equation 5.1 consists of additive

TABLE 5.1
Six Possible Values of M

$$\begin{pmatrix} 100 \\ 010 \\ 001 \end{pmatrix} \quad \begin{pmatrix} 100 \\ 001 \\ 010 \end{pmatrix} \quad \begin{pmatrix} 010 \\ 100 \\ 001 \end{pmatrix}$$

$$\begin{pmatrix} 001 \\ 100 \\ 010 \end{pmatrix} \quad \begin{pmatrix} 010 \\ 001 \\ 100 \end{pmatrix} \quad \begin{pmatrix} 001 \\ 010 \\ 100 \end{pmatrix}$$

constants only and has no impact on the reversibility of the gate. Taking all this into consideration, the general description of a 3-bit reversible gate then becomes:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \end{pmatrix} \oplus N \begin{pmatrix} yz \\ xz \\ xy \end{pmatrix} \tag{5.2}$$

The columns in M must be either linearly independent or be the three non-zero vectors in a two-dimensional space for the gate to be reversible. The gates can be classified into seven classes in terms of four vectors V_i which are the exclusive or of columns of M. Let M_1 represent the first column of M, M_2 represent the second column of M, and so on. Then $V_1 = M_2 \oplus M_3$, $V_2 = M_1 \oplus M_3$, $V_3 = M_1 \oplus M_2$, $V_4 = M_1 \oplus M_2 \oplus M_3$. If the columns of M are linearly independent, all of the vectors V_i are different. If the columns of M are the non-zero vectors in a two-dimensional space, though, two of the V_i will be the same. The rest of this discussion will assume the case in which the columns of M are linearly independent. The six possible matrices for M can be seen in table 5.1.

Notice that the effect of the M matrix is to permute the output. The N matrix determines how the inputs interact with each other to produce the outputs. N can be considered as permuting the V_i vectors. The two matrices can be separated in considering the classes of reversible gates. The N matrix will be used to classify the gates, and the M matrix will determine the different gates switching the classes.

Based on how N permutes the four V vectors, the set of reversible gates can be divided into seven classes. The description of the N matrix for each class can be seen in table 5.2.

The seven classes of gates are described in [5] in terms of the vectors V_i. The classes exchange and cycle the four vectors in various ways. This nomenclature is very useful for physicists but much less so for computer scientists. Translating the gate classes into Boolean logic reveals a new type of symmetry between the classes. This is described below.

5.2. A New Symmetry. In considering these gates for classical reversible computing, it is more useful to translate these classes into the language of computer science, boolean logic. The first step is to write the seven classes of gates in terms of the basic components: the columns of M and the inputs themselves. This representation can be seen in table 5.3.

In words, the effect of the seven classes of gates can be described as follows:

- Class 1: There is no interaction between the input bits.
- Class 2: Conditionally swaps x and y based on z.

TABLE 5.2
Definition of N columns for Seven Classes of Reversible Gates

Class 1:	$N_1 = N_2 = N_3 = 0$
Class 2:	$N1 = N_2 = M_1 \oplus M_2$ $N_3 = 0$
Class 3:	$N_1 = M_1 \oplus M_2$ $N_2 = M_2 \oplus M_3$ $N_3 = M_1 \oplus M_3$
Class 4:	$N_1 = M_1$ $N_2 = N_3 = 0$
Class 5:	$N_1 = M_1$ $N_2 = N_3 = M_2 \oplus M_3$
Class 6:	$N_1 = M_1 \oplus M_2$ $N_2 = M_2$ $N_3 = 0$
Class 7:	$N_1 = M_1 \oplus M_2$ $N_2 = M_2 \oplus M_3$ $N_3 = M_3$

- Class 3: For each input bit, if the bit is zero, the result is the AND of the other two bits. If the bit is one, the result is the OR of the input "above" and the negation of the input "below" (where x is above y which is above z which is above x).
- Class 4: Beginning with this gate class each input bit is treated differently. x', the x output bit, is the exclusive or of the x input and the AND of y and z. The y and z inputs are passed to the output unchanged.
- Class 5: Each output bit is calculated by a different function. x' is the exclusive or of x and the AND of y and z. y' is the exclusive or of y, the AND of y and x, and the AND of z and x. z' is the exclusive or of z, the AND of z and x, and the AND of x and y.
- Class 6: x' is the exclusive or of x and the AND of y and z. y' is the exclusive or of y, the AND of y and z, and the AND of z and x. z' is the unchanged z input.
- Class 7: x' is the exclusive or of x and the AND of y and z. y' is the exclusive or of y, the AND of y and z, and the AND of z and x. z' is the exclusive or of z, the AND of z and x, and the AND of x and y.

When the classes are described in terms of boolean logic, a new type of symmetry and a new type of classification is revealed. There is a set of four functions that can be used to describe each of the outputs described in the above list. Analogous to a high-level programming language function, the order of the arguments is important. It is interesting that the 21 output values (7 classes of gates x 3 outputs each) can be

Table 5.3
Seven Classes of Reversible Gates

Gate Class	Gate Actions	Inverse Actions
$G_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z$	$\begin{pmatrix} IV \\ IV \\ IV \end{pmatrix}$	$\begin{pmatrix} IV \\ IV \\ IV \end{pmatrix}$
$G_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus (M_1 \oplus M_2)yz \oplus (M_1 \oplus M_2)xz$	$\begin{pmatrix} I \\ II \\ IV \end{pmatrix}$	$\begin{pmatrix} I \\ II \\ IV \end{pmatrix}$
$G_3 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus (M_1 \oplus M_2)yz \oplus (M_2 \oplus M_3)xz \oplus (M_1 \oplus M_3)xy$	$\begin{pmatrix} II \\ II \\ II \end{pmatrix}$	$\begin{pmatrix} V \\ V \\ V \end{pmatrix}$
$G_4 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus M_1yz$	$\begin{pmatrix} III \\ IV \\ IV \end{pmatrix}$	$\begin{pmatrix} III \\ IV \\ IV \end{pmatrix}$
$G_5 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus M_1yz \oplus (M_2 \oplus M_3)xz \oplus (M_2 \oplus M_3)xy$	$\begin{pmatrix} III \\ I \\ II \end{pmatrix}$	$\begin{pmatrix} III \\ VI \\ VII \end{pmatrix}$
$G_6 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus (M_1 \oplus M_2)yz \oplus M_2xz$	$\begin{pmatrix} III \\ II \\ IV \end{pmatrix}$	$\begin{pmatrix} V \\ III \\ IV \end{pmatrix}$
$G_7 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M_1x \oplus M_2y \oplus M_3z \oplus (M_1 \oplus M_2)yz \oplus (M_2 \oplus M_3)xz \oplus M_3xy$	$\begin{pmatrix} III \\ II \\ II \end{pmatrix}$	$\begin{pmatrix} V \\ V \\ VIII \end{pmatrix}$

described with only four functions. The key insight is in the relative ordering of the arguments. Consider that x is always considered to be above y which is above z which wraps around to be above x. One can use A,B,C to represent the relative positions of the function arguments. Allow A to be above B which is above C which is above A. In other words, if A=x then B=y and C=z. If A=y, then B=z and C=x; and if A=z then B=x and C=y. The value of A is fixed by the output row being calculated.

Given this representation, there are only four functions that occur in the seven gate classes. They are:

- Type I: $A \oplus AC \oplus BC$
- Type II: $A \oplus AB \oplus BC$
- Type III: $A \oplus BC$
- Type IV: A (no change)

where A is the input position (i.e. x in the top position of the vector, y in the middle position, and x in the bottom position) and B and C are determined by the value of A.

The gates that invert the seven classes add four more types of operations:

- Type V: $A(\neg C) \vee CB$
- Type VI: $(\neg B)(\neg C)A \vee B(A \vee C)$
- Type VII: $(\neg C)(\neg B)A \vee C(A \vee B)$
- Type VIII: $A(\neg B) \vee B(A \oplus C)$

Using these types, the classes can be described as in the last two columns of table 5.3. Any one of the gates described here could be used as a basic building block for

fully reversible QCA circuits.

6. Conclusion. All of this work continues to be ongoing. These intermediate results identify a path toward a reversible microarchitecture for QCA. The reversible language survey reveals the high level constrcuts that need to be supported by a reversible ISA and microarchitecture. The collapsed Bennett layout strategy now needs to take the crossover problem into consideration. Future work will include this as well as designing QCA circuits that will implement a universal reversible gate efficiently.

REFERENCES

[1] Henry G. Baker, *Nreversal of fortune: The thermodynamics of garbage collection*, Springer-Verlag, 1992.

