SANDIA REPORT

SAND2007-7146
Unlimited Release
Printed August 26, 2008

Final Report on LDRD Project: Coupling Strategies for Multi-Physics Applications

Russell Hooper, Matt Hopkins, Roger Pawlowski, Brian Carnes, Harry K. Moffat

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

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

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

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.osti.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/help/ordermethods.asp?loc=7-4-0#online

SAND2007-7146
Unlimited Release
Printed August 26, 2008

Final Report on LDRD Project: Coupling Strategies for Multi-Physics Applications

Russell Hooper, Matt Hopkins, Roger Pawlowski, Brian Carnes, Harry K. Moffat
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185

Abstract

Many current and future modeling applications at Sandia including ASC milestones will critically depend on the simultaneous solution of vastly different physical phenomena.

Issues due to code coupling are often not addressed, understood, or even recognized.

The objectives of the LDRD has been both in theory and in code development. We will show that we have provided a fundamental analysis of coupling, i.e., when strong coupling vs. a successive substitution strategy is needed. We have enabled the implementation of tighter coupling strategies through additions to the NOX and Sierra code suites to make coupling strategies available now. We have leveraged existing functionality to do this. Specifically, we have built into NOX the capability to handle fully coupled simulations from multiple codes, and we have also built into NOX the capability to handle Jacobi Free Newton Krylov simulations that link multiple applications. We show how this capability may be accessed from within the Sierra Framework as well as from outside of Sierra.

The critical impact from this LDRD is that we have shown how and have delivered strategies for enabling strong Newton-based coupling while respecting the modularity of existing codes. This will facilitate the use of these codes in a coupled manner to solve multi-physic applications.

Acknowledgment

We would like to acknowledge Thomas M. Smith for his invaluable help in getting Premo to couple to Calore to enable the Premcal Sierra application.

Contents

Contents 5

List of Figures 7

List of Tables 9

1 Introduction and Overview 11

1.1 Scope and Limitations of Work . . . 12

2 Overview of Algorithms 13

2.1 Strong coupling . . . 13

2.2 Weak coupling . . . 14

2.3 Intermediate Coupling . . . 15

2.4 Software Implementations . . . 19

3 Prototyping Results 21

3.1 Interfacial Coupling Prototype . . . 21

3.2 Shared-Domain Coupling Prototype . . . 24

3.3 Prototyping Conclusions . . . 29

4 Sierra Multi-Physics 31

4.1 Interfacial Coupling via Premcal, Coupled Premo-Calore . . . 31

4.2 Shared Domain Coupling via Aria . . . 43

4.3 Conclusions . . . 47

5 Analysis and Observations 51

5.1 Introduction and Setup of the Theory . . . 51

5.2 Analysis of Weak Coupling . . . 52

5.3 Conclusion . . . 54

6 Numerical Analysis of Coupling Using Matlab 55

6.1 Matrix Calculations Using MATLAB 55
6.2 Example: A Coupled Voltage, Displacement, and Temperature MEMS Problem . . . 56

7 Nonlinear Elimination for Circuit/Device Coupling 65

7.1 Theory 65
7.2 Results 67
7.3 Summary 70

8 Example of Coupled Heat Conduction and Enclosure Radiation 71

8.1 Overview 71
8.2 Problem Formulation 72
8.3 Discrete Approximation of the Coupled Problem 74
8.4 Coupling Strategies 75
8.5 Numerical Examples 77
8.6 Conclusions 87

9 Project Summary and Future Work 89

References 91

List of Figures

3.1 Schematic of the 1-dimensional interfacial coupling prototype problem. 21

3.2 Weak coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. 23

3.3 Weak coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. 24

3.4 JFNK coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. 25

3.5 Number of nonlinear iterations required to achieve $\|R\| < 10^{-8}$ at each time step. Loose coupling is the weak coupling of Section 2.2, and Newton coupling is the JFNK algorithm using a block-diagonal approximation to the Jacobian computed every nonlinear iteration as a preconditioning matrix as described in Section 2.3.2.1. 27

3.6 Representative nonlinear convergence behavior at a single time step. Newton results are the same as in figure 3.5. 28

4.1 Outer fluid and inner solid domains for external compressible flow over a thermally conducting circular cylinder. A two-element planar slab of thickness 1.0 is used. 32

4.2 Blunt-wedge geometry. 32

4.3 Temperature profiles for case with $T_{z=\pm 0.5} = 100.0$. 36

4.4 Temperature profiles for case with $T_{z=\pm 0.5} = 300.0$. 37

4.5 Temperature profiles for case with $T_{z=\pm 0.5} = 500.0$. 38

4.6 Comparison of convergence behavior for weak and JFNK coupling methods represented as coupled residual norm $||\mathbf{R}(\mathbf{W}, T)||$ vs number of nonlinear iterations. Note that the number of nonlinear iterations for weak coupling corresponds to those for the sequence of Premo problems. 40

4.7 Temperature profiles for Blunt-wedge problem. 41

4.8 Comparison of convergence behavior for weak and JFNK coupling methods represented as coupled residual norm $||\mathbf{R}(\mathbf{W}, T)||$ vs number of nonlinear iterations for the Blunt-wedge problem. Note that the number of nonlinear iterations for weak coupling corresponds to those for the sequence of Premo problems. 42

4.9 MEMS thermal actuator used as a switch. 43

4.10 Aria 2D MEMS actuator after voltage loading with $V = 1.934V$. Displacements have been amplified by a factor of 300 for aiding visualization. 45

4.11 Convergence behavior for pure Newton coupling in Aria 46

4.12 Non-zero structure of "Jacobi" (top) and "Broyden" (bottom) matrix approximations. 48

4.13 Convergence rate behavior for weak and JFNK coupling in Aria for $\Delta V = 1.7V$ 49

6.1 Aria Region and SiC Materials block 57

6.2 Example of a solution control and transfer block 58

6.3 Special Sierra region added to the calculation to calculate the global solution norm . . 59

6.4 Convergence rates of the Jacobi and Gauss Seidel methods compared with Matlab calculations 60

6.5 Example of Sierra's input blocks for implementing NOX's JFNK algorithm 61

7.1 Diode rectifier circuit test model. 67

7.2 Diode rectifier results: Plot of Voltage (V) vs. time (sec). 68

7.3 npn BJT circuit test model. 68

7.4 npn BJT: Plot of the electric potential. 69

7.5 npn BJT: Norm of the error in the residual as a function of outer iteration. 70

8.1 Temperature solution for 3D transient ER coupling problem 78

8.2 Spatial adaptivity failure from weak coupling in a 3D transient ER problem 80

8.3 Temperature solution for 3D stationary ER coupling problem 80

8.4 Comparison of error indicator versus CPU for stationary adaptivity 81

8.5 Temperature solution for ER verification problem 82

8.6 Temperature solution for adjoint ER problem 86

8.7 Adjoint temperature field. (left) fully coupled adjoint (right) weak coupled adjoint. . . 87

8.8 Temperature field. (left) fully coupled adjoint (right) weak coupled adjoint. 87

List of Tables

1.1 Terminology describing degree of coupling of physics and algorithms. 11

3.1 Convergence rates for weak coupling with $R = 0$ and JFNK coupling with $R = 0$ and $R = 5.67$. 23

3.2 Results for Newton-based coupling using Lagged Inexact and Broyden matrix approximations to $\mathbf{J}$. Simulations are for the coupled T, C and U prototype problem with each physics discretized using 1000, 1000 and 2000 1-dimensional finite elements, respectively. 50 times steps of size $\Delta t = 0.50$ were taken. 27

3.3 Results for JFNK coupling using the matrices of Table 3.2 as preconditioners. The same spatial and temporal discretizations hold as for Table 3.2. 28

3.4 Results comparing weak coupling to strong coupling demonstrating improvements of the latter on both robustness and efficiency. 29

4.1 Comparison of weak coupling and JFNK coupling for the circular cylinder geometry. 39

4.2 Number of nonlinear and total linear iterations required to converge the MEMS actuator problem at various voltage loadings up to the critical value of $\Delta V = 2.879$. . . . 46

6.1 Nonlinear Iteration Counts for JFNK coupling vs. strong coupling 63

7.1 Convergence performance as a function of residual tolerances. 69

8.1 Nonlinear solve times for 3D transient ER coupling problem 79

8.2 Nominal parameter values for verification problem. 83

8.3 Verification of weak coupling analytic rate estimate. 84

8.4 84

8.5 85

Chapter 1

Introduction and Overview

It is important to begin a study and discussion of coupling strategies by defining some terminology. Lack of precise definitions has often led to confusion in the literature and among working communities.

First, there should be a recognizable distinction between coupled physics and coupled codes. It is often the case that a single application code accounts for multiple interacting physics, e.g. an application code developed to solve reacting flow typically consists of solving for vector-valued velocities, a scalar pressure and various scalar species concentrations. Contrast this with the desire to augment the coupled physics within an application by coupling to another code responsible for the added physics, e.g. accounting for thermal effects in reacting flow by coupling a reacting flow code to a separate code that solves a heat equation. This distinction is important given the large disparity in algorithm development and understanding between the two. Much effort has gone into developing and tuning algorithms for multi-physics within a single application code. However, little attention to detail and implications has been given to multi-physics coupling across multiple application codes. One goal of the current work is to elucidate and address the issues of the latter while preserving the benefits of the former. for example, specialized solution techniques embodied within an application code should not be abandoned in order to "better" couple to another application code.

Given the recent and growing interest to enable multi-physics involving multiple application codes, we must next define terminology that allows one choice of coupling algorithm to be compared to another and possibly considered "better." We deem the ability to obtain a converged solution to be of greater importance than the rate of convergence. Simply stated, it is better to be able to obtain a solution than to not obtain one efficiently. We therefore define a coupling algorithm to be better than another primarily if it is more robust and secondarily if it is more efficient. As might be expected, algorithm robustness and efficiency is often related to the degree of mutual dependence of the interacting physics. For example, a multi-physics problem may exhibit one-way coupling in which one physics affects another but not vice-versa. Moreover, for mutually-dependent multi-physics the degree of physical coupling can vary and is often reflected by one or more physical parameters. We are naturally led then to consider and distinguish both the degree of coupling among the various physics as well as how well a coupling algorithm and associated implementation (i.e. software) account for this [(1)]. Taken together, these considerations lead to the terminology summarized in Table 1.1. Intuitively, a good balance between robustness and efficiency should result

Physics	Loose	Tight
Algorithms		
Weak		
Strong		

Table 1.1. Terminology describing degree of coupling of physics and algorithms.

along the diagonal of the table and be questionable off the diagonal. The goal of this work is to generally provide and determine appropriate algorithms for various multi-physics scenarios.

1.1 Scope and Limitations of Work

The scope of the work represented by this LDRD aims ultimately to impact the complete field of multi-physics coupling from algorithms to software implementation. We have found that attempts at impacting one area, algorithms for example, while excluding other areas masks relevant issues that must ultimately be addressed for a useful general coupling capability. So, Chapter 2 provides an overview of coupling algorithms ranging from weak algorithms based on fixed-point implementations to strong coupling embodied as a pure implementation of Newton's method. Here we lay out a general approach to intermediate coupling which seeks to leverage the simplicity of requirements for weak coupling to achieve the performance of strong coupling. Chapter 3 describes results of coupling algorithms applied to representative prototyping problems. Trade-offs among the choices for strength of coupling algorithm are elucidated. We also discuss our software implementation within the Trilinos solver library and NOX in particular that makes our coupling algorithms generally available. Extending our work to real application environments is the subject of chapters 4-8 where we show performance for real-world applications of importance to Sandia. Some analysis work is presented in Chapter 5. Finally, Chapter 9 describes ongoing and future projects incorporating the work of this LDRD to varying degrees.

Our work is limited in that it does not seek to address issues such as operator-splitting with regard to temporal accuracy and stability, for example. However, we do touch on practical issues related to these in chapter 7. In general for time-dependent problems, our work assumes an adequate time-discretization method and seeks to better solve the resulting coupled nonlinear system of equations. We acknowledge that there is an interplay between the mode of time stepping and the details of the solution of the coupled system at a given time step, but a rigorous study of such is beyond the scope of this work.

Chapter 2

Overview of Algorithms

This chapter provides an overview of coupling algorithms ranging from weak coupling based on fixed point iterations to strong coupling represented at the extreme by a pure Newton method. Most of the discussion will center around intermediate coupling strategies which seek to provide the benefits of strong coupling but with little to no requirements imposed on applications codes beyond providing weak coupling data. In all approaches, the object is to find a solution to the following nonlinear system of equations:

$$\mathbf{R}(\mathbf{x}^*) = \mathbf{0} \tag{2.1}$$

where $\mathbf{R}$ is called the residual and $\mathbf{x}^*$ is the solution to the nonlinear system when eq. (2.1) is satisfied. Within the context of coupled multi-physics involving multiple applications, eq. (2.1) can be partitioned to reflect contributions to the overall coupled problem arising from individual applications, e.g. for multi-physics arising from two coupled applications A and B:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix}, \tag{2.2}$$

where $\mathbf{x}_A$ are the unknowns for the A application and $\mathbf{x}_B$ are the unknowns for the B application. The partitioned residual equation corresponding to eq. (2.1) is:

$$\begin{bmatrix} \mathbf{R}_A(\mathbf{x}_A^*, \mathbf{x}_B^*) \\ \mathbf{R}_B(\mathbf{x}_A^*, \mathbf{x}_B^*) \end{bmatrix} = \mathbf{0} \tag{2.3}$$

2.1 Strong coupling

In this work, we consider the strongest form of coupling algorithm to be that of a pure Newton method. By pure, we mean a Newton method implemented without any approximations beyond representing numbers with finite machine precision. This excludes variants of Newton's method such as implementations employing an iterative linear solver, Jacobian matrices filled numerically using finite differences, secant updates to an approximate Jacobian matrix, etc. We consider all of these and others to fall within the realm of intermediate coupling strategies to be discussed in section 2.3.