[2] Charles Bennett, *Time/space trade-offs for reversible computation*, SIAM J. Comput., 18 (1989), pp. 766–776.

[3] Gary H. Bernstein, Islamshah Amlani, Alexei Orlov, Craig Lent, and Gregory Snider, *Observation of switching in quantum-dot cellular automata cell*, Nanotechnology, (1999), pp. 166–173.

[4] Erik DeBenedictis, *The path to extreme computing*, in Zettaflops Workshop, held in conjunction with LACSI'04, 2004.

[5] Erik DeBenedictis and Carlton Caves, *Notes and homeworks from phys 581.* Course taken by Erik DeBenedictis at the University of New Mexico in spring 2006, taught by Carlton Caves, 2006.

[6] Michael P. Frank, *Reversibility for Efficient Computing*, PhD thesis, Massachusetts Institute of Technology, 1999.

[7] Sarah Frost, *Memory architecture for quantum-dot cellular automata*, master's thesis, University of Notre Dame, March 2005.

[8] C.S. Lent, S.E. Frost, and P.M. Kogge, *Reversible computation with quantum-dot cellular automata*, in 1st International Workship on Reversible Computing, 2005.

[9] Craig S. Lent and Beth Isaksen, *Clocked molecular quantum-dot cellular automata*, IEEE Trans. on Electron Devices, 50 (2003), pp. 1890–1896.

[10] Craig S. Lent and P. Douglas Tougaw., *A device architecture for computing with quantum dots*, Proceedings of the IEEE, 85 (1997).

[11] Marya Lieberman, Sudha Chellamma, Bindhu Varughese, Y uliang Wang, Craig Lent, Gary Bernstein, Gregory Snider, and Frank Peiri s, *Quantum-dot cellular automata at a molecular scale*, Ann. N.Y. Acad. Sci., (2002), pp. 225–239.

[12] Chris Lutz, *Janus: A time-reversible language.* personal letter, April 1986. Web-published by Michael Frank *http://www.cise.ufl.edu/ mpf/rc/janus.html*.

[13] Michael T. Niemier, *Designing digital systems in quantum cellular automata*, master's thesis, University of Notre Dame, April 2000.

[14] ———, *Personal communication.* July 2006.

[15] Michael T. Niemier and Peter M. Kogge, *Designing complex logic systems with qca devices*, in Great Lakes Symposium of VLSI, March 1999.

[16] ———, *Logic-in-wire: Using quantum dots to implement a microprocessor*, in International Conference on Electronics, Circuits, and Systems (ICECS '99), September 1999.

[17] ———, *Exploring and exploiting wire-level pipelining in emerging technologies*, in International Symposium of Computer Architecture, Sweden, July 2001, ISCA 2001, pp. 166–177.

[18] ———, *Problems in designing with qcas: Layout = timing*, Int. J. of Circuit Theory and Applications, 29 (2001), pp. 49–62.

[19] A. O. Orlov, I. Amlani, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, *Experimental demonstration of a binary wire for quantum-dot cellular automata*, Applied Physics Letters, 74 (1999), pp. 2875–2877.

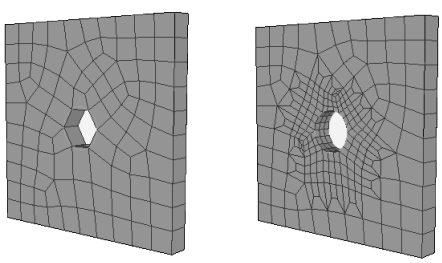
COMPOSITE REFINEMENT OF UNSTRUCTURED CONFORMAL HEXAHEDRAL MESHES

MICHAEL H. PARRISH*, MATTHEW L. STATEN†, AND MICHAEL BORDEN‡

Abstract. Localized hexahedral (hex) modification algorithms modify unstructured conformal meshes allowing greater user control over mesh density and quality. Hex refinement increases the density of the mesh in a specified region, proving extremely useful in many analyses. Previous research, using solely directional refinement theory, made the implementation of localized hex modification computationally expensive as well as unable to handle concave refinement regions and self-intersecting hex sheets. Composite Refinement is a new procedure that combines old and new theories to create an efficient and robust algorithm that is able to handle the above stated problems. The Composite Refinement procedure uses two different methodologies in the refinement process. These are: 1) Total Hex Refinement and 2) Directional Hex Refinement. Total Hex Refinement refines a hex in one step using one of seven templates. As each hex is processed, the correct template is selected and replaces the processed hex. Directional Refinement which was previously used for all refinement is only used when the more efficient Total Hex Refinement algorithm is impossible. A ranking system allows Directional Refinement to be done in the proper order thus eliminating conformity problems.

1. Introduction. Hex refinement is a mesh modification procedure which increases element density in a localized region. A refined mesh can increase the accuracy of the analysis within the refined region because of the increased element density as shown in Figure 1.1(a) and Figure 1.1(b). While hex refinement can be an invaluable tool, previous theory and implementation were unable to provide needed capabilities and made hex refinement computationally expensive. The work presented in this paper therefore, attempts to solve these issues.

The following section discusses previous hex refinement theory and highlights problems addressed by the current work. Composite Refinement will then be introduced, accompanied with an abbreviated outline of the algorithm. A comparison between Composite Refinement and the old hex refinement algorithm will follow showing that Composite Refinement is a more robust and efficient way to perform hex refinement.



(a) Before Refinement (b) After Refinement

FIG. 1.1. *Purpose of Hex Refinement*

2. Background. Conformal refinement of unstructured hexahedral meshes was previously done utilizing the dual of the mesh also known as the spatial twist con-

*Brigham Young University
†Sandia National Laboratories

tinuum (STC) [1]. The dual is a geometric representation that explicitly defines the connectivity characteristics of a mesh [2]. A component of the dual is a hex sheet as shown in Figure 2.1. Hexes were refined in hex sheets thus making the refinement conformal because of the connectivity properties associated with the dual of the mesh. While refinement using hex sheets has the advantage of conformity, three problems have arisen that limit both capability and speed of this theory. These three critical issues are: 1) Self-intersecting hex sheets, 2) Concavities, and 3) Scalability. Composite Refinement is an attempt to solve these three issues while maintaining a conformal mesh.

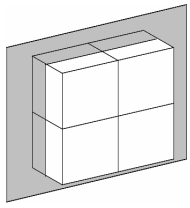


FIG. 2.1. *Hex Sheet(Shown in Gray) of 4 Hexes*

2.1. Self-Intersecting Hex Sheets. A hex sheet starts at a boundary and ends at a boundary or it will form a closed loop. Sometimes meshing algorithms will create self-intersecting hex sheets. This means that before reaching the end, the hex sheet will include hexes more than one time. The old algorithm was unable to handle multiple refinement directions in the same hex sheet. In fact, it would create a non-conformal mesh which made an accurate analysis impossible. The example below in Figure 2.2 depicts the self-intersecting hex sheet.

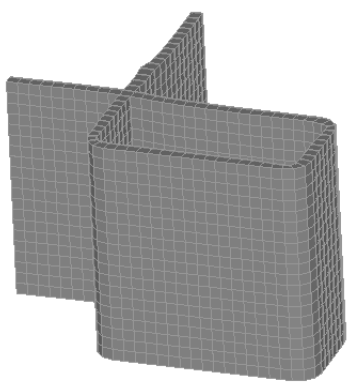


FIG. 2.2. *Example of Self-Intersecting Hex Sheet*

2.2. Concavities. Figure 2.3 shows a simple concave region where the concavity is formed by the shaded hexes. Concavities present a new set of problems for hex refinement. The old algorithm could not handle concavities so it would add hexes[1] to the hex sheet until no concavities remained. While this solves the concavities problem, it leads to another problem which is excessive refinement. Excessive refinement can

[1]Non-shaded hex represents one of four hexes that would be added to remove the concavity

cause unwanted modification of a mesh,slow down the analysis process, and reduce accuracy. The templates to handle concavities exist but were not implemented in the old algorithm [3].