Newton's method is derived by expanding eq. (2.1) about the solution to the coupled nonlinear problem,

$$\mathbf{R}(\mathbf{x}^*) = \mathbf{R}(\mathbf{x}) + \mathbf{J}(\mathbf{x})(\mathbf{x}^* - \mathbf{x}) + O\left(\|\mathbf{x}^* - \mathbf{x}\|^2\right) \tag{2.4}$$

where $\mathbf{J}$ is the Jacobian matrix that expresses the sensitivity of each equation to each unknown,

$$J_{ij} \equiv \frac{\partial R_i}{\partial x_j}. \tag{2.5}$$

Truncating second and higher order terms causes $\mathbf{x}^*$ to not represent the true solution in general. However, if we assume that it does, i.e. $\mathbf{R}(\mathbf{x}^*) = \mathbf{0}$, eq. (2.4) can be rearranged to give the classical form for computing solution updates using Newton's method,

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{R} \tag{2.6}$$

where both $\mathbf{J}$ and $\mathbf{R}$ are evaluated at the current nonlinear solution iterate $\mathbf{x}^k$. An iteration of Newton's method involves solving the linear problem of eq. (2.6) and using the result to update the nonlinear solution iterate,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k \tag{2.7}$$

where superscript k represents the iteration number.

Eq. (2.6) is equally valid for solving a nonlinear system arising from coupled physics applications as it is for standalone applications. It is instructive to write the equation in partitioned form corresponding to eq. (2.3) as follows

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{J}_{AB} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_A \\ \Delta\mathbf{x}_B \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_A \\ \mathbf{R}_B \end{bmatrix}. \tag{2.8}$$

Within the context of coupling multiple physics applications, a pure Newton algorithm suffers from one paramount issue. Namely, it requires all sensitivities of equations to variables to be correctly evaluated. With reference to eq. (2.8) application A must be able to supply correct sensitivities to application B variables and vice-versa. These are reflected in the off-diagonal blocks of the Jacobian matrix, i.e. $\mathbf{J}_{AB}$ and $\mathbf{J}_{BA}$. In production application environments, such a requirement is excessively stringent. For example, either or both applications A and B may be actively developed, proprietary, or even without available source code. Moreover, not only sensitivities of how one application uses another's variables must be determined but also sensitivities to how the variables are transferred between applications. Data transfers can be complicated and often comprise an active area of research requiring rapid performance assessments of candidate methods. These and other factors make a pure Newton algorithm prohibitive as a general coupling capability.

2.2 Weak coupling

From a software standpoint, the simplest means of coupling multiple application codes together to achieve a multi-physics capability is based on some form of fixed-point iteration also known as successive substitution. As opposed to a pure Newton algorithm, weak coupling represents the most accommodating and general approach. Its appeal is its simplicity and the minimal requirements placed upon application codes to participate. and may even be preferred over stronger coupling when the inter-dependence among participating physics is loose. The idea is represented algorithmically for coupling three physics applications in eqs. (2.9)-(2.15).

$$\text{Do while} \quad \|\mathbf{R}(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C)\| > \epsilon \quad :$$

$$\text{Solve for } A\text{:} \quad \mathbf{R}_A(\mathbf{x}_A, \tilde{\mathbf{x}}_\mathbf{B}, \tilde{\mathbf{x}}_\mathbf{C}) = \mathbf{0} \tag{2.9}$$

$$\Downarrow \quad \tilde{\mathbf{x}}_\mathbf{A} \leftarrow \mathbf{x}_A \tag{2.10}$$

$$\text{Solve for } B\text{:} \quad \mathbf{R}_B(\tilde{\mathbf{x}}_\mathbf{A}, \mathbf{x}_B, \tilde{\mathbf{x}}_\mathbf{C}) = \mathbf{0} \tag{2.11}$$

$$\Downarrow \quad \tilde{\mathbf{x}}_\mathbf{B} \leftarrow \mathbf{x}_B \tag{2.12}$$

$$\text{Solve for } C\text{:} \quad \mathbf{R}_C(\tilde{\mathbf{x}}_\mathbf{A}, \tilde{\mathbf{x}}_\mathbf{B}, \mathbf{x}_C) = \mathbf{0} \tag{2.13}$$

$$\Downarrow \quad \tilde{\mathbf{x}}_\mathbf{C} \leftarrow \mathbf{x}_C \tag{2.14}$$

$$\text{Evaluate :} \quad \|\mathbf{R}(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C)\| \tag{2.15}$$

In words, fixed-point iteration consists of solving each physics in a sequence for its own problem variables while holding all other physics applications' variables fixed. Updated values are transferred to the next physics application in the sequence which in turn computes a solution for its own variables and so on. After the sequence has been been traversed, an overall coupled residual is evaluated using the most current values for all physics application variables, and measures of convergence

are tested to determine whether or not to repeat the sequence. This last step typically involves a synchronization of all variables to all physics applications to ensure consistency of updated values. A typical convergence criterion would be reduction of the L_2-norm of the coupled problem residual below some prescribed tolerance, for example. Eqs.(2.10), (2.12) and (2.14) represent transfers of updated variables form a physics application to a form needed by the subsequent physics module. These steps can be as simple as nodal interpolations of variables from one mesh to another (with copying of values on conformal meshes being a special case) to complex mappings, not necessarily one-to-one. An example of the latter would be the transfer of post-processed stresses across non-conformal interfaces performed so as to conserve momentum.

More generally, the algorithm of eqs.(2.10)-(2.14) can be expressed as a nonlinear fixed-point vector mapping for the entire set of coupled problem unknowns as follows:

$$\mathbf{x}^{k+1} = \mathbf{F}\left(\mathbf{x}^{k}\right) \tag{2.16}$$

Fixed-point theory shows that iterates of eq.(2.16) convergence to a solution, $\mathbf{x}^*$ if

$$\left|\left| \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} \right|\right| < 1 \tag{2.17}$$

Moreover, the rate of convergence is linear with a constant rate determined by eq.(2.16), i.e.

$$\left|\left|\mathbf{x}^{k+1} - \mathbf{x}^*\right|\right| \leq \gamma \left|\left|\mathbf{x}^{k} - \mathbf{x}^*\right|\right| \tag{2.18}$$

with $\gamma \leq ||\partial \mathbf{F}/\partial \mathbf{x}|_{\mathbf{x}^*}||$. As will be seen in both our prototyping studies and our experiences with real-world applications, these theoretical predictions are observed and provide a foundation on which to base expectations regarding the existence and efficiency of weak-coupling convergence.

We want to pause here to make an important point regarding classification of the overall coupling algorithm compared to the algorithm(s) used by each physics application to update its own problem variables, which could be different for each physics application in general. It is entirely possible and even common to perform the updates represented by eqs.(2.9), (2.11) and (2.13) using a robust method, e.g. Newton's method. However, the overall coupling algorithm is classified as weak because each physics application updates its own variables independent of any concurrent changes to other application's variables. It is therefore based on a fixed-point approach.

Because of simplicity and straightforward implementation, weak coupling is often the first and sometimes only approach considered to enable multi-physics capability by coupling multiple physics applications. As will be shown later, fixed-point algorithms can exhibit strongly problem dependent convergence behavior that under-performs Newton algorithms in general. It would therefore be very desirable to develop coupling algorithms that preserve the simplicity of weak coupling with the performance of strong coupling. Such intermediate coupling algorithms are the focus of this work and are discussed in the following section.

2.3 Intermediate Coupling

This section presents several strategies to obtain performance approaching that of strong coupling with requirements approaching that of weak coupling. We do this by leveraging weak coupling functionality to effect approximations to Newton's method for the coupled problem. Our approach permits a spectrum of algorithms. The more functionality applications can provide, the fewer the approximations to a pure Newton algorithm.

2.3.1 Jacobian-Free Newton-Krylov (JFNK)

Perhaps the most important algorithm we consider for intermediate coupling is that of JFNK (2). Simply stated, it is a Newton method that employs a Krylov-based iterative linear solver, e.g. GMRES, conjugate gradient, etc. Its utility lies in recognizing that the Krylov linear solver only requires the action of the Jacobian on a vector. The Jacobian matrix does not have to be explicitly formed, and in this way JFNK bypasses the paramount obstacle associated with a pure Newton algorithm. It has other desirable properties as well such as significantly reduced memory requirements. JFNK is derived as a variation of Newton's method:

$$\mathbf{R}(\mathbf{x}+\epsilon\mathbf{p}) = \mathbf{R}(\mathbf{x}) + \mathbf{J}(\mathbf{x}+\epsilon\mathbf{p}-\mathbf{x}) + O\left(\|\epsilon\mathbf{p}\|^2\right). \tag{2.19}$$

Truncating second and higher order terms and rearranging gives

$$\mathbf{J}\mathbf{p} \approx \frac{\mathbf{R}(\mathbf{x}+\epsilon\mathbf{p}) - \mathbf{R}(\mathbf{x})}{\epsilon} \tag{2.20}$$

which provides the action of the Jacobian on a vector $\mathbf{p}$ accurate to first order. The parameter ϵ is a perturbation parameter, and eq. (2.20) can be viewed as an approximation to the directional derivative of the residual in direction $\mathbf{p}$. Eq. (2.20) is ideally suited to our goal of providing strong coupling convergence behavior with weak coupling functionality in that it captures sensitivities of all equations to all variables with simple residual fills. Moreover, it has the added benefit of reduced memory requirements and does not require determination of any sensitivities of equations to variables, even its own. Compared to a pure Newton method, JFNK is an approximate Newton method with approximations resulting from the inexact convergence of the iterative linear solves and, most importantly, from the approximation error associated with the action of the Jacobian. The latter source of approximation can represent the greatest disadvantage of this algorithm. The approximation error is directly related to the selection of a value for ϵ which should represent a meaningfully small perturbation for all variables of the coupled problem. This assumption breaks down when variables associated with different physics applications being coupled differ by orders of magnitude. The coupled problem is then said to be poorly scaled. Unless otherwise noted, values for ϵ used in this work were computed using a heuristic expression that has proved to work well in a variety of contexts,

$$\epsilon = \lambda * \left(\lambda + \frac{\|\mathbf{x}\|}{\|\mathbf{p}\|}\right) \tag{2.21}$$

with $\lambda = 10^{-6}$.

Another issue that is essential to the viability of JFNK is that of preconditioning. It is in this area that the matrix-free appeal of eq.(2.20) must yield to some degree to the construction and use of a preconditioning matrix and is the reason this algorithm is referred to as Jacobian-free instead of matrix-free. As with any iterative linear solver, adequate preconditioning is a necessity for obtaining converged linear solutions with reasonable effort. The preconditioned form of eq. (2.20) is

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{p} \approx \frac{\mathbf{R}(\mathbf{x}+\epsilon\mathbf{M}^{-1}\mathbf{p}) - \mathbf{R}(\mathbf{x})}{\epsilon} \tag{2.22}$$

where $\mathbf{M}$ is a preconditioning matrix which has been applied as a right-preconditioner as shown. In practice, right-preconditioning is applied in a two-step process that involves first solving for a preconditioned form of the vector:

$$\mathbf{M}\mathbf{y} = \mathbf{p} \tag{2.23}$$

and then using $\mathbf{y}$ in place of $\mathbf{p}$ in eq.(2.20). A good preconditioner should be relatively easy to evaluate and apply while retaining as much of the character of the Jacobian as possible. In addition to problem scaling, preconditioner effectiveness can also be highly problem-dependent. Of all issues affecting the viability and performance of JFNK, preconditioning is often most important and the most challenging. For this reason, much of this work has involved implementing various types of preconditioning for JFNK and assessing their relative merits in a variety of contexts.

2.3.2 Matrix Approximations

This section describes a variety of ways of constructing matrix approximations to the true Jacobian. These can be used as either an approximate Jacobian or as a preconditioning matrix $\mathbf{M}$ for JFNK as in eq. (2.22). As a point of reference, the partitioned Jacobian for the coupled problem in eqs. (2.9)-(2.15) is represented as

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{AA} & \mathbf{J}_{AB} & \mathbf{J}_{AC} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} & \mathbf{J}_{BC} \\ \mathbf{J}_{CA} & \mathbf{J}_{CB} & \mathbf{J}_{CC}. \end{bmatrix} \tag{2.24}$$

We next describe a variety of ways of forming approximations to this matrix.

2.3.2.1 Inexact Jacobian

For physics applications that employ a Newton method to update their own variables, a simple and straightforward construction of an approximate Jacobian consists of combining each application's Jacobian matrix into a composite block-diagonal approximation, e.g.

$$\tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_{AA} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{BB} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{CC}. \end{bmatrix} \tag{2.25}$$

If used in place of the true Jacobian, eq. (2.25) amounts to a weak coupling algorithm without transfers of updated variables to the next problem in the sequence, i.e. solve eqs. (2.9), (2.11) and (2.13) simultaneously without eqs. (2.10), (2.10) and (2.10). However, strong coupling behavior is obtained by using eq. (2.25) as a preconditioning matrix $\mathbf{M}$ within the JFNK algorithm of eq. (2.22). This is precisely our objective of achieving strong coupling performance by leveraging weak coupling functionality.

2.3.2.2 Finite Differencing

Another means of evaluating Jacobian entries is to replace the continuous derivatives of eq. (2.5) with discrete approximations as follows:

$$J_{ij} \approx \frac{R_i(\mathbf{x} + \delta \mathbf{e}_j) - R_i(\mathbf{x})}{\delta} \quad . \tag{2.26}$$

This approach has the advantage of selecting a meaningful perturbation size δ for each variable, i.e.

$$\delta = \alpha * |x_j| + \beta \tag{2.27}$$

where we have employed $\alpha = 10^{-4}$ and $\beta = 10^{-6}$. The disadvantage of this approach is the enormous cost. For a total problem size of N variables, forming a numerical approximation to the Jacobian using eq. (2.26) would require N^2 function evaluations, i.e. the entire residual vector $\mathbf{R}$ must be filled for each variable x_j. To mitigate this cost, variables can be grouped into colors such that each equation R_i depends on at most one variable of a particular color. This allows all variables of a given color to be perturbed simultaneously and reduces the cost for numerically approximating the Jacobian to cN with c being the number of colors. For applications that solve discretized differential equations over a mesh, the number of colors reflects the connectivity of the mesh and is typically

$c < 100$ independent of mesh refinement. For static meshes, e.g. no mesh refinement, the coloring algorithm is performed once on the structure or matrix graph of the Jacobian matrix and represents a one-time overhead which is amortized over the course of the simulation. Application of matrix coloring transforms eq. (2.26) into the following,

$$J_{ij(m)} \approx \frac{R_i(\mathbf{x} + \delta_m \cdot \mathbf{c}_m) - R_i(\mathbf{x})}{\delta_m \cdot \mathbf{e}_{j_m}} \quad , \tag{2.28}$$

where $\mathbf{c}_m$ is the mth coloring vector of length N with ones in positions associating a variable with this color, and subscript $j(m)$ reflects a mapping into the appropriate columns of the Jacobian matrix for this color. For each variable in the set for a given color, a perturbation is computed using eq. (2.27). In this work, we use a greedy coloring algorithm (3) implemented in the Trilinos nonlinear solver package NOX.

2.3.2.3 Secant Approximation

A broad class of inexpensive methods to update approximate Jacobian matrices falls under the category of secant or Broyden methods. Their use as approximate Jacobian matrices produces what is commonly called quasi-Newton methods. In short, an existing approximation for the Jacobian matrix is updated after each Newton iteration using a multi-dimensional secant evaluation. These are inexpensive, recursive, rank-one updates given by the following:

$$\tilde{\mathbf{J}}_{k+1} = \tilde{\mathbf{J}}_k + \frac{(\mathbf{y}_k - \tilde{\mathbf{J}}_k \mathbf{s}_k)\mathbf{s}_k^T}{\mathbf{s}^T\mathbf{s}} \tag{2.29}$$

where $\mathbf{y}_k \equiv \mathbf{R}_{k+1} - \mathbf{R}_k$ and $\mathbf{s}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k$. This is equivalent to a multi-dimensional secant approximation to the Jacobian which preserves the following properties:

$$\tilde{\mathbf{J}}_{k+1}\mathbf{s}_k = \tilde{\mathbf{J}}_k\mathbf{s}_k \tag{2.30}$$

$$\tilde{\mathbf{J}}_{k+1}\mathbf{q} = \tilde{\mathbf{J}}_k\mathbf{q}, \quad \forall \mathbf{q} : \mathbf{s}_k^T\mathbf{q} = 0 \tag{2.31}$$

As written, eq. (2.29) generates updated matrices that are dense in general. Within the context of massively parallel multi-physics simulations, dense matrices lead to unacceptable memory requirements and poor scalability. Because of this, we employ a modified form of eq. (2.29) that preserves the sparsity of the original Jacobian structure,

$$\tilde{\mathbf{J}}_{k+1} = \tilde{\mathbf{J}}_k + P_Z \left[\mathbf{D}^{-1} \left(\mathbf{y}_k - \tilde{\mathbf{J}}_k \mathbf{s}_k \right) \mathbf{s}_k^T \right] \tag{2.32}$$

where P_Z is a matrix projection operator that zeros elements of the update that lie outside the sparsity pattern of the original Jacobian matrix graph, and $\mathbf{D}$ is a diagonal matrix with entries $D_{ii} = \mathbf{s}_i^T\mathbf{s}_i$. This sparsity-preserving secant update produces approximate Jacobian matrices satisfying $\tilde{\mathbf{J}}_{k+1}\mathbf{s}_k = \mathbf{y}_k$, but the matrices no longer exhibit properties of eqs. (2.30) and (2.31).

2.3.2.4 Hybrid Approximations

Here we wish to simply point out that some or all of the preceding techniques can be employed together to produce a composite approximation to the Jacobian matrix which can then be used as either the coupled problem Jacobian or as a preconditioning matrix with JFNK. As an example, the blocks of the Jacobian in eq. (2.24) could be provided as follows: $\mathbf{J}_{AA}$ and $\mathbf{J}_{CC}$ taken as-is from each respective application, $\mathbf{J}_{BB}$ filled numerically using colored finite differences based on structural information provided from the B application, and the remaining blocks filled using secant updates to values in locations having nonzero sparsity in blocks $\mathbf{J}_{AB}$, $\mathbf{J}_{AC}$, $\mathbf{J}_{BA}$, $\mathbf{J}_{BC}$, $\mathbf{J}_{CA}$ and $\mathbf{J}_{CB}$. Moreover, some or all of these updates can be performed less frequently than every Newton iteration.

2.3.3 Nonlinear Elimination

All of the algorithms and variants described in Section 2.3.1 rely on one key piece of functionality, formation of the nonlinear residual for each application. It is not uncommon for even this seemingly simple requirement to be unavailable or overly burdensome. It is also possible that a certain application's solution technology embodies a significant investment in time and resources that would be lost if forced into the context of JFNK. Together these and other considerations lead to the need for an intermediate coupling approach that is as general as possible in that an application is not required to fill a residual, and any specialized solution capabilities are preserved. Coupling algorithms based on what we call nonlinear elimination do exactly this.

Nonlinear elimination involves simply eliminating one or more application's variables from the coupled problem at the appropriate time. Within the context of JFNK, each residual evaluation would involve the elimination of any application variables targeted using nonlinear elimination. Choices on which application(s) to treat in this manner are often problem-dependent. To see our use of this mode of coupling, see Sections 7.1 and 8.4.2

2.4 Software Implementations

We have built our multi-physics coupling technology as extensions to existing algorithm capability originally developed for solving nonlinear problems in single physics applications. As such, the JFNK algorithm of Section 2.3.1 and the matrix approximations of Sections 2.3.2.2 and 2.3.2.3 were either present or added to the Trilinos nonlinear solver package NOX. We then added another layer of abstraction via object-oriented extensions to handle the composition and decomposition of vector and matrix objects utilized in the coupling algorithms. The extensions allow a monolithic view of the coupled problem which delegates callback support to underlying individual NOX solvers attached to each physics application being coupled. This allows construction of the complete coupled problem residual, for example, from pieces obtained from callbacks to each physics application followed by an assembly step. Moreover, any additional functionality an application can supply can be readily used to enrich the coupling algorithms. For example, if an application can fill its own Jacobian, these values can be used in a hybrid matrix approximation as described in Section 2.3.2.4. If only the structure of an application's Jacobian is known, e.g. locations of non-zero entries, this information can be used to numerically approximate its block matrix contribution using either colored finite differences or secant updates of Sections 2.3.2.2 or 2.3.2.3, respectively. At present, use of NOX represents a minimum requirement for an application to participate in a coupled multi-physics simulation employing the intermediate coupling strategies described previously. We anticipate working to relax this requirement as our coupling work becomes integrated into subsequent efforts.

Chapter 3

Prototyping Results

This chapter applies the coupling algorithms from the previous chapter to two canonical coupling prototypes. We consider the canonical cases of coupling across an interface and coupling among physics applications within a shared physical domain. Using these idealizations we assess performance of coupling algorithms in the absence of complications arising from real applications.

3.1 Interfacial Coupling Prototype

We begin with what is perhaps the simplest coupling scenario involving two 1-dimensional problems coupled at a single node. This problem allows us to identify the limitations inherent in weak coupling approaches based on fixed-point analysis first presented in the literature (4). We then demonstrate how our Intermediate coupling algorithms of Section 2.3 can use the same functionality employed in weak coupling to effect dramatically more robust and efficient convergence.

3.1.1 Problem Description

The physical problem we consider was first presented in the literature (4) and consists of conjugate heat transfer between two 1-dimensional domains shown in figure 3.1. Each physical domain involves

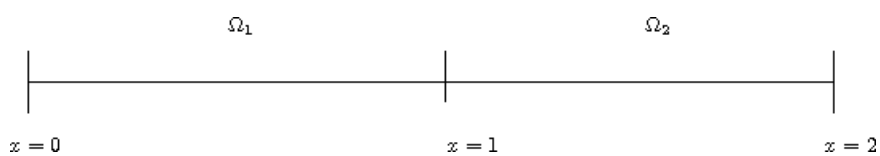


Figure 3.1. Schematic of the 1-dimensional interfacial coupling prototype problem representing conjugate heat transfer.

conductive heat transfer governed by the equations,

$$\frac{d^2T}{dx^2} - c\frac{dT}{dx} = 0, \forall x \in \Omega_1 \tag{3.1}$$

$$\kappa\frac{d^2T}{dx^2} = 0, \forall x \in \Omega_2 \tag{3.2}$$

with c a constant, and κ the ratio of thermal conductivity in Ω_2 to that in Ω_1. The temperature at the far ends of each domain are specified via Dirichlet boundary conditions, i.e.

$$T|_{x=0} = T_0 \tag{3.3}$$
$$T|_{x=2} = T_2 \tag{3.4}$$

The problem is well-posed after specifying boundary conditions at $x = 1$ consistent with conjugate heat transfer. Normally, these take the form of enforcing continuity of temperature, $T|_{x=1^-} = T|_{x=1^+}$, and of heat flux, $q|_{x=1^-} = q|_{x=1^+}$, across the interface at $x = 1$. However, we enforce these conditions in the same manner as (4) in which the two conditions are weighted by an arbitrary fraction and combined. This produces two Robin-type boundary conditions at the shared interface,

$$\alpha \left(T|_{x=1^-} - T|_{x=1^+}\right) + (1-\alpha)\left(q|_{x=1^-} - q|_{x=1^+}\right) = 0 \tag{3.5}$$

$$\beta \left(T|_{x=1^-} - T|_{x=1^+}\right) + (1-\beta)\left(q|_{x=1^-} - q|_{x=1^+}\right) = 0 \tag{3.6}$$

where $\alpha \in [0,1]$, $\beta \in [0,1]$ and $\alpha \neq \beta$ for well-posedness. The fractions α and β represent two adjustable parameters that control how information is communicated across the shared interface.

In the domain Ω_1, heat flux at the interface is governed by Fourier's law of heat conduction, and in Ω_2 it is considered to involve conduction as well as radiation transport. Together, these are expressed as,

$$q|_{x=1^-} = -\left.\frac{dT}{dx}\right|_{x=1^-} \tag{3.7}$$
$$q|_{x=1^+} = -\left.\frac{dT}{dx}\right|_{x=1^+} + R\left(T|_{x=1^+}^4 - T_2^4\right) \tag{3.8}$$

where R is a constant that reflects the amount of heat transfer due to radiation and represents the only source of nonlinearity in the coupled problem. When $R = 0$ each problem in domains Ω_1 and Ω_2 as well as the coupled problem is linear.

The 1-dimensional conjugate heat transfer prototype problem is perhaps the simplest canonical coupling problem one could construct. As formulated, there are three adjustable parameters: α, β and R, with R determining the nonlinearity of the coupled problem.