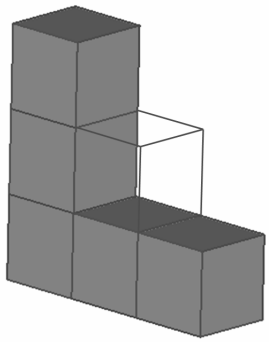
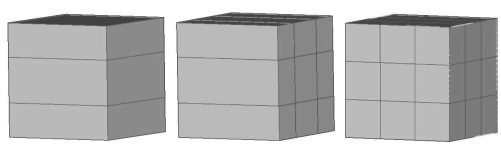


FIG. 2.3. *Example of Concave Region*

2.3. Scalability. Refining a mesh using hex sheets ensures a conformal mesh, however the scalability of using such a process is on the order of n^3. The reason for this is the creation and deletion of intermediate hexes. The process occurs in the following manner. First as shown in Figure 2.4(a), the original hex is deleted and three new hexes are created in its place. Second, the three intermediate hexes are deleted and nine more intermediate hexes are created as in Figure 2.4(b). Finally, the nine intermediate hexes are deleted and 27 final hexes are created as in Figure 2.4(c). In this example, 13 hexes were deleted and 40 hexes were created. This requires significant computational power and time.



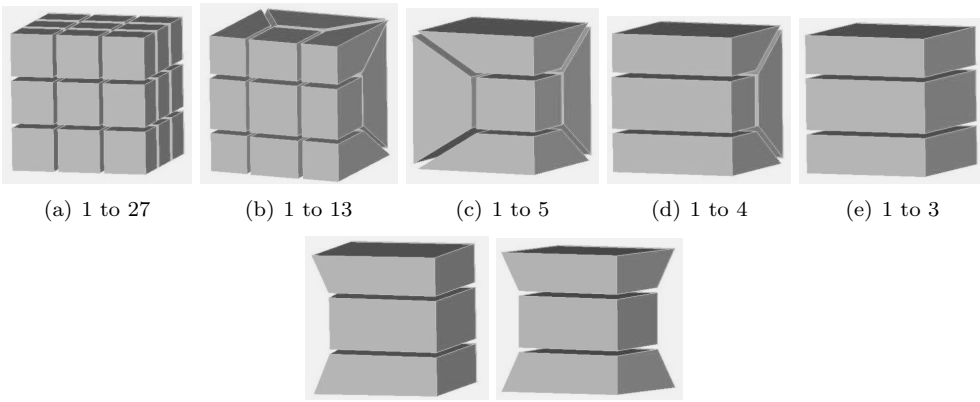
(a) 3 Hexes (b) 9 Hexes (c) 27 Hexes

FIG. 2.4. *Three Phase Process of Old Algorithm*

3. Composite Refinement. Composite Refinement is a new algorithm that attempts to make hex refinement more robust while reducing computational cost. This new algorithm changes the fundamental way in which hexes are processed. Rather than being processed in hex sheets, hexes are processed one at a time. This means that once a hex is processed, it is completely refined and removed from the refinement algorithm. Immediately, this methodology solves the self-intersecting sheet problem and it will be shown later that Composite Refinement also solves the other problems stated in the previous section. It is also hoped that by processing hexes individually, future problems that may arise can be solved with more ease.

Composite Refinement, as its name suggests, includes two distinct processes. These are: 1) Total Hex Refinement and 2) Directional Hex Refinement. The remainder of this section will discuss how these two types of refinement work and how they function together to form the Composite Refinement algorithm.

3.1. Templates. Seven templates are used in Composite Refinement and are shown in the following figure. Figure 3.1(a), Figure 3.1(b), and Figure 3.1(c) are the most common templates used in the refinement process. In fact, any refinement not requiring concavities can be refined using these three templates. All seven templates are used in Total Hex Refinement while Figure 3.1(a) and Figure 3.1(b) are not used in Directional Hex Refinement. Figure 3.1(f) and Figure 3.1(g) are used to handle concavities in the refinement process.



(a) 1 to 27 (b) 1 to 13 (c) 1 to 5 (d) 1 to 4 (e) 1 to 3

(f) 1 to 3 with 1 concavity (g) 1 to 3 with 2 concavities

FIG. 3.1. *Templates Used in Total Hex Refinement*

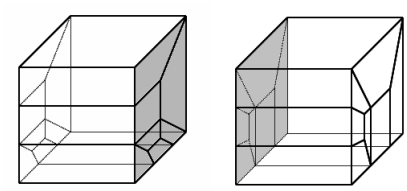
3.2. Total Hex Refinement. Total Hex Refinement refines a hex in one step. The initial hex is deleted and the final hexes are created using one of the seven templates described in the previous subsection. No intermediate hexes are created or deleted thus increasing the efficiency. This type of refinement is preferred and is used as the primary refinement method.

3.3. Directional Hex Refinement. If all meshes were refined using solely Total Hex Refinement, dozens of templates would be required. Most of these required templates are currently unknown or create low quality elements. Therefore, to reduce the number of templates required, Directional Hex Refinement is used. Directional Hex Refinement only requires five templates which are shown above in Figures 3.1(c) to 3.1(g).

Directional Hex Refinement refines along the three principle axes in series. Since hexes are processed individually in Composite Refinement, new techniques had to be developed to maintain a conformal mesh and ensure that refinement occurred correctly. The remainder of this section discusses the advancements made to refine hexes directionally on a hex by hex basis.

3.3.1. The Conformity Problem. Conformity becomes a real problem for Directional Hex Refinement. An example of the conformity problem is shown in Figure 3.2 with two hexes that share a single face. Figure 3.2(a) is refined front to back and then bottom to top. Figure 3.2(b) is refined bottom to top and then front to back. Both contain valid refinement schemes yet the shared face[2] does not match

[2]Shown in gray



(a) Refine Order 1 (b) Refine Order 2

FIG. 3.2. *Conformity Issues*

up. This problem would occur frequently since each hex is refined independently of its neighbors.

3.3.2. Ranking System. To solve the conformity problem, we used the concept of the dual of the mesh discussed previously. This concept was implemented as a ranking system. The ranking system works in the following manner. An initial arbitrary edge is selected and given a rank of 1. All opposite edges of adjacent faces are located for the selected edge. If these new edges are split and need to be directionally refined, they are ranked and become selected edges themselves. The process repeats itself and the ranking system propagates outward. The entire process repeats itself until all applicable edges are ranked. Refinement then occurs on a hex by hex basis starting in the direction with the lowest rank and continuing in ranked order until the hex is completely refined and the algorithm moves onto the next hex.

3.3.3. Projection Scheme. After a hex is refined in a principle direction, new edges exist that may need to be split in order that the refinement works properly. To handle this, we developed a projection scheme to determine which edges need to be split. Figure 3.3 shows a 2D example of a hex that has been refined in one direction. After the refinement has occurred, the projection scheme checks edges 1 and 4. If both edges are split then it will split edges 2 and 3. If only one or no edges are split then edges 2 and 3 will not be split. This process occurs on the four faces that are not parallel to the initial direction of refinement.

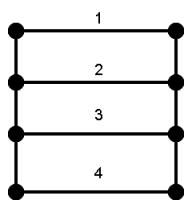
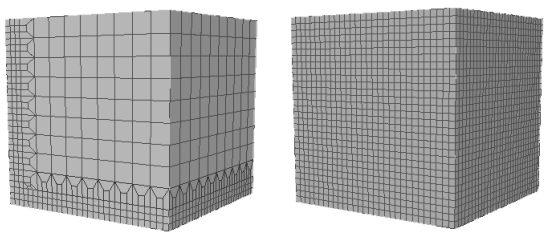


FIG. 3.3. *Projection Scheme*

3.4. Algorithm. The Composite Refinement algorithm starts by applying the the 1 to 27 template to the target hexes as shown in step 3.2. The boundary hexes are all that remain after this step. Because Total Hex Refinement is more efficient, it is applied first in step 3.4. The remaining hexes are then ranked as shown in algorithm step 3.11. Finally, the remaining hexes are refined directionally from lowest to highest rank.

Algorithm 3 Composite Refinement

```
loop target hexes                                   ▷ Total Hex Refinement
    apply 1 to 27 template to target hex
end loop
loop boundary hexes
    if template applies then
        refine hex using template
    else
        add to directional hex list
    end if
end loop
loop directional hex list
    apply ranking system
end loop
loop directional hex list                           ▷ Directional Hex Refinement
    directionally refine hexes
end loop
```



(a) Composite Refinement (b) Old Algorithm

FIG. 4.1. *Comparison of Concavity Capability*

4. Comparison between Composite Refinement and Old Algorithm. Composite Refinement, because it refines individual hexes, solves the self-intersecting sheet problem. Composite Refinement also is able to handle concavities. Figure 4.1(a) and Figure 4.1(b) show the results of both algorithms while trying to refine the bottom and left surfaces of a cube. Composite refinement works correctly while the old algorithm excessively refines the whole cube.