3.1.2 Results

3.1.2.1 Weak Coupling

We applied the weak coupling algorithm of Section 2.2 to the interfacial, conjugate heat transfer problem just described and obtained results consistent with the literature and reproduced in Figure 3.2. By reproducing published results, we were able to verify our weak coupling implementation and to establish a context in which to compare the merits of our intermediate coupling algorithms. The results of figure 3.2 are revealing in that they show that the existence of weak coupling convergence can be very problem-dependent. The values of α (x-axis) and β (y-axis) for which weak coupling converges when the problem is linear, i.e. $R = 0$, is almost completely inverted when the problem becomes markedly nonlinear, i.e. $R = 5.67$. In addition to the existence of convergence, the rate of convergence for weak coupling is also very problem-dependent. Figure 3.3 shows the rate of convergence as the norm of the coupled problem residual for each weak coupling algorithm iteration for the case $\alpha = 0.5$, $R = 0.0$ and three values of β approaching the critical value of $\beta = \alpha = 0.5$ at which

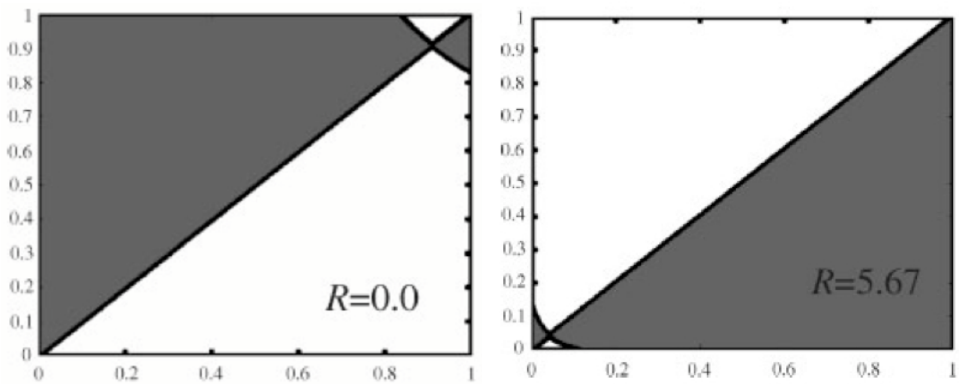


Figure 3.2. Weak coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. The x-axis and y-axis correspond to values for α and β, respectively. Shaded regions indicate regions of non-convergence. The R-values are used in eq.(3.8) and reflect the amount of nonlinearity in the prototype problem.

weak coupling fails to converge. Similar dependence of convergence rate on problem parameters is exhibited for the nonlinear problems, i.e. $R > 0$. To be consistent with the literature, we performed weak coupling for the coupled problem by employing Newton's method for each problem in Ω_1 and Ω_2. This is common in real-world application environments that employ fixed-point based weak coupling for multi-physics simulations.

3.1.2.2 JFNK Coupling

By performing weak coupling with each problem in Ω_1 and Ω_2 solved using Newton's method, we were able to construct an approximate preconditioning matrix for use with JFNK as described in Section 2.3.2.1. The existence of convergence and rate of convergence were dramatically improved over weak coupling using essentially no additional application information, e.g. without a full Jacobian matrix. A plot corresponding to Figure 3.2 is shown in Figure 3.4 for JFNK. Here, all R-values are represented, and convergence is achieved for all values of α and β except the degenerate case $\beta = \alpha$ for which no solution exists.

In addition to dramatically improved convergence, the rate of convergence is similarly improved using JFNK. Table 3.1 shows the number of iterations required to converge the coupled problem for weak coupling with $R = 0$ as well as for JFNK coupling with $R = 0$ and $R = 5.67$. When $R = 0$,

		Coupling Iterations		
α	β	Weak, $R = 0$	JFNK, $R = 0$	JFNK, $R = 5.67$
0.5	0.40	33	1	3
0.5	0.45	60	1	3
0.5	0.49	253	1	3
0.5	0.60	Fail	1	3

Table 3.1. Convergence rates for weak coupling with $R = 0$ and JFNK coupling with $R = 0$ and $R = 5.67$.

both problems in Ω_1 and Ω_2 as well as the coupled problem are all linear. JFNK coupling recaptures

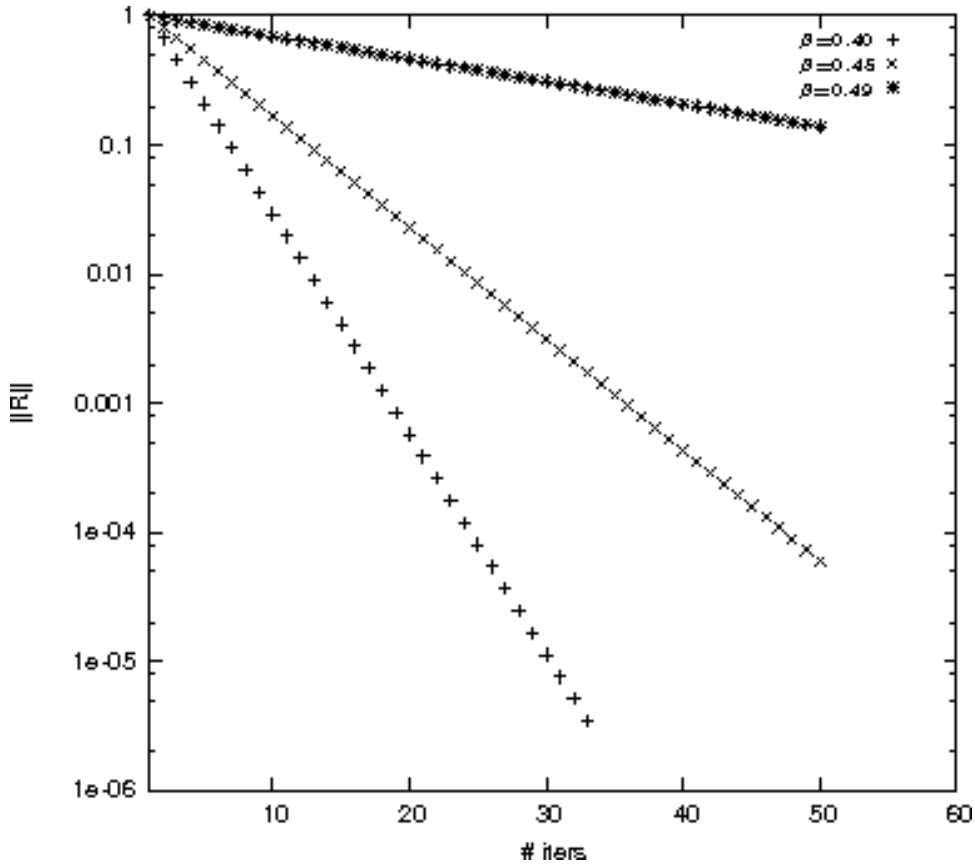


Figure 3.3. Weak coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. The x-axis and y-axis correspond to values for α and β, respectively. Shaded regions indicate regions of non-convergence. The R-values are used in eq.(3.8) and reflect the amount of nonlinearity in the prototype problem.

the linearity of the coupled problem as indicated by requiring only one nonlinear iteration. This is consistent with the performance of a Newton-based method applied to linear problems. Moreover, JFNK applied to the nonlinear coupled problem with $R = 5.67$ shows very good convergence rates consistent with quadratic convergence characteristic of Newton-based methods.

3.2 Shared-Domain Coupling Prototype

We next consider a canonical coupling scenario involving coupled physics within the same physical space. This coupling prototype consists of three physics representing temperature, species and one-dimensional compressible flow. Each is time-dependent. The temperature and species physics contain a diffusion term and are mutually coupled through a nonlinear source term present in both. One-dimensional compressible flow is represented by a Burgers equation having a temperature dependent viscosity. In addition to coupling within the same physical space, this problem also coupled partial differential equations (PDEs) of differing characteristic type, i.e. temperature and species being two parabolic PDEs and Burgers being a hyperbolic PDE. Temperature, species and

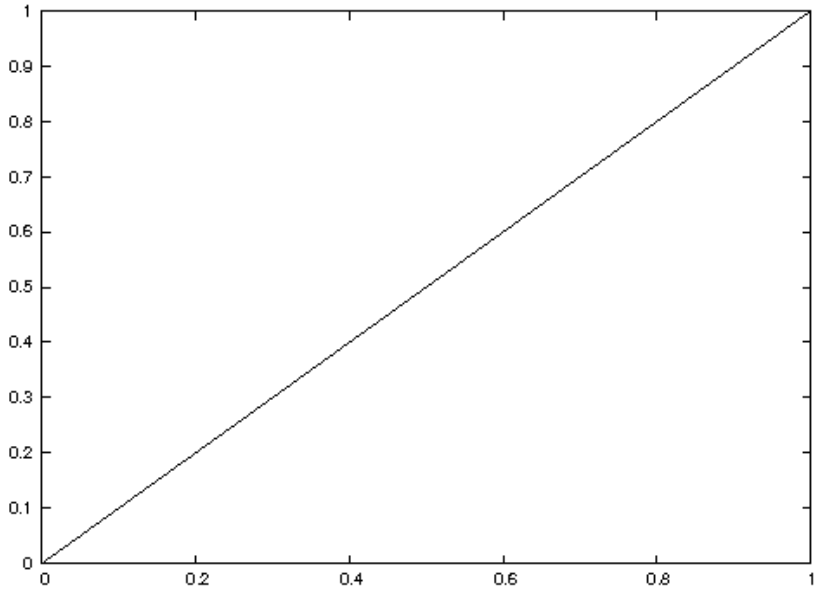


Figure 3.4. JFNK coupling convergence behavior for 1-dimensional conjugate heat transfer prototype. Axes are the same as in Figure 3.2. The line $\beta = \alpha$ is a degenerate case for which no solution exists. All R-values are represented.

compressible flow are represented by the following governing equations, respectively:

$$\frac{\partial T}{\partial t} = D_1 \frac{\partial^2 T}{\partial x^2} + \alpha + T^2 C - (1+\beta)T \tag{3.9}$$

$$\frac{\partial C}{\partial t} = D_2 \frac{\partial^2 C}{\partial x^2} - T^2 C + \beta T \tag{3.10}$$

$$\frac{\partial U}{\partial t} = \mu \frac{\partial^2 U}{\partial x^2} - \frac{\partial}{\partial x}\left(\frac{U^2}{2}\right) \tag{3.11}$$

where for this problem, α and β are constants and viscosity, μ represents a temperature-dependent viscosity expressed as, $\mu = T^{1.5}$. Dependent variables, T, C and U are all dimensionless. Equations (3.9) and (3.10) together represent a well-studied coupled system known as a Brusselator problem (5). Again, the Burgers equation of eq.(3.11) is added to the Brusselator system to provide a richer prototype module for the coupled system in that it introduces some degree of hyperbolicity to the system. We wish to test for limitations to our algorithms for coupling scenarios involving mixed PDE types before launching into real-world applications environments. Boundary conditions for each PDE are specified respectively as

$$T(t,0) = T(t,1) = \alpha = 0.6 \tag{3.12}$$

$$C(t,0) = C(t,1) = \beta/\alpha = 2.0/0.6 \tag{3.13}$$

$$U(t,0) = 1.0$$

$$U(t,1) = -1.0 \tag{3.14}$$

Equations (3.9)-(3.11) are discretized spatially using a Galerkin finite-element approach with linear

basis functions. This method is known to breakdown in the presence of solution discontinuities such as resolving a shock in compressible flow. The Burgers equation of eq.(3.11) with boundary conditions of eq.(3.14) can admit a shock depending on the value of viscosity μ. The range of temperature-dependent viscosity values employed in this work avoid a shock but do lead to large velocity gradients.

Temporal discretization is provided using the unconditionally stable backward Euler method. Time steps of $\Delta t \leq 1.0$ are used and provide adequate accuracy for the purposes of our study.

3.2.1 Results

This section present results for the nonlinearly coupled temperature, species and flow prototype problem obtained using the weak and strong coupling algorithms described in Chapter 2 and their implementation in the NOX nonlinear solver library in Trilinos.

For both weak and JFNK coupling algorithms, linear solves were performed using GMRES with incomplete LU factorization with no fill, i.e. ILU0, with a Krylov subspace of dimension 200, no restarts, and convergence defined as a reduction of the linear residual norm by eight orders of magnitude. Convergence for all nonlinear solves was defined to be $\|\mathbf{R}\| < 10^{-8}$.

3.2.1.1 Weak Coupling Results

Here, we establish a weak coupling baseline against which to compare performance of our Newton-based coupling strategies. The time-dependent problem of section 3.2 was run for 50 time steps of size $\Delta t = 0.50$ corresponding to about two periods of the periodic system. Each time step involved solution of the coupled nonlinear system. The weak coupling approach of section 2.2 required a total of 533 fixed-point iterations and took 111 seconds to complete. Figures 3.5 and 3.6 show convergence behavior for the weak coupling approach along with results for JFNK coupling using only problem information employed in weak coupling.

3.2.1.2 Results for Lagged Inexact & Secant Approximations to J

This section provides performance comparisons for approximating the Jacobian matrix $\mathbf{J}$ using the methods of Section 2.3.2 in addition to the extreme approximation of simply using the identity matrix $\mathbf{I}$ to approximate $\mathbf{J}$. Comparisons are also made for attempts at improving algorithm efficiency by updating the approximation matrices less frequently than each nonlinear iteration. The frequency of updating the approximation matrices are denoted in the table as follows: t_0, occurring only once for the entire simulation using the initial guess for the solution at the initial time step; Δt, occurring at the beginning of each time step using the initial guess for the solution and used for all nonlinear iterations during this time step, and 1, occurring for each nonlinear iteration as is traditionally done. The results of Table 3.2 show that a minimal amount of representation of the Jacobian is required for convergence of the coupled nonlinear system. Using only the identity matrix $\mathbf{I}$ in place of the Jacobian amounts to a plain Picard method and is clearly inadequate to converge the problem. Filling the diagonal blocks once with $\tilde{\mathbf{J}}$ evaluated at $t = 0$ and then updating the off-diagonal blocks using the modified Broyden approach also proves inadequate. Recomputing the matrices more frequently leads to convergence of all Jacobian approximations with performance of the variations differing by little more than a factor of 3.