Probably the greatest advantage of Composite Refinement over the old algorithm is scalability. It has already been stated that the scalability of the old algorithm was on the order of n^3. The scalability of Composite Refinement is nearly linear in comparison. Figure 4.2 decisively shows that Composite Refinement out-performs the old algorithm. The old algorithm was unable to refine anything with an initial element count larger than 100000 elements. Contrast this with Composite Refinement which was able to go up to almost half a million initial elements and would go further if more data was taken.

Composite Refinement has become powerful mesh modification tool. While further testing is needed, the data given in this section supports this conclusion. Composite Refinement can handle self-intersecting hex sheets and concavities. The scalability is nearly linear and by modifications to directional refinement theory, Composite Re-

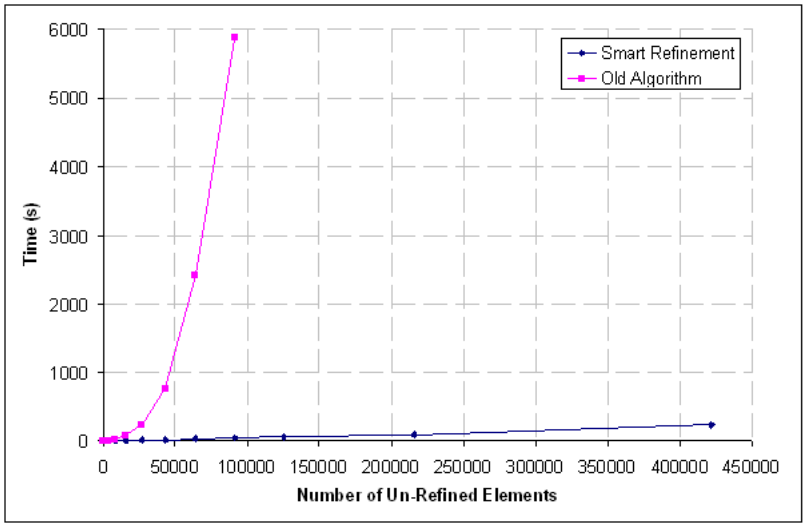


FIG. 4.2. *Scalability comparison*

finement is able to maintain conformity. For these reasons, Composite Refinement should replace the old refinement algorithm.

5. Conclusion. Composite Refinement was developed to solve problems with old theory and implementation. It can handle self-intersecting hex sheets and concavities. The scalability of Composite Refinement is nearly linear which is a dramatic improvement over cubic scalability found in the old algorithm. With its increased capability and scalability, Composite Refinement is a powerful mesh modification tool. Composite Refinement will allow increased accuracy in the analysis of target areas.

REFERENCES

[1] N. HARRIS, *Conformal Refinement of All-Hexahedral Finite Element Meshes*, M.S. thesis, Brigham Young University, Provo, UT, 2004.

[2] P. MURDOCH, AND S. BENZLEY, *The spatial twist continuum*, in Proceedings, 4th International Meshing Roundtable, pp. 243–252.

[3] S. BENZLEY, N. HARRIS, M. SCOTT, M. BORDEN, AND S. OWEN, *Conformal Refinement and Coarsening of Unstructured Hexahedral Meshes*, Journal of Computing and Information Science in Engineering, 5 (2005), pp. 330–337.

USING RECONFIGURABLE FUNCTIONAL UNITS WITH SANDIA SCIENTIFIC APPLICATIONS

K. RUPNOW[‡] AND K. UNDERWOOD[§]

Abstract. Several groups have proposed using Reconfigurable Functional Units (RFUs) to accelerate standard programs and embedded media applications. However, previous work has demonstrated the significant differences between Sandia's scientific applications and scientific benchmarks such as those used in previous RFU studies. Furthermore, past RFU's have significantly limited abilities compared to the needs of Sandia applications. In our study, we select patterns of common integer computation (dataflow graphs) and accelerate them in an RFU. We use application traces and execution based simulation to simulate potential application speedup using an RFU. We present data demonstrating the potential acceleration of Sandia applications using a Reconfigurable Functional Unit to accelerate the integer computation.

1. Introduction. Many groups have utilized Reconfigurable Functional Units (RFUs), however they have concentrated on embedded applications and media streaming applications. RFU designs for embedded and streaming applications target significantly different program behavior than that seen in scientific applications. Interlock collapsing [14] only considers graphs with two nodes. Other techniques such as the CCA [4] and Dataflow Mini-Graphs [1] relax the limitation of two instructions, but still limit the inputs and outputs of the graph. In addition, the CCA limits implementable graphs to the subset that can be placed in their static RFU placement. While these techniques have shown significant speedup with their designs, previous work [16] has shown that dataflow in Sandia's scientific applications is significantly more complex than dataflow in SPEC-FP. Sandia's dataflow graphs average more instructions, more inputs and outputs and larger topographical height and width. Therefore, for our architecture we relax the topographical shape constraints and the input and output constraints, however we still place a premium on building a realistic RFU in terms of register file bandwidth, instruction encoding, and chip area.

Reconfigurable Functional Units provide two main benefits: increased issue logic efficiency and graph acceleration. Issue logic increases in efficiency because of the more efficient instruction encoding an RFU can achieve. One RFU instruction may represent many instructions, thus when an RFU instruction is issued we are effectively issuing many instructions, possibly in excess of the processor's issue width. Furthermore, because one RFU instruction may represent many instructions, there is extra issue window space, which increases the effective issue window size. Graphs can be accelerated because an RFU can contain closely coupled functional units and configurable communication patterns. By encoding the communication pattern into the RFU, we prevent the need for bypass logic. Furthermore, because certain computations such as logical operations complete in less than one cycle, we may combine several such operations into one cycle. For this work we separate the acceleration due to issue logic efficiency from the graph acceleration. We will assume zero graph acceleration and measure application speedup based solely on increased issue logic efficiency. For our simulations, we select graphs that cover 1% or more of the integer computation instructions that are contained by any graph. We then use the selected graphs as our set of configurations for the RFU. In Sandia applications, the top 16 graphs cover 89% of the integer computations contained by a graph, while the top

[‡]University of Wisconsin–Madison
[§]Sandia National Laboratories

16 graphs cover 84% of integer computations in SPEC-FP. Using the selected graphs, most applications achieve speedup of between 2% and 6% over the baseline.

2. Related Work. Dataflow research originated in the 1960's in studies of graph theoretical models of parallel computation by Karp and Miller [12] and Rodriguez [15]. These papers studied parallel computation models as an alternative to the von Neumann model of sequential execution. The main goal of parallel computation models is to extract dataflow locality, which is the recurrence of common computation and communication patterns. Dataflow is expressed as a graph, where nodes are instructions and edges are the communication between instructions. Note that dataflow architectures have no program counter, and thus no branching behavior. All decisions that would be made by branches in a standard microprocessor are expressed by processing units that conditionally send data to other processing units.

The first dataflow architectures that appeared in the early 1970's [5, 6] correspond closely to the graph model; the hardware consisted of functional units that communicate directly using FIFO queues. Dataflow architecture designs of the 1980's [8, 10, 13, 17] built on early designs by adding tagging schemes, dynamic instruction allocation, and communication hierarchy. However, the designs still required dataflow specific programming languages and the memory and silicon integration technology of the time struggled to provide the machines with sufficient memory bandwidth.

More recently, dataflow architectures have experienced a resurgence with architectures such as WaveScalar [19], RAW [20], and TRIPS [18]. However, all of these dataflow architectures still depend on an explicitly parallel computing paradigm. Despite the improvements to the capabilities of dataflow processors, traditional microprocessors are still the dominant technology. Furthermore, there is a significant amount of dataflow locality in von Neumann control-flow architectures, and Reconfigurable Functional Units (RFUs) are poised to extract that dataflow locality. RFUs execute common sequences of instructions that were detected either at compile time or dynamically. Logically, this execution model extracts the dataflow locality of programs that were compiled for sequential execution. Many groups have suggested RFUs and corresponding execution models. Chimera [11], and the CCA [4] are two examples of RFUs that identify common sequences of instructions during compile time. Other projects [2, 7] chose to perform similar instruction sequence profiling, but they create application specific processing units rather than configurations for a RFU.

In contrast to previous RFU designs, we relax several graph constraints such as graph depth, number of inputs and number of outputs, while still requiring a realistic design. Our previous work demonstrated the importance of handling more complex dataflow graphs, which motivated the relaxed constraints as compared to the CCA [4] or Dataflow Mini-Graphs [1]. In addition, we chose an RFU tightly coupled with the microprocessor pipeline, rather than the co-processor model used in Chimera [11].