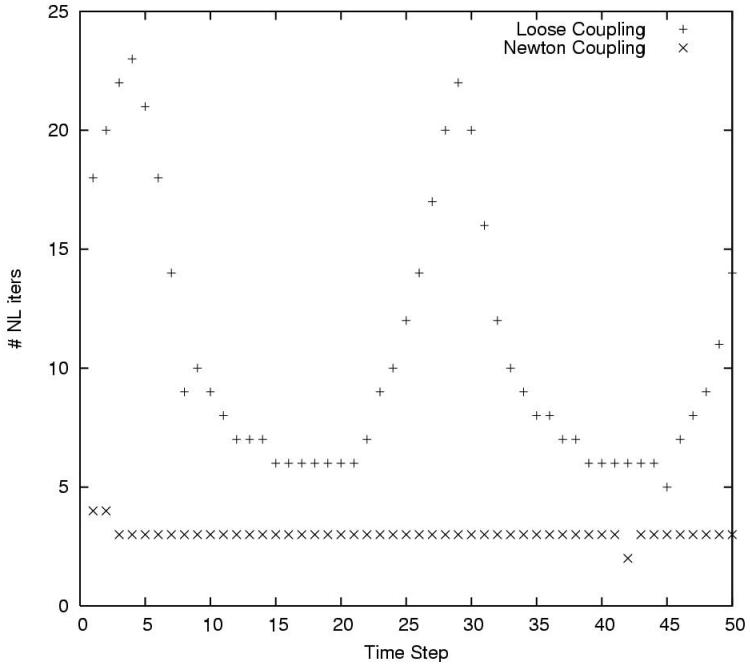


Figure 3.5. Number of nonlinear iterations required to achieve $\|R\| < 10^{-8}$ at each time step. Loose coupling is the weak coupling of Section 2.2, and Newton coupling is the JFNK algorithm using a block-diagonal approximation to the Jacobian computed every nonlinear iteration as a preconditioning matrix as described in Section 2.3.2.1.

Recompute Frequency	Jacobian	NL iters	LS iters	Time (s)
t_0	$\mathbf{I}$	Fail	Fail	Fail
t_0	$diag(\mathbf{I})$	Fail	Fail	Fail
t_0	$\tilde{\mathbf{J}}$	1348	1348	77.4
t_0	$\tilde{\mathbf{J}}_B$	Fail	Fail	Fail
t_0	$\mathbf{J}$	1067	6238	103
Δt	$\tilde{\mathbf{J}}$	783	783	48.2
Δt	$\tilde{\mathbf{J}}_B$	455	3228	50.2
Δt	$\mathbf{J}$	190	1501	29.4
1	$\tilde{\mathbf{J}}$	777	777	91.6
1	$\tilde{\mathbf{J}}_B$	446	3174	75.3
1	$\mathbf{J}$	151	1253	44.0

Table 3.2. Results for Newton-based coupling using Lagged Inexact and Broyden matrix approximations to $\mathbf{J}$. Simulations are for the coupled T, C and U prototype problem with each physics discretized using 1000, 1000 and 2000 1-dimensional finite elements, respectively. 50 times steps of size $\Delta t = 0.50$ were taken.

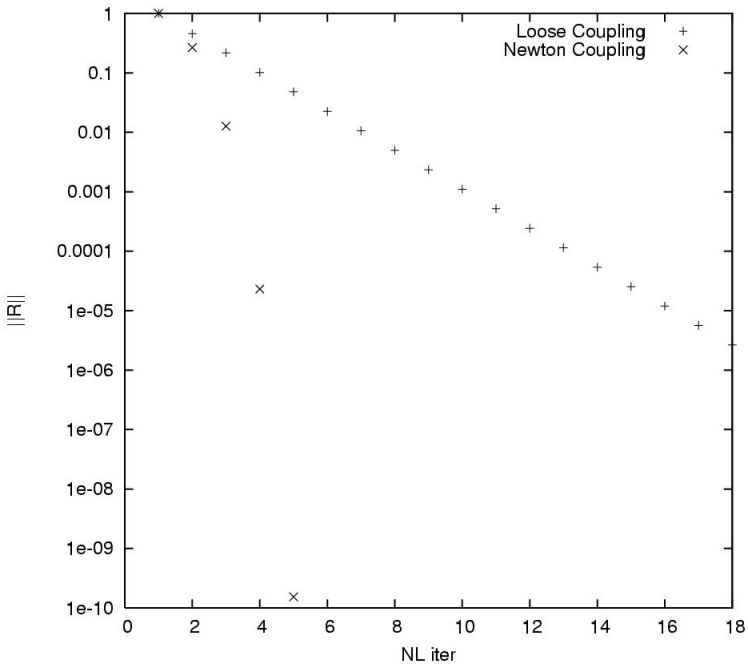


Figure 3.6. Representative nonlinear convergence behavior at a single time step. Newton results are the same as in figure 3.5.

3.2.1.3 JFNK Results

We now use JFNK with each of the matrix approximations of the preceding section now used as preconditioning matrices. Results are shown in Table 3.3. Here, all variants converge, even the effectively unpreconditioned case using $\mathbf{I}$. Whereas JFNK has recovered algorithm robustness, the choice of constructing a preconditioning matrix is now seen to be crucial to algorithm efficiency. We anticipate this comparison of results for a representative multi-physics prototype to provide insight and guidance for tuning analogous real-world coupling scenarios for greater efficiency.

Recompute Frequency	Preconditioner	NL iters	LS iters	Time (s)
t_0	$\mathbf{I}$	499	99,800	2420
t_0	$\tilde{\mathbf{J}}$	151	2578	61.4
t_0	$\tilde{\mathbf{J}}_B$	151	2582	71.0
t_0	$\mathbf{J}$	151	2666	66.5
Δt	$\tilde{\mathbf{J}}$	151	1311	43.0
Δt	$\tilde{\mathbf{J}}_B$	151	1390	52.8
Δt	$\mathbf{J}$	151	1371	54.3
1	$\tilde{\mathbf{J}}$	151	1206	46.6
1	$\tilde{\mathbf{J}}_B$	151	1316	55.9
1	$\mathbf{J}$	151	1256	70.3

Table 3.3. Results for JFNK coupling using the matrices of Table 3.2 as preconditioners. The same spatial and temporal discretizations hold as for Table 3.2.

3.2.1.4 Coupling Comparison

As a means of comparing weak coupling with a Newton-based coupling using the same problem information, figures 3.5 and 3.6 show convergence behavior of both approaches. The Newton approach uses the Jacobian-free Newton-Krylov operator along with preconditioner $\tilde{\mathbf{J}}$ updated every nonlinear iteration. Thus, the same Jacobian blocks used in weak coupling are now used to construct a preconditioner for the strong coupling. Strong coupling shows improvement in efficiency by more than halving the total time to solution. Robustness is seen in figure 3.5 which shows a nearly constant number of nonlinear iterations for the Newton-based approach compared to weak coupling which shows greater sensitivity to the point along the solution path. Improvements in both robustness and efficiency are further borne out by the results of table 3.4. Increasing the coupled problem size by a factor of 10 leads to a roughly proportional increase in the total time for both weak and strong coupling approaches. Doubling the time step from $\Delta t = 0.5$ to $\Delta t = 1.0$ causes the weak coupling approach to fail, whereas the strong Newton-based approach converges to a solution without incurring any additional cost over that of the easier time step size.

Method	Mesh (TxCxU)	Δt	Time, s
Weak	1000x1000x2000	0.5	111
JFNK	1000x1000x2000	0.5	46.6
Weak	10000x10000x20000	0.5	911
JFNK	10000x10000x20000	0.5	482
Weak	1000x1000x2000	1.0	Fail
JFNK	1000x1000x2000	1.0	46.0

Table 3.4. Results comparing weak coupling to strong coupling demonstrating improvements of the latter on both robustness and efficiency.

3.3 Prototyping Conclusions

The preceding results show that use of the same problem information used to perform straightforward successive substitution (weak coupling) can be better used to effect strong Newton-based coupling. Several variants of Newton-based coupling are enabled by object-oriented extensions to an existing nonlinear solver library, making these strong coupling strategies generally available to any set of codes or any physics simulation framework employing the library. The Newton variants allow for performance tuning which the preliminary results presented here suggest can be problem dependent.

Chapter 4

Sierra Multi-Physics

This chapter extends the prototyping studies to multi-physics scenarios involving production applications in the Sierra framework. Similar in presentation to the prototyping results, we begin with a scenario involving interfacial coupling. In particular, we worked to enable coupled conjugate heat transfer involving a compressible fluid modeled using Premo coupled to a thermally conducting body modeled using Calore. We then investigated coupled multi-physics within the same physical domain via multiple instances of Aria. In addition to assessing the performance of our coupling algorithms relative to Sierra's default mode of weak coupling, this part of our work allowed us to address software challenges associated with deploying our coupling algorithms in a production applications environment.

Briefly, the Sierra applications framework is a common applications development environment developed at Sandia to leverage basic services such as I/O, message-passing, element libraries, transfers of mesh data, etc. that are common among physics applications. Application codes brought into this environment can then send and receive data from other applications, thereby enabling multi-physics simulations without requiring significant re-writing of code.

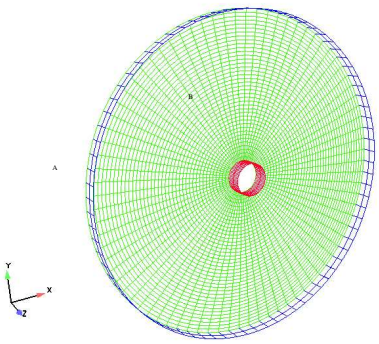
4.1 Interfacial Coupling via Premcal, Coupled Premo-Calore

Our first coupling study in Sierra required that we identify a relevant and realizable multi-physics scenario. We decided to couple the thermal code Calore to the compressible flow code Premo to enable conjugate heat transfer across a shared interface analogous to the prototype study of Section 3.1. Our first task, which turned out to be quite involved, was to create a new Sierra application, Premcal, which supported both the default weak coupling as well as our new Newton-based coupling algorithms.

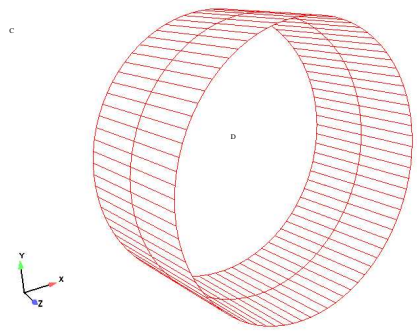
4.1.1 Problem Description

In this section, we describe the physical problem we targeted with Premcal. As shown in figures 4.1 and 4.2 we consider external compressible flow over a solid body whose temperature is governed by linear heat conduction. For both geometries, fluid flow external to the solid body is modeled as viscous and compressible via the Navier-Stokes equations. These can be expressed in the following form:

$$\int_{\mathcal{V}} \frac{\partial \mathbf{W}}{\partial t}\, d\mathcal{V} + \oint_{\Gamma} \left(\mathbf{F}^c + \mathbf{F}^v\right) \cdot \mathbf{n}\, d\Gamma = 0 \tag{4.1}$$

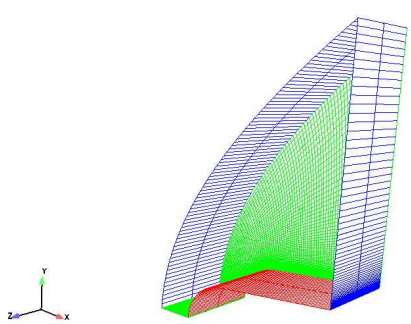


(a) Fluid domain and mesh having outer radius of 10.0.

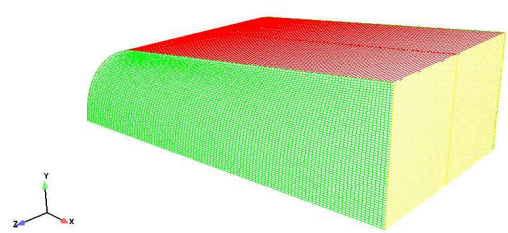


(b) Solid domain surface mesh having radial dimension of 1.0.

Figure 4.1. Outer fluid and inner solid domains for external compressible flow over a thermally conducting circular cylinder. A two-element planar slab of thickness 1.0 is used.



(a) Blunt-wedge fluid geometry and mesh configured to capture a shock at $Ma = 3.0$.



(b) Blunt-wedge conducting solid geometry.