3. Method. For all of our tests, we use trace based execution of G5 PPC [9] trace files. Our analysis program uses the SimpleScalar [3] based definition file and decode structure. We use 14 traces covering programs from the SPEC-FP 2000 benchmarking suite, and 16 traces from seven Sandia applications (ALEGRA, CTH, cube3, ITS, LAMMPS, MPSalsa, and Xyce). Each trace is a representative four billion instructions selected using performance register profiling and source code analysis. Our simulation is performed in two phases, which simulates a simplified form of compiler based graph selection. The trace-based graph selection algorithm uses the first one billion instructions from the trace. The trace-based execution and timing simulation uses the entire four-billion instruction trace.

3.1. Graph Selection. For the graph selection phase of our simulations, we select static dataflow graphs in a method similar to compiler based approaches. We limit dataflow graphs to instructions within the same basic block of the program. If all of a dataflow graph's component instructions are within the same basic block, then we can ensure that whenever that graph is encountered it is valid to execute the entire graph. This simplifies pipeline cleanup in the event of a branch misprediction. For the first phase we construct dataflow graphs to represent the complete set of dataflow that does not cross branch boundaries. We keep track of all unique graphs, and count the number of executions of each unique graph, as well as the total integer computation instructions. For each simulation, we select graphs that contribute 1% or more of the total integer computations executed as part of a DFG to use as our pre-selected list of graphs for the timing simulation. This technique limits the total number of dataflow graphs handled by the Reconfigurable Functional Unit, which limits the expected complexity of maintaining multiple configuration contexts.

3.2. RFU Architecture. For the architecture of our Reconfigurable Functional Unit, we have selected an extremely conservative design. This design does not assume that the DFG is accelerated any as compared to a processor with sufficient free resources. For this reason, any application acceleration we see in the timing analysis is due to the increased virtual issue width and virtual issue window size. The issue width and issue window size are effectively increased because one RFU instruction can represent many instructions. Consequently, issuing an RFU instruction effectively issues many instructions, leaving the remaining issue width available for other independent operations. Additionally, since the RFU instruction takes fewer entries in the issue queue, there are additional queue entries available for other instructions; this increases the effective issue window size. To limit both register file bandwidth requirements, and instruction word encoding requirements, we limit the amount of data one RFU opcode can contain to three input registers and one output register, or two output registers. Based on this limit, one dataflow graph will be encoded in one or more RFU opcodes. The depth of the graph determines graph execution latency. When we encounter a DFG during program execution, the graph is split into one or more RFU opcodes to encode all inputs and outputs based on the previous requirements. Inputs are sorted into the order in which they will be required during graph execution, the RFU can make intermediate progress on the graph execution as the inputs become available (inputs may only become available in groups associated with the opcode they were encoded in). We then remove the original instructions from the instruction stream, insert the RFU instructions, and reorder the instruction stream as necessary to enforce register dependence ordering.

4. Graph Selection Analysis. This section presents analysis of the graphs selected during the first phase of simulation. We first examine the average dataflow graph size, comparing the SPEC-FP benchmark suite to the scientific applications used at Sandia. In addition, we compare the average graph size under two constraints, minimum graph size of two instructions, and minimum graph size of three instructions. We then compare the average percentage of integer computation instructions that are contained in a dataflow graph, and conclude by examining the percentage of integer computation instructions contained in a dataflow graph versus the number of selected graphs.

4.1. Average Graph Size. On average, the dataflow graphs in Sandia's applications are larger than those in the SPEC-FP application. For graphs with a

minimum of two nodes, Sandia averages 40% larger graphs, while for graphs with a minimum of three nodes, Sandia averages 33% larger graphs. For SPEC-FP, including graphs of size two reduces the average number of nodes by 41%, while for Sandia including graphs of size two reduces the average by 38%. In Figure 4.1, we see the average graph size for both SPEC-FP and Sandia for each of the minimum graph sizes. Sandia's larger average graph size is further emphasized because graphs from Sandia applications also cover more of the total integer computation instructions on average.

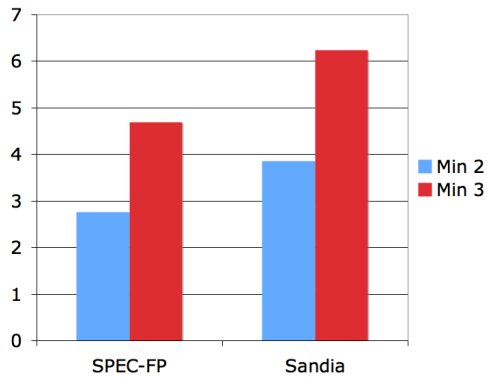


FIG. 4.1. *Average Dataflow Graph Size*

4.2. Average Integer Instruction Coverage. On average graphs from Sandia applications cover a larger percentage of the integer computation instructions, however the difference is more exaggerated when the minimum graph size is three nodes. In Figure 4.2, we see the average coverage of SPEC-FP and Sandia for both a minimum graph size of two and minimum graph size of three. As expected, the average coverage is higher when graphs of size two are included. Interestingly, when the minimum graph size is three, Sandia covers 9% more instructions on average while the addition of size two graphs lowers that advantage to only 2%. In addition, average coverage in SPEC-FP doubles when size two graphs are included while Sandia only covers 40% more instructions. This indicates a prevalence of important two instruction graphs in SPEC-FP applications to a degree not seen in Sandia applications. While total coverage is important, it is more important to examine how much of the coverage can be obtained with a small number of unique graphs.

4.3. Integer Instruction Coverage vs. Number of Selected Graphs. In Figure 4.3, we see the average integer computation coverage versus the number of selected graphs. For both SPEC-FP and Sandia, most of the obtainable instruction coverage is achieved with the top ten graphs, with especially high coverage obtained with the top five graphs. In the case with a minimum graph size of three, Sandia has uniformly higher coverage throughout the range, however in the case with a minimum graph size of two, Sandia's coverage only overtakes SPEC-FP's coverage with more than five graphs. For SPEC-FP, we see that the minimum of two starts at a significantly higher coverage, which indicates that SPEC-FP contains a very important two instruction graph on average. Instruction coverage in Sandia applications, however, starts at a level very similar to the minimum of three and rises more rapidly after the first graph. This indicates that while Sandia does contain important two instruction

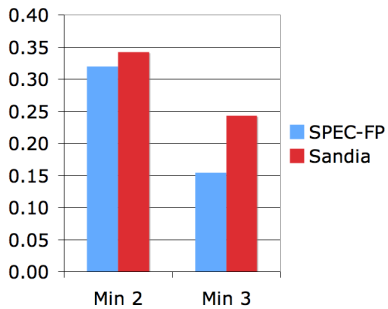


FIG. 4.2. *Average Integer Computation Instruction Coverage*

graphs, they are of lesser importance than the larger graphs on average. Overall, it is encouraging that most of the potential dataflow graph executions can be covered by a small number of unique dataflow graphs. This shows that while we need configurability to capture all of the graphs, a small number of different graphs can capture most of the potential.

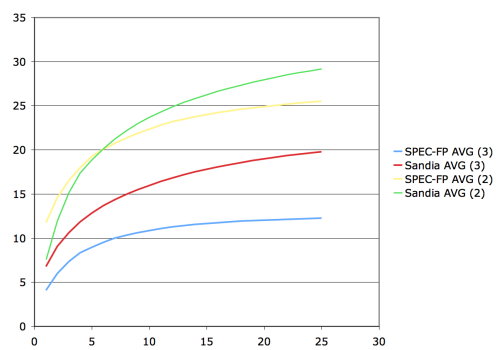


FIG. 4.3. *Integer Computation Coverage vs. Number of Selected Graphs*

5. Graph Execution Analysis. This section presents analysis of the execution and timing simulation of our RFU architecture. We begin by examining the speedup of our architecture over the baseline case. We then examine the number of RFU operations that executed as compared to the number we would expect from the selection, and conclude by showing the average number of RFU operations per graph.

5.1. Percent Speedup. For both sets of selected graphs, most of the benchmarks or applications achieve a speedup in the range of 2%-6%. Note again that this speedup is solely based on the increased efficiency of issue logic, as the graph is not accelerated separately. This demonstrates that the grouping of common calculations into dataflow graphs can help the issue logic more efficiently handle the instruction stream. There are a few benchmarks which show a slowdown of 1.5%-2%. This is caused by the introduction of a serialization case that was not in the original code. In the event that a graph has multiple outputs, it is possible that the additional latency to wait for the entire graph to complete causes instructions waiting for that output to wait longer than they would otherwise. To mitigate the effect of this problem,

we will be investigating allowing the RFU to output intermediate data as it becomes available, which is a more realistic implementation. In Figure 5.1 and Figure 5.2, we see the speedup over baseline for both sets of selected graphs for the SPEC-FP benchmarks and Sandia applications, respectively.