Figure 4.2. Blunt-wedge geometry.

where $\mathbf{W} = \{\rho,\, \rho u,\, \rho v,\, \rho w,\, \rho E\}^T$ is the solution vector of conserved variables and

$$\mathbf{F}^c \cdot \mathbf{n} = \left\{ \begin{array}{c} (\rho u_i) n_i \\ (\rho u_1 u_i + p\delta_{i1}) n_i \\ (\rho u_2 u_i + p\delta_{i2}) n_i \\ (\rho u_3 u_i + p\delta_{i3}) n_i \\ ((\rho E + p)\, u_i) n_i \end{array} \right\}$$

is the inviscid flux vector, and

$$\mathbf{F}^v \cdot \mathbf{n} = \left\{ \begin{array}{c} 0 \\ \tau_{i1} n_i \\ \tau_{i2} n_i \\ \tau_{i3} n_i \\ (-q_i + u_j \tau_{ij}) n_i \end{array} \right\}$$

is the viscous flux vector. In the above, ρ is the density; $u_i = \mathbf{u}$ with $u = u_1$, $v = u_2$, and $w = u_3$ are the Cartesian velocity components; p is the pressure; E is the total energy defined by $E = e + \mathbf{u} \cdot \mathbf{u}/2$; e is the specific internal energy defined by $e = C_v T$; C_v is the specific heat at constant volume; T is the temperature defined for an ideal gas by $p = \rho R T$; R is the gas constant, $n_i = \mathbf{n}$ is the surface normal vector; and δ_{ij} is the Kronecker delta. Corresponding to the conservative state vector $\mathbf{W}$ we define a primitive state vector $\mathbf{U} = \{\rho, u, v, w, p\}^T$.

The viscous stress tensor is given by,

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \tag{4.2}$$

where μ is the molecular viscosity coefficient. Heat fluxes $\mathbf{q}$ are assumed to obey Fourier's Law,

$$\mathbf{q} = -\kappa \nabla T \ . \tag{4.3}$$

and a constant Prandtl number, Pr is used.

The solid body is shown as the red surfaces in figures 4.1 and 4.2 and is treated as impermeable to fluid flow and having no-slip. The former condition implies

$$\mathbf{F}^b = \mathbf{F} \cdot \mathbf{n}^b = \left\{ \begin{array}{c} 0 \\ p\mathbf{n}^b \\ 0 \\ 0 \end{array} \right\} \ ,$$

and the latter condition implies $\mathbf{u} = 0$. Superscript b denotes enforcement at the fluid-body interface. Far-field boundary conditions are depicted as blue surfaces in the figures and are specified via an assumed infinity profile, $\mathbf{u} = \mathbf{U}_\infty$. The green surfaces correspond to symmetry planes involving no flow across and no traction along them.

Heat transfer within the solid body is modeled as a balance of thermal transport governed by the following equation,

$$\rho C \frac{\partial T}{\partial t} - \nabla \cdot (\mathbf{K} \nabla T) - \dot{q} = 0 \tag{4.4}$$

where T is the unknown temperature field within the body, $\mathbf{K}$ is a thermal conductivity matrix, and $\dot{q}$ is a volumetric source term. For the present work, $\mathbf{K}$ is treated as a scalar constant, e.g. $\mathbf{K} = \kappa$, and the source term is assumed to be zero. We also limit this work to steady-state conditions. Together, these assumptions reduce eq.(4.4) to the Laplace equation governing linear heat conduction within the solid body.

Thermal boundary conditions at the fluid-body interface (red surfaces) represent the coupling between the fluid and solid domains. Conjugate heat transfer is modeled by requiring temperature and heat flux normal to the interface to be continuous across the interface:

$$T_f^b = T_s^b \tag{4.5}$$

$$\mathbf{n} \cdot \mathbf{q}_f^b = \mathbf{n} \cdot \mathbf{q}_s^b \ , \tag{4.6}$$

where subscript f denotes the fluid phase, and subscript s denotes the solid body. Equation (4.5) represents a Dirichlet boundary condition, while eq.(4.6) is of Neumann type.

4.1.2 Summary of Applications Codes

This section briefly describes the two codes Premo and Calore which solve the equations governing compressible viscous fluid flow and heat transfer, respectively.

Premo is based on a node-centered, edge-based, finite-volume algorithm similar to the formulation in Haselbacher and Blazek (6) and Luo *et al.* (7). More detail can be found in ref (8, 9). Calore is a computational heat transfer application code based on galerkin finite elements that solves a wide variety of both linear and nonlinear thermal physics models.

Premcal embodies the coupling of the two codes Premo and Calore via enforcement of the boundary conditions of eqs.(4.5) and (4.6). The design of Premo and Calore is such that we were required to enforce eq.(4.5) within Premo as a Dirichlet condition with values transferred from Calore. Conversely, eq.(4.6) is enforced by first having Premo compute a heat flux normal to the solid body at each surface node on the fluid/solid interface, then compute a corresponding set of convective heat transfer coefficients, and finally transfer these coefficients to Calore for use in its surface integration along the interface within the context of enforcing Neumann-type boundary conditions using Galerkin finite elements. Hence, continuity of heat flux across the interface is enforced via the expression:

$$-\kappa \mathbf{n} \cdot \nabla T = h\left(T_s^b - T_{ref}\right) \tag{4.7}$$

where Premo computes the left expression, and then determines values for h using an arbitrary but realistic value for T_{ref}.

4.1.3 Coupling Considerations

Premcal seeks to find a solution to the residual equation,

$$\mathbf{R} = \mathbf{R}(\mathbf{W}, T) = (\mathbf{R_W}, \mathbf{R}_T) = \mathbf{0} \tag{4.8}$$

with $\mathbf{R_W}$ defined by eq.(4.1) and $\mathbf{R}_T$ by eq.(4.4). The values $\mathbf{W}$ represent the vector degrees of freedom for Premo (5 per node in the discretized fluid domain), and T is the dependent scalar field within the solid body and represent the unknowns for Calore.

We applied both weak coupling and Newton-based coupling algorithms to effect solutions to eq.(4.8). For weak coupling, we had two choices for initiating the algorithm that differ by which application to solve first. This signifies yet another problem-dependent sensitivity of weak coupling performance as will be seen in the results section. By way of example, weak coupling initiated by solving flow

(Premo) first, leads to what we denote as W-T iterations corresponding to eq.(2.16):

$$
\begin{aligned}
&\text{Do while} \quad \|\mathbf{R}(\mathbf{W},T)\| > \epsilon \quad : \\
&\qquad \text{Solve Premo for } \mathbf{W}\text{:} \quad \mathbf{R}_{\mathbf{W}}(\mathbf{W},\tilde{T}) &&= \mathbf{0} && (4.9)\\
&\qquad\qquad\qquad\qquad\qquad\qquad \Downarrow && \tilde{\mathbf{W}} \leftarrow \mathbf{W} && (4.10)\\
&\qquad \text{Solve Calore for } T\text{:} \quad \mathbf{R}_T(\tilde{\mathbf{W}},T) &&= \mathbf{0} && (4.11)\\
&\qquad\qquad\qquad\qquad\qquad\qquad \Downarrow && \tilde{T} \leftarrow T && (4.12)\\
&\qquad\qquad \text{Evaluate :} \quad \|\mathbf{R}(\mathbf{W},T)\| && && (4.13)
\end{aligned}
$$

where over-tildes indicate values obtained from another application and treated as fixed data during the solution step shown, and eqs.(4.10) and (4.12) represent a transfer of variables to a form needed by the subsequent physics application. The transfers are handled by the Sierra framework. The norm calculation of eq.(4.13) requires re-computation of all physics application residuals using updated variable values. The entire weak coupling cycle is repeated until some measure of overall convergence is attained, e.g. eq.(4.13). Solutions to eqs.(4.9) and (4.11) are obtained using Newton's method for each problem, e.g. for Calore's nonlinear iteration k,

$$
\begin{aligned}
\mathbf{J}_{TT}^{k}\Delta T^{k} &= -\mathbf{R}_{T}^{k} && (4.14)\\
T^{k+1} &= T^{k} + \Delta T && (4.15)
\end{aligned}
$$

Iterations are continued until the norm $\|\mathbf{R}_T\| < \epsilon_T$. For our study, the linear problems of eq.(4.14) were solved iteratively using a Generalized Minimum Residual iterative method. Because one primary objective of our coupling strategies involves enabling coupled multi-physics while minimizing additional requirements on any existing application, we evaluate the Jacobians $\mathbf{J}_{TT}$ and $\mathbf{J}_{\mathbf{WW}}$ numerically using finite-differencing with coloring of the matrix graphs. This feature is available in the NOX nonlinear solver library.

We also applied our JFNK coupling algorithm to solve eq.(4.8). We leveraged the application data used in weak coupling to construct a preconditioning matrix as described in Section 2.3.2.1 which has the following form:

$$
\mathbf{M} = \begin{bmatrix} \mathbf{J}_{\mathbf{WW}} & \\ & \mathbf{J}_{TT} \end{bmatrix} \tag{4.16}
$$

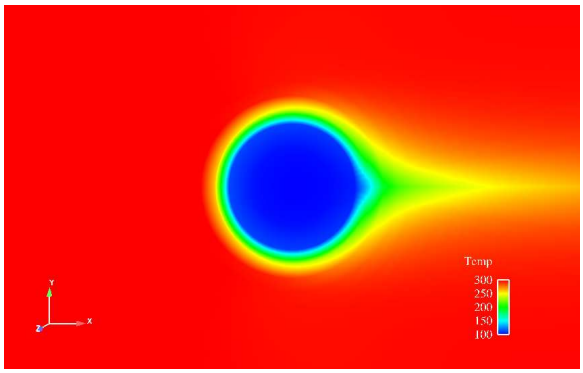
Our implementation of JFNK coupling is consistent with our priority of minimizing requirements for physics codes to participate in a coupled multi-physics simulation. The JFNK operator as well as construction of the preconditioning matrix are all performed using only residual evaluations.

4.1.4 Coupling Results

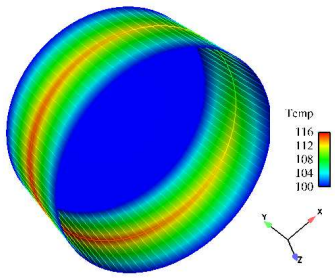
In this section, we provide results for fluid-structural thermal coupling computed using both weak and JFNK coupling strategies for external flows over both the circular cylinder and blunt-wedge bodies.

4.1.4.1 Circular Cylinder

The fluid and solid material properties for the circular cylinder geometry are chosen to reflect a somewhat facile problem in order to ensure converged solutions and thereby allow comparisons between the coupling solution strategies to be made. Accordingly, the flow is characterized by $Re = 374$, $Ma = 0.3$ and $Pr = 120$. The far-field flow is represented by $\mathbf{U}_\infty = \{\rho, u, v, w, p\} = \{$



(a) Fluid temperature along plane $z = 0$

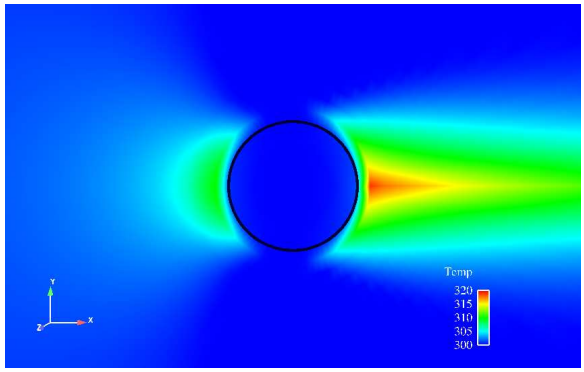


(b) Surface temperature profile.

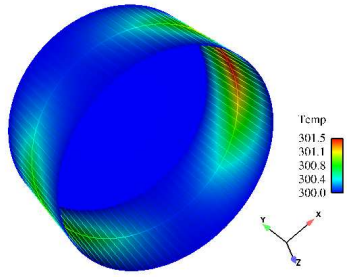
Figure 4.3. Temperature profiles for case with $T_{z=\pm 0.5} = 100.0$.

1.18 kg/m^3, 104 m/s, 0 m/s, 0 m/s, 101325 Pa }. Assuming ideal gas gives $T_\infty = 300K$. The solid circular cylinder has thermal conductivity $\kappa = 401$. A Dirichlet temperature boundary condition $T = T_w$ is imposed on the faces of the circular cylinder at $z = \pm 0.5$. Three cases are considered: $T_w = 100, 300$ and $500K$. Results for the temperature profile on the cylinder surface as well as in the fluid in the vicinity of the cylinder are shown in figures 4.3 - 4.5, respectively. Noteworthy in figures 4.3 - 4.5 are that the temperatures on the fluid/solid interface from the fluid and the solid temperature solutions agree to within the convergence tolerance as they should according to eq.(4.5) and that the fluid solutions **W** differ markedly from those obtained using Premo with either isothermal or adiabatic conditions applied to the cylinder surface. This indicates the physical coupling is non-trivial and suggests improved model quality over the idealized boundary conditions for solid walls available in Premo.

Comparison of the solution convergence behavior using weak and JFNK coupling strategies are shown in figure 4.6 and are summarized with total required CPU times in table 4.1. Included in the figure and table are results for weak coupling performed by starting the successive substitution using either Premo (W-T cycle) or Calore (T-W cycle). For weak coupling the number of nonlinear iterations corresponds to cumulative Premo nonlinear iterations. This choice was made because computational cost is dominated by the Premo problem compared to the Calore problem which comprises only 4443 degrees of freedom out of a total of 53,643 and is linear. Clearly, the performance of weak coupling depends critically on the order of solves within each coupling iteration. JFNK coupling

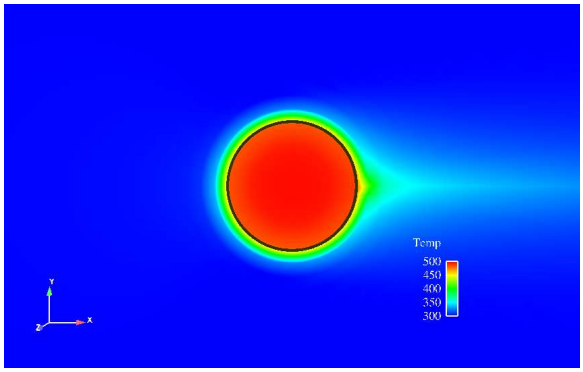


(a) Fluid temperature along plane $z = 0$

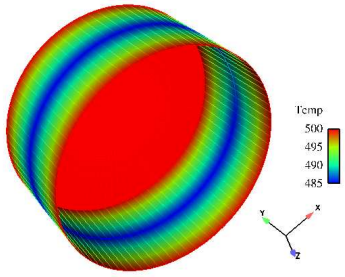


(b) Surface temperature profile.

Figure 4.4. Temperature profiles for case with $T_{z=\pm 0.5} = 300.0$.



(a) Fluid temperature along plane $z = 0$



(b) Surface temperature profile.

Figure 4.5. Temperature profiles for case with $T_{z=\pm 0.5} = 500.0$.

consistently performs slightly better than weak coupling (T-W cycle) in terms of total time required for convergence.

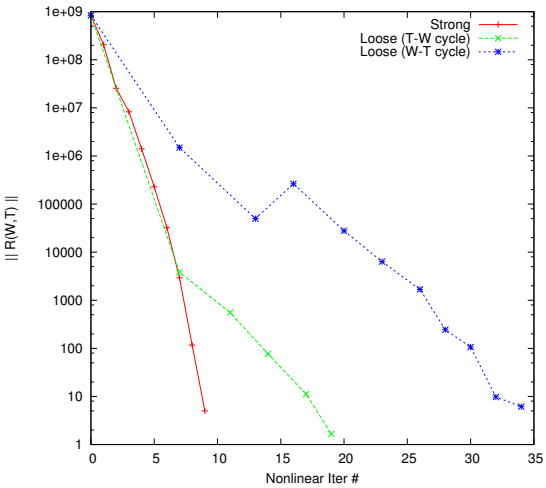
Table 4.1. Comparison of weak coupling and JFNK coupling for the circular cylinder geometry.

Coupling Method	Coupling Iters*	Total Time (CPU s)
$T_{z=\pm 0.5} = 100.0K$		
Weak (W-T cycle)	10	756
Weak (T-W cycle)	5	429
JFNK	9	421
$T_{z=\pm 0.5} = 300.0K$		
Weak (W-T cycle)	8	543
Weak (T-W cycle)	3	276
JFNK	7	267
$T_{z=\pm 0.5} = 500.0K$		
Weak (W-T cycle)	10	750
Weak (T-W cycle)	3	317
JFNK	8	299

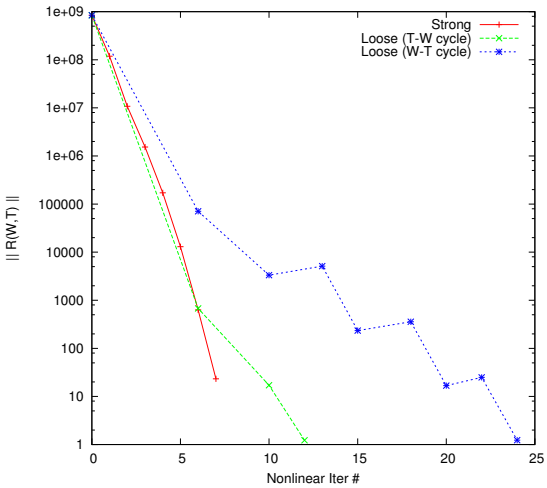
* Total number of coupling iterations. For weak coupling, this is the number of cycles of successive substitution and differs from the number of Premo nonlinear iterations reported in figure 4.6.

4.1.4.2 Blunt-wedge

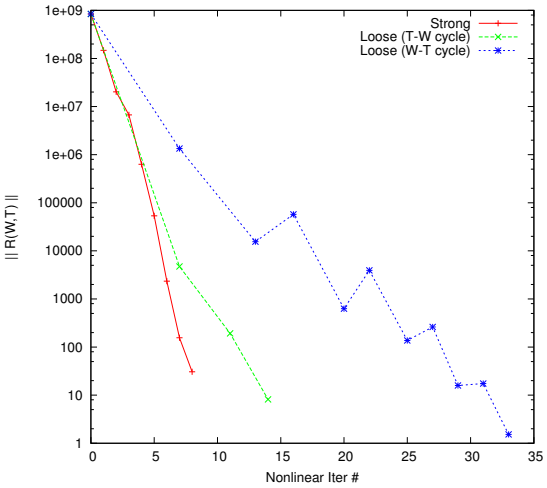
The blunt-wedge geometry was tailored and meshed to capture a $Ma = 3.0$ shock. This problem was made considerably more difficult than the circular cylinder problem by specifying $Re = 2.4x10^5$ and $Pr = 0.72$. Rather than beginning with uniform initial conditions, we ran Premo with an isothermal temperature of $300K$ on the blunt-wedge solid body to obtain a starting solution with the shock qualitatively formed. Coupling to the solid body was then effected by setting the back face of the body to $400K$ and all other surfaces except the fluid/solid interface to have no penetration and slip. The temperature field in the fluid and solid is shown in figure 4.7, and convergence behavior is compared in figure 4.8. For this problem, the quality of convergence is dramatically different with weak coupling (T-W cycle) behaving about as well as for the easier circular cylinder problem but with JFNK coupling exhibiting very poor behavior. Besides more challenging physics, this problem also manifests a non-overlapping surface mesh at the interface, having a solid surface interface meshed at twice the refinement level as the fluid mesh on the interface. This mismatch makes the transfer of variables non-trivial, and the action of the Jacobian performed using directional derivatives now captures the sensitivity of equations near the interface to the details of how the variable transfers are done. Evidence that this may be a source of pollution is observed in the sharp upturns in convergence behavior occurring at nonlinear iterations for which the iterative linear solver failed to achieve the requested linear solve tolerance. This result reveals the importance of how transfers of data are done across non-conforming discrete interfaces. We see further investigation of this within the context of multi-physics coupling algorithms as a beneficial future study.



(a) Convergence comparison for case with $T_{z=\pm 0.5} = 100.0K$.

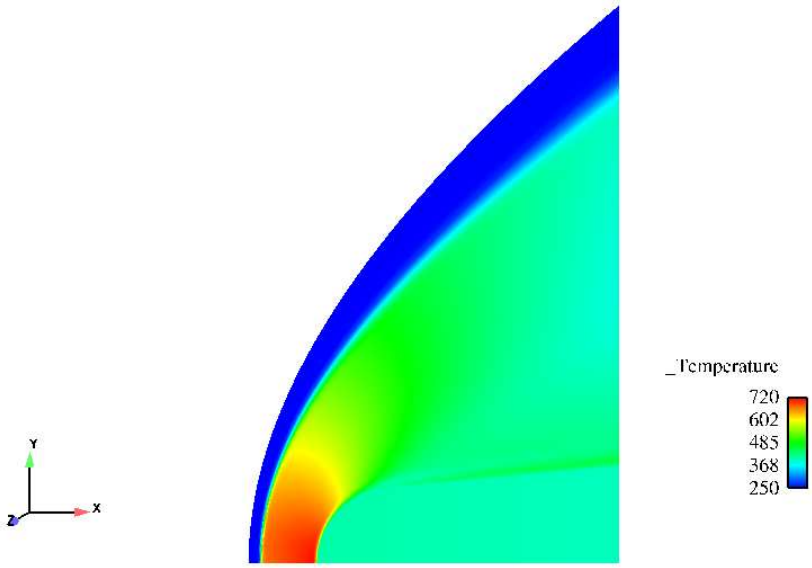


(b) Convergence comparison for case with $T_{z=\pm 0.5} = 300.0K$.

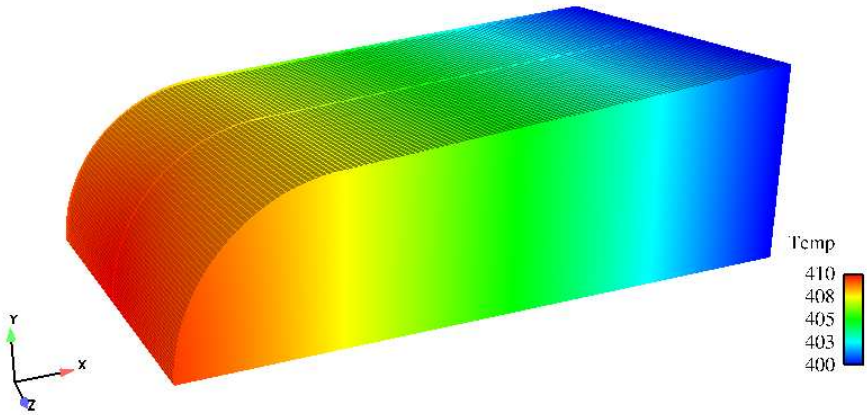


(c) Convergence comparison for case with $T_{z=\pm 0.5} = 500.0K$.

Figure 4.6. Comparison of convergence behavior for weak and JFNK coupling methods represented as coupled residual norm $||\mathbf{R}(\mathbf{W}, T)||$ vs number of nonlinear iterations. Note that the number of nonlinear iterations for weak coupling corresponds to those for the sequence of Premo problems.



(a) Fluid temperature profile.



(b) Surface temperature profile.

Figure 4.7. Temperature profiles for Blunt-wedge problem.

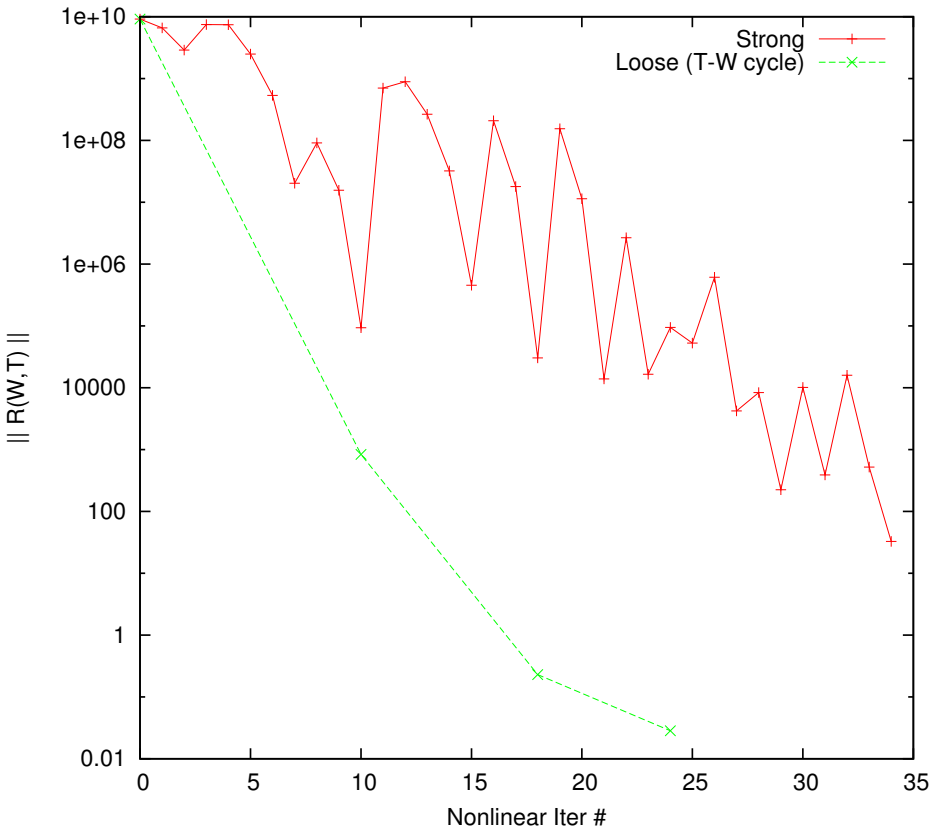


Figure 4.8. Comparison of convergence behavior for weak and JFNK coupling methods represented as coupled residual norm $||\mathbf{R}(\mathbf{W}, T)||$ vs number of nonlinear iterations for the Blunt-wedge problem. Note that the number of nonlinear iterations for weak coupling corresponds to those for the sequence of Premo problems.

4.2 Shared Domain Coupling via Aria

Following the interfacial studies of the previous section, we next sought to impact coupled multi-physics within the same physical domain. We selected Aria, a finite element analysis code for the solution of coupled multi-physics problems, as an ideal target application for several reasons. First, it supports a wide variety of multi-physics internally and can solve these using strong coupling via a pure Newton algorithm as described in Section 2.1. We can then break the multi-physics problem into separate physics each represented by an instance of Aria. This allows both weak coupling and our intermediate coupling algorithms to be used and compared against each other as well as against strong coupling. Also, Aria makes heavy use of Sierra's Solution Control functionality which provides the ideal place to incorporate our intermediate coupling algorithms in order to make them generally available to any other applications using this Sierra functionality. Finally, Aria is actively used by the analysis community to simulate important problems to Sandia's mission.