The results for LAMMPS particularly emphasize the serialization problem in the current architecture. In lmp.flow.langevin, there are 175 Million graphs found and executed during the trace, which causes a 3.3% speedup over the base case. In lmp.lj, however, there are 200 Million graphs executed which causes a slow down of 8%. In both cases there are sufficient graphs to achieve significant speedup, however the serialization problems in lmp.lj cause a slowdown where a speedup would be expected. Similarly, we see serialization issues in many of the applications, where the minimum of 2 simulation is slower that the minimum of 3. This should never be the case except as caused by serialization.

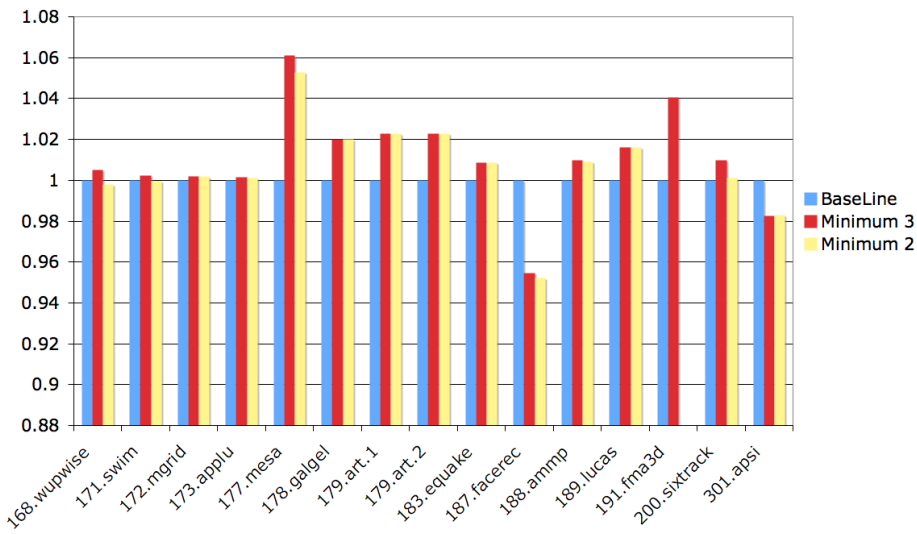


FIG. 5.1. *Speedup: SPEC-FP Benchmarks*

5.2. Average Number of RFU ops per Graph. We measure the average number of RFU ops per graph to track how average graph size affects the acceleration potential of our applications. As previously stated, one dataflow graph may require one or more RFU ops to encode all of the input and output registers. Larger graphs will require more RFU ops, but they also encode a larger number of instructions. On average, there are 2.25 RFU ops per graph with a minimum graph size of three, and 1.9 RFU ops per graph with a minimum graph size of two. The highest average for any benchmark or application is 3. When we compare this to the average graph size for the executed graphs, we see a direct correspondence between the graph size and the required number of RFU ops, as expected. We will be examining the packing efficiency of graph data into the RFU ops to determine if our current encoding scheme is the most efficient choice to encode RFU ops. We wish to minimize the number of required RFU ops per graphs while maximizing the number of instructions per graph; fewer RFU ops per graph will achieve better speedup.

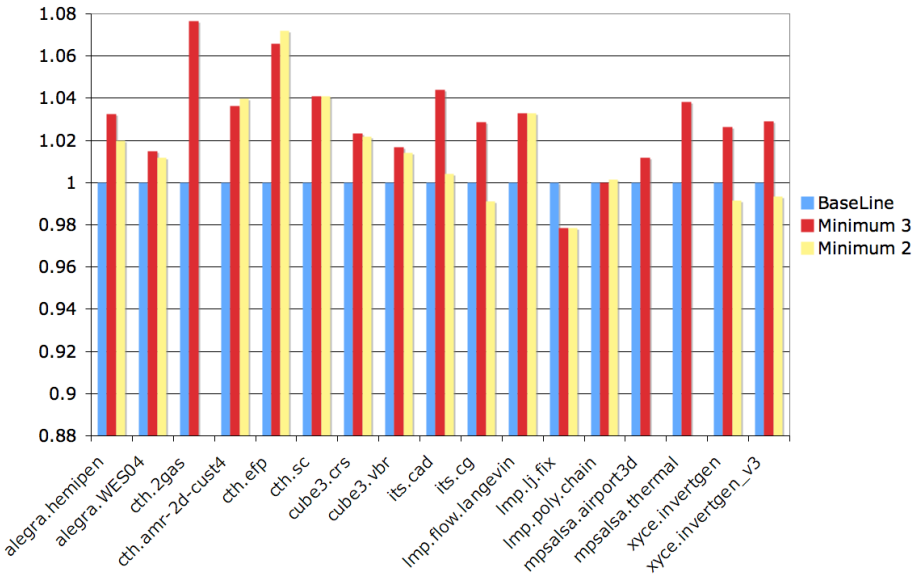


FIG. 5.2. *Speedup: Sandia Applications*

6. Conclusions. There is significant potential in using a Reconfigurable Functional Unit with scientific applications at Sandia. Even with a conservative RFU architecture, we have achieved modest speedups simply by increasing the efficiency of the issue logic. Furthermore, future schemes will improve on these results by mitigating serialization impact and minimizing the number of required RFU operations to encode one dataflow graph. Future RFU designs can contain realistic graph acceleration values, which would further improve the speedup numbers. In addition, techniques to combine dataflow graphs from multiple basic blocks could improve the scope and coverage of this technique.

REFERENCES

[1] ANNE BRACY, PRASHANT PRAHLAD, AND AMIR ROTH, *Dataflow mini-graphs: Amplifying superscalar capacity and bandwidth*, in MICRO 37: Proceedings of the 37th annual International Symposium on Microarchitecture, IEEE Computer Society, 2004, pp. 18–29.

[2] PHILIP BRISK, ADAM KAPLAN, RYAN KASTNER, AND MAJID SARRAFZADEH, *Instruction generation and regularity extraction for reconfigurable processors*, in CASES '02: Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems, ACM Press, 2002, pp. 262–269.

[3] D. C. BURGER AND T. M. AUSTIN, *The simplescalar tool set, version 2.0*, Tech. Report CS-TR-97-1342, 1997 1997.

[4] NATHAN CLARK, JASON BLOME, MICHAEL CHU, SCOTT MAHLKE, STUART BILES, AND KRISZTIAN FLAUTNER, *An architecture framework for transparent instruction set customization in embedded processors*, ISCA, 1 (2005), pp. 0–12.

[5] A. L. DAVIS, *The architecture and system method of ddm1: A recursively structured data driven machine*, in ISCA '78: Proceedings of the 5th annual symposium on Computer architecture, ACM Press, 1978, pp. 210–215.

[6] JACK B. DENNIS AND DAVID P. MISUNAS, *A preliminary architecture for a basic data-flow processor*, in ISCA '75: Proceedings of the 2nd annual symposium on Computer architecture, ACM Press, 1975, pp. 126–132.

[7] DAVID GOODWIN AND DARIN PETKOV, *Automatic generation of application specific processors*,

in CASES '03: Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems, ACM Press, 2003, pp. 137–147.

[8] V. G. Grafe, G. S. Davidson, J. E. Hoch, and V. P. Holmes, *The epsilon dataflow processor*, in ISCA '89: Proceedings of the 16th annual international symposium on Computer architecture, ACM Press, 1989, pp. 36–45.

[9] Apple Architecture Performance Groups, *Computer Hardware Understanding Development Tools 2.0 Reference Guide for MacOS X*, Apple Computer Inc, July 2002.

[10] J. R. Gurd, C. C. Kirkham, and I. Watson, *The manchester prototype dataflow computer*, Communications of the ACM, 28 (1985), pp. 34–52.

[11] S. Hauck, T. W. Fry, M. M. Hosler, and J. P. Kao, *The chimaera reconfigurable functional unit*, in Proceedings of the 5th IEEE Symposium on FPGA-Based Custom Computing Machines (FCCM '97), IEEE Computer Society, 1997, p. 87.

[12] R. M. Karp, R. E. Miller, and S. Winograd, *Properties of a model for parallel computations: Determinacy, termination, queueing*, SIAM J. Appl. Math, 14 (1966), pp. 1390–1411.

[13] Masasuke Kishi, Hiroshi Yasuhara, and Yasusuke Kawamura, *Dddp-a distributed data driven processor*, SIGARCH Comput.Archit.News, 11 (1983), pp. 236–242.

[14] Nadeem Malik, Richard J. Eickemeyer, and Stamatis Vassiliadis, *Interlock collapsing alu for increased instruction-level parallelism*, in MICRO 25: Proceedings of the 25th annual international symposium on Microarchitecture, IEEE Computer Society Press, 1992, pp. 149–157.

[15] J. E. Rodriguez, *A graph model for parallel computation*, Sept. 1967 1967.

[16] Kyle Rupnow, Arun Rodrigues, Keith Underwood, and Katherine Compton, *Scientific applications vs. spec-fp: A comparison of program behavior*, in ICS06: Proceedings of the IEEE/ACM International Conference on Supercomputing, IEEE Computer Society, 2006.

[17] S. Sakai, y. Yamaguchi, K. Hiraki, Y. Kodama, and T. Yuba, *An architecture of a dataflow single chip processor*, in ISCA '89: Proceedings of the 16th annual international symposium on Computer architecture, ACM Press, 1989, pp. 46–53.

[18] Karthikeyan Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh, Nitya Ranganathan, Doug Burger, Stephen W. Keckler, Robert G. McDonald, and Charles R. Moore, *Trips: A polymorphous architecture for exploiting ilp, tlp, and dlp*, ACM Trans.Archit.Code Optim., 1 (2004), pp. 62–93.

[19] Steven Swanson, Ken Michelson, Andrew Schwerin, and Mark Oskin, *Wavescalar*, in MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture, IEEE Computer Society, 2003, p. 291.

[20] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, Jae-Wook Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal, *The raw microprocessor: a computational fabric for software circuits and general-purpose programs*, Micro, IEEE, 22 (2002), pp. 25–35.

MASSIVELY MULTITHREADED APPLICATIONS FOR DEEPLY PIPELINED QCA ARCHITECTURES

MEGAN VANCE* AND KEITH UNDERWOOD†

Abstract. To achieve good processor utilization in a deeply pipelined nanoscale architecture, traditional serial application codes will not be sufficient. This study investigates the parallelization of a massively multithreaded depth first search. With straightforward techniques and a small future computation library, a DFS search algorithm is shown to provide thousands of independent lightweight threads when executed on a simple multithreaded processor.

1. Introduction. As shrinking transistor size becomes increasingly problematic to the design of future CMOS-based processing devices, several design methods have come to the fore as possible alternatives for computing at the nano-scale. In particular, nanotechnologies aim to overcome the problems of heat dissipation, high fabrication costs, and fundamental limits to the density of CMOS transistors. Though there are several different nanotechnologies currently being investigated, each of them presents the similar challenge to software design, in the form of higher logic density and increased memory latencies. For example, in CMOS, higher latencies result from extremely small, high resistance wires.

Increasing memory latency exacerbates the von Neumann bottleneck extant in current architectures, where processing and logic are physically separated. Tolerating these latencies enough to provide good performance will not be possible with the current paradigm of serial or coarse grained parallel computations. Neither of these programming models can provide enough independent instructions to overcome latencies which may be in the tens of thousands of cycles.

A particular technology, quantum-dot cellular automata (QCA), is used here as the basis for a nano-scale architecture. QCA lends itself to a mode of computation distributed throughout the memory structure, addressing the processor-memory bottleneck which limits current processor performance. In addition, QCA design allows for computation to proceed "in the wire", meaning that calculations can be completed while data is in transit. This naturally lends itself to processors with very deep pipelines. While current processors rely primarily on instruction re-ordering of serial codes to fill up pipeline issue slots, dependence on Instruction Level Parallelism (ILP) will not be sufficient to efficiently utilize processing power on the nano level where pipelines will be deeper and memory latencies longer. To achieve good utilization, applications may be designed to exploit Thread Level Parallelism (TLP) on a massive scale.

Creating massive numbers of threads, on the order of tens of thousands, is feasible when the threads themselves require very little overhead for creation, movement, and destruction. Current investigations into large-scale multithreading achieve this through the paradigm of lightweight threads. As opposed to traditional kernel or user level threads in a UNIX-type system, lightweight threads need very little state – minimally a program counter, some local data, and status information. Since thread state is small, thread creation can be as fast as the allocation and copy of a small amount of data. By associating register data with the thread state, movement becomes a small memory copy.

*University of Notre Dame
†Sandia National Laboratories

Graph algorithms provide a good target for both large-scale multithreading and novel architectures. Most importantly, graph algorithms are increasingly of interest for a large number of real world applications, particularly data mining. Searching extremely large graphs also lends itself particularly well to decomposition into independent threads, since searches often consist of code which visits each node in the graph, examines the node, and possibly makes some change to it. Visiting each node may typically proceed in parallel, with serialization only required for concurrent access to a single node. Because a visit is typically a short operation, requiring only enough local data to identify the node to visit, this operation maps well to a lightweight thread.

Though graph algorithms provide useful functionality, their performance on traditional architectures has been poor. This is due to the hierarchical memory system designs which favor ordered, consecutive memory accesses. Since graph algorithms proceed by traversing edges which may connect any set of nodes, memory accesses appear random to the memory system, which cannot compensate for long latencies by caching recent accesses or prefetching nearby data. Using a novel architectural approach with a thread based execution model obviates this typically bad behavior.

1.1. QCA. Quantum-dot cellular automata [3] technology is based on the activity of two electrons arranged in a cell with four dots arranged in a square. Two excess electrons are attached to the cell and may move between the four dots, meaning that two stable configurations exist as shown in figure 1.1. Either the electrons repel each other toward the top-left/bottom right configuration, connotating a binary 0, or the opposite, a binary 1. Wires are formed by placing cells in close proximity and preventing a cell's electrons from tunneling so that they may influence the state of a neighbor cell, who's electrons are allowed to tunnel.

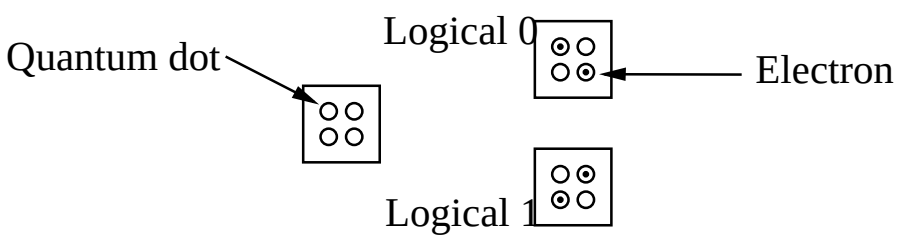


FIG. 1.1. *Quantum-dot Cell*

Clocking operates, then, by raising and lowering an electric field to either lock a cell into a stable configuration, or allow it to take on the configuration of a neighbor which is stable. As the field rises, the cell becomes more stable, so during an upward "clock" pulse, the cell takes on a configuration and holds it until the field begins to drop, at which point the cell looses its configuration altogether until the next rise. Cell state acquiring value, storing value, dropping value, and lacking value are called respectively switch, hold, release, and relax as shown in figure 1.2.

Utilizing the clocking fields in the appropriate configuration yields the basic computation block for QCA, the majority gate. A central cell is influenced by three neighbors (inputs) into taking on the value held by the majority. Then, this central cell acts as a drive to a fourth neighbor (output). Fixing one input to 1 produces an OR gate, and to 0, an AND gate. Figure 1.3 shows the majority gate with inputs A, B, and C giving output AB+BC+AC.

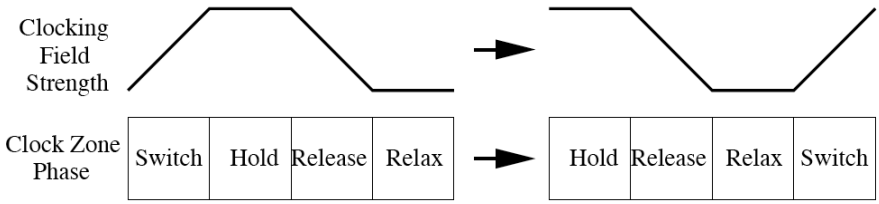


FIG. 1.2. *QCA Clock Phases: switch, hold, release, relax*

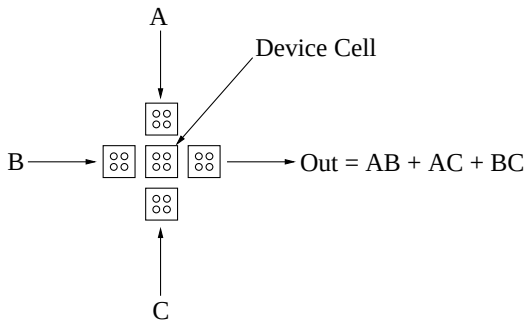


FIG. 1.3. *QCA Majority Gate*

2. Previous Work. A thread based execution model based on QCA technology is presented in [7]. Because the memory structure is deeply pipelined and dense, moving instructions and data to a single execution location would be inefficient. Instead, thread state is encapsulated into a small bundle which moves throughout memory between instructions and data. As the thread is routed to its next destination, computation is performed in the routers which are spread throughout memory, so individual pipeline stages complete different routers as the thread moves.

In contrast to traditional processors, the Tera/Cray MTA [4] [1] does not rely on ILP to hide latencies. Applications written for the MTA [8] [2] contain massive numbers of independent control streams. This is accomplished by parallelizing loops and recursive structures at a fine-grain. Recursive structures are managed via a runtime system which stores thread futures in a queueing system which is accessed by available execution resources. The approach is similar in this study, except that threads are created as software entities rather than allocated as hardware resources, which simplifies the implementation.

Exposing the inherent abilities of graph algorithms to a massively multithreaded architecture has been accomplished via a MultiThreaded Graph Library (MTGL) [6]. MTGL draws inspiration the Boost Graph Library (BoostGL) [5] but was not designed as an extension, allowing design to focus on MTA performance rather than genericity. This study adopts the multithreaded approach to graph algorithms afforded by the MTA language constructions in MTGL.

3. Methodology. In order to study the behavior of a massively multithreaded DFS without over-reliance on a particular microarchitecture, a structural simulation framework was used to model system behavior as the high level interaction of components. Programs are compiled for a PowerPC target and interact through a middle

layer of software to a set of C++ objects which interpret computation and memory operations as a series of tokens which make changes to the state of the system. Therefore the same binary can be used to observe performance on a conventional processor or a set of simple integrated processor and memory nodes. Thread creation is supported as fast (3-5 instruction) calls to the simulator internals.

Software described here was written in C++ and compiled using g++ version 3.3. No optimization was used for the future library itself, while the actual DFS search algorithm was compile with the `-O2` flag.

3.1. Futures Library. Derived from the basic model presented in the MTA programming environment, futures are a method to parallelize loops which contain function calls. Figure 3.1 shows how a future would be created in the MTA compiler. Rather than immediately create a thread to run each new instance of rfoo, a structure called a future is created.

```
void rfoo (ArgClass arg) {

  #pragma mta assert parallel
  #pragma mta loop future
  for (int i = 0; i < totalIterations; i++)
     rfoo (args.moreArgs[i])
}
```

FIG. 3.1. *Loop Futures in the MTA*

Essentially a future stores all of the relevant information needed to call the function. Since this consists of only a function pointer and arguments, the future itself can be small. Futures are enqueued into a FIFO queue from which other threads, called virtual processors (VPs), can dequeue futures and instantiate a new thread by creating the necessary memory space for its stack and assigning it to a processor. Stack space may not be necessary if the thread is short and uses few local values.

```
void rfoo (ArgClass args) {
  Future *ftr;
  int iterationsPerThread;

  int i = 0;
  while (i < totalIterations) {
     ftr = Future<void,ArgClass>::create(NULL,
                                         &rfoo,
                                         &(args.moreArgs[i]),
                                         iterationsPerThread);
     futureQ_addWork(ftr);
     i += iterationsPerThread
  }

  futureQ_wait();
}
```

FIG. 3.2. *Loop Futures*

To provide the same kind of functionality as the `#pragma mta` commands, a

futures library was written which is invoked as shown in figure 3.2. Because the MTA compiler contains advanced parallelization optimizations, work is done behind the scenes which is made explicit in the calls to the future library. Because `totalIterations` may be extremely large compared to the number of threads which the programmer wishes to create, multiple iterations are assigned to each future. This happens when the `Future` object is created. In order, the arguments to future creation are:

1. a pointer where an array of return values from the iterations may be stored
2. a pointer to the function to be called
3. an array of arguments, where each argument will be used in a separate call to the function
4. the number of calls to be made, serially, to the function in the created thread

As the basic structure of the graph search is recursive, so the threads which are taken from the future queue may create more futures. By convention, when a thread creates a set of futures in a loop, it will not continue in its computation until all of these futures return. Whereas the MTA handles this action by detecting a blocked stream and re-allocating its resource, the futures library has the parent thread block explicitly using the `futureQ_wait` library call, at which point it is removed from its virtual processor.

At any point in computation, there are at most two types of threads available for a free virtual processor to choose for execution, future threads, and threads which have run for a while, blocked, and are now unblocked. The latter type will always have priority over the former and are stored in a separate queue, call the unblocked queue. If no threads exist, the virtual processor thread will block. When a thread unblocks or is added as a future, a blocked VP will be unblocked, if such a VP exists. All blocking and unblocking incurs very little overhead, since these functions are handled via full/empty bits on a single memory location.

3.2. DFS Search Algorithm. For this study, a DFS search was constructed which uses the graphs generated in MTGL and proceeds through a DFS search with similar logic. Rather than use a DFS as a basis for another algorithm, such as connected component detection, this search simply traverses the graph, marking each node it encounters.

For a non-marked node, the algorithm marks the node and proceeds to visit each node to which it is connected by an edge. If the node has above a static threshold of neighbors, 20 in this case, multiple threads are spawned to visit the neighbors, at a granularity of 8 visits per thread. Dropping the threshold or granularity would create more threads per node visit.

3.3. Configuration. Execution was modeled on a single multithreaded processor. Each issue slot in the pipeline must contain an instruction from a different thread. Though this may not be a requirement for a particular multithreaded processor, it simplifies processor logic, obviating the need for hazard detection or branch prediction logic. Simpler design suits nanoscale devices as well as any logic which is to be repeated a large number of times on a single chip.

To model the large memory bandwidth expected from closely coupling computation and memory access, the simulated processor was directly connected to 2 DRAM modules which contained the graph data. By avoiding a bottleneck to memory, the application benefits from many threads which may have long latency delays but will not suffer from constricted bandwidth.

4. Results and Future Work. With the threshold and granularity numbers given above, a DFS search on a graph with 2M vertices 30M Edges produced 440K threads during execution. As simulator development continues, these benchmarks will be used to compare performance in a deeply pipelined conventional architecture versus a deeply pipelined multithreaded processor described in section 3.3. Also, preliminary testing was done on an MTGL connected components search on a smaller graph of 256 vertices and 65K edges, from which it was possible to derive 35K threads by reducing the number of node visits per thread to 2.

Once a fully functional combination of MTGL and the future library is available for simulation, the performance of several algorithms can be investigated across a range of architectural configurations and pipeline depths.

5. Conclusion. In a QCA architecture, real applications will require large numbers of independent instructions to hide memory latencies. Here, we have shown that a lightweight library can provide thousands of threads with only minimal changes to the existing code. Using useful graph algorithms, we have simulated the kind of massive lightweight parallelism required by the bouncing threads model of QCA execution.

REFERENCES

[1] Robert Alverson, David Callahan, Daniel Cummings, Brian Koblenz, Allan Porterfield, and Burton Smith, *The Tera computer system*, in Proceedings of the1990 International Conference on Supercomputing, 1990, pp. 1–6.

[2] J. Boisseau, L. Carter, K. Gatlin, A. Majumdar, and A. Snavely, *NAS benchmarks on the Tera MTA*, 1998.

[3] C.S. Lent and P.D. Tougaw, *A Device Architecture for Computing with Quantum Dots*, in Proceedings of the IEEE, vol. 85, 1997, pp. 541–557.

[4] Gail Alverson and Preston Briggs and Susan Coatney and Simon Kahan and Richard Korry, *Tera hardware-software cooperation*, in Supercomputing '97: Proceedings of the 1997 ACM/IEEE conference on Supercomputing , New York, NY, USA, 1997, ACM Press, pp. 1–16.

[5] J. Siek and L-Q. Lee, and A. Lumsdaine., *The Boost Graph Library*, Addison-Wesley, 2002.

[6] Jonathan Berry and Bruce Hendrickson and Simon Kahan and Petr Konecny, *Graph Software Development and Performance on the MTA-2 and Eldorado*, in Cray Users Group Conference (CUG), 2006.

[7] Sarah E. Frost, Arun F. Rodrigues, Charles A. Giefer, Peter M. Kogge, *Bouncing Threads: Merging a new execution model into a nanotechnology memory*, in Proceedings. IEEE Computer Society Annual Symposium on VLSI, February 2003, pp. 19–25.

[8] Christian Stork, *Exploring the Tera MTA by Example*. 2000.