4.2.1 Problem Description

As a rich and relevant coupled multi-physics problem, we selected a problem related to modeling the micro-electrical-mechanical system (MEMS) shown in Figure 4.9. The device is an electro-thermo-mechanical actuator or switch that utilizes a Silicon Carbide (SiC) material for fast response. The geometry is such that an applied voltage load across the actuator will heat the SiC to produce a thermal expansion which will eventually contact the shuttle with the plate to complete the signal circuit. It represents coupled voltage, temperature and material displacement fields within the same

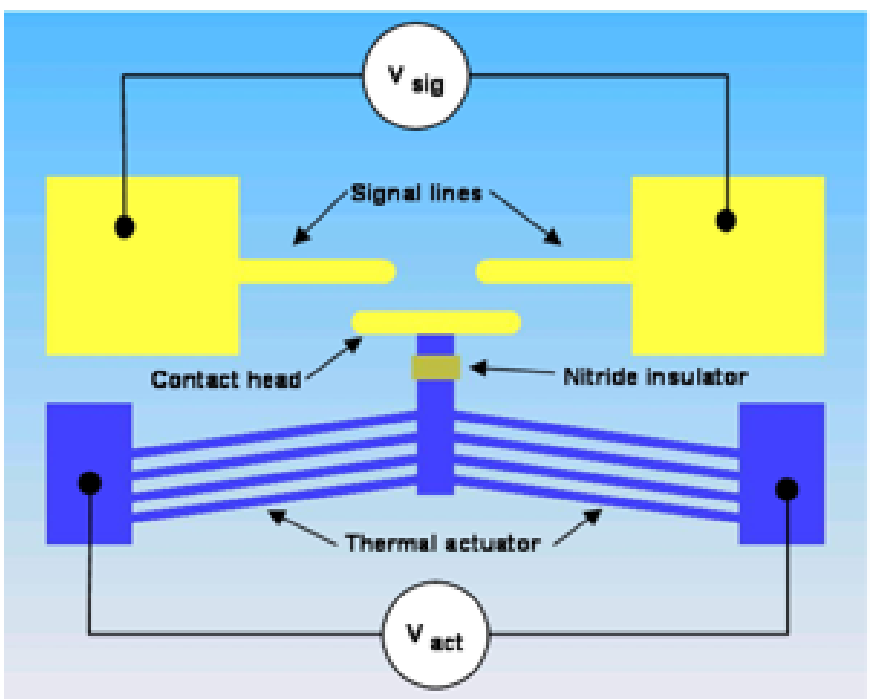


Figure 4.9. MEMS thermal actuator used as a switch.

physical domain. Voltage, V, is modeled by the electrostatic electrical current equation,

$$\nabla \cdot (\sigma \nabla V) = 0, \tag{4.17}$$

where we have rewritten all current fluxes $\mathbf{j}$ in terms of voltages $\mathbf{j} = -\sigma\nabla V$. The electrical conductivity σ depends on temperature as follows:

$$R = \frac{1}{\sigma} = 1.179 \times 10^{-10} \exp(-7.976 \times 10^{-3} T). \tag{4.18}$$

where R is the electrical resistance. This shows that resistance decreases as temperature increases (conductivity increases with temperature), leading to a run-away temperature increase as the voltage drop through the material is increased.

The scalar temperature field, T, is modeled by a steady-state energy transport equation, with a heat source accounting for Joule heating,

$$\boldsymbol{\nabla} \cdot (\kappa \nabla T) = \sigma(\nabla V \cdot \nabla V). \tag{4.19}$$

κ, the thermal conductivity, is equal to 3.33×10^8 $\mathtt{pW}$ $\mu\mathtt{M}^{-1}$ $\mathtt{K}^{-1}$. The term on the right hand side represents Joule heating equivalent to I^2R, where I is the the current. This term exhibits an exponential dependence on temperature and reflects the critical non-linearity in this problem.

Finally, the vector displacement field, $\mathbf{d}$, is found by modeling the material as a linear elastic solid, and solving the steady-state form of the solid momentum equation,

$$\hat{\nabla} \cdot (\lambda \ \mathrm{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} - \beta(T - T_{ref})\mathbf{I}) = 0, \tag{4.20}$$

where λ and μ are the Lame coefficients, $\mathbf{E}$ is the deformation tensor, β is the coefficient of thermal stress, and T_{ref} is the solid stress reference temperature and is set to 20 C. The operator $\hat{\nabla}$ is the gradient operator in the undeformed state, e.g. $\mathbf{d} = \mathbf{0}$. The relationships between these parameters and the material properties is given by:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{4.21}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{4.22}$$

$$\beta = \frac{\alpha E}{1-2\nu)}. \tag{4.23}$$

The deformation tensor is defined as $\mathbf{E} = (\hat{\nabla}\mathbf{d} + \hat{\nabla}\mathbf{d}^{\mathbf{T}})/2$. The linear thermal stress causes material deformation due to the temperature field. A summary of the dependencies between equations is as follows:

a) The current equation, eq.(4.17), depends on the deformed state of the material $\mathbf{d}$ found by solving the displacement equation, eq.(4.20). The current equation is otherwise linear.

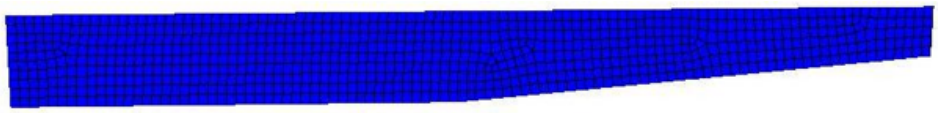
b) The energy equation, eq.(4.19), depends on the deformed state of the material $\mathbf{d}$ found by solving the displacement equation.

c) The energy equation depends on the voltage V via the Joule heating source term. Because the electrical conductivity σ is exponentially dependent on temperature, the possibility of a runaway temperature condition exists.

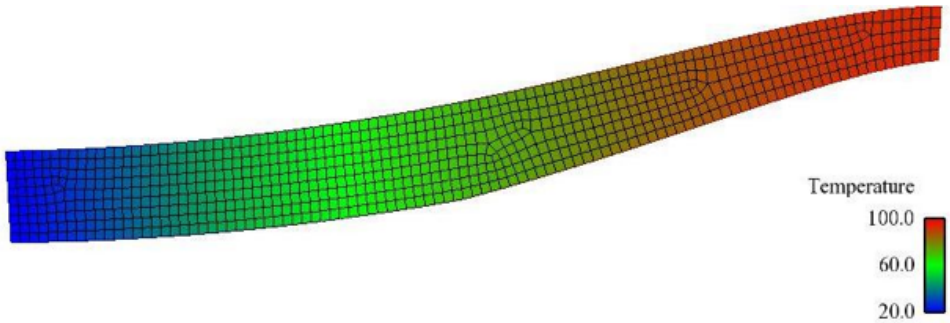
d) The energy equation, eq.(4.19), has a material property dependence on temperature T and represents another nonlinearity in the coupled problem.

e) The displacement equation, eq.(4.20), is dependent on the temperature T through the thermal stress term.

f) Because all of the differential operators in the displacement equation are in the undeformed state, it is linear in its variable and linear overall.



(a) Aria 2D MEMS actuator mesh before voltage loading.



(b) Aria 2D MEMS actuator after voltage loading with $V = 1.934V$.

Figure 4.10. Aria 2D MEMS actuator after voltage loading with $V = 1.934V$. Displacements have been amplified by a factor of 300 for aiding visualization.

We consider a two-dimensional representation of the MEMS thermal actuator of Figure 4.9 and also take advantage of the vertical symmetry that exists through the middle of the shuttle. The resulting mesh colored by temperature before and after application of a voltage load is shown in figure 4.10

4.2.2 Coupling Results

Our selection of Aria for our coupling studies was partly due to its ability to perform strong coupling via a pure Newton algorithm (cf Section 2.1). We consider this the gold standard coupling algorithm and seek to approach its performance using our intermediate coupling algorithms with weak coupling application data. Using Aria's strong coupling for the MEMS actuator problem allows us to identify limits to convergence arising from problem complexity. As shown in Figure 4.11, pure Newton coupling achieves converged solutions up to a voltage loading of $\Delta V = 2.879$ with increased difficulty reflected in more nonlinear iterations as this limit is approached.

Given voltage loading as a means of tuning problem difficulty, we next compared weak coupling and variations of our intermediate coupling algorithms. Table 4.2 shows a summary of our results. The second and third bi-columns of Table 4.2, named "Jacobi" and "Broyden", represent approximate matrices described in Sections 2.3.2.1 and 2.3.2.3, respectively, used as the Jacobian matrix $\mathbf{J}$. The next two bi-columns, "JFNK w/ Jacobi" and "JFNK w/ Broyden", employ the JFNK Jacobian operator of eq.(2.22) with the approximate matrices used as a preconditioning matrix, $\mathbf{M}$. The "Jacobi" matrix approximations have the structure shown in Figure 4.12a and consist of an aggregated composite of stand-alone Jacobian matrices from each independent Aria physics. The upper-left block has dimensions twice the other two blocks and represents the stand-alone mesh displacement problem Jacobian, while the other two blocks correspond to the stand-alone temperature and current problem Jacobians. We added the ability to augment the structure of the coupled problem Jacobian to account for accommodate non-zero entries associated with inter-problem dependencies

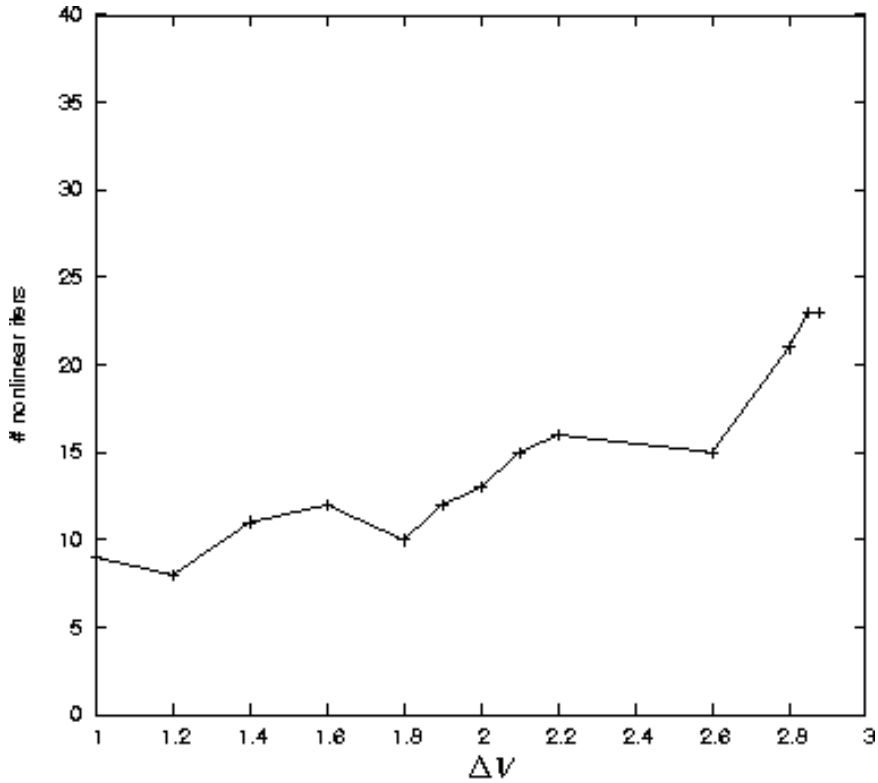


Figure 4.11. Convergence behavior for pure Newton coupling in Aria. The plot shows the number of Newton iterations required to achieve convergence for various applied voltage loads. Convergence failed for $\Delta V > 2.879$.

	Jacobi		Broyden		JFNK w/ Jacobi		JFNK w/ Broyden		Pure Newton	
ΔV	NL iters	Lin iters	NL iters	Lin iters	NL iters	Lin iters	NL iters	Lin iters	NL iters	Lin iters
1.000	12	800	18	1323	9	597	9	681	9	646
1.200	15	1023	14	993	8	521	8	607	8	576
1.400	16	1065	21	1498	11	777	11	791	11	825
1.600	19	1239	23	1779	11	707	11	865	12	915
1.800	30	2007	22	1578	9	572	9	711	10	775
1.900	36	2401	21	1467	11	729	11	811	12	903
2.000	FAIL		24	1673	12	847	12	995	13	1020
2.100			38	2704	13	814	13	887	15	1130
2.200			FAIL		16	1091	15	1100	16	1198
2.600					15	1019	15	1315	15	1124
2.800					20	1279	21	1574	21	1509
2.850					23	1676	22	1748	23	1754
2.878					24	1743	24	1915	23	1857
2.879					FAIL		FAIL		FAIL	

Table 4.2. Number of nonlinear and total linear iterations required to converge the MEMS actuator problem at various voltage loadings up to the critical value of $\Delta V = 2.879$.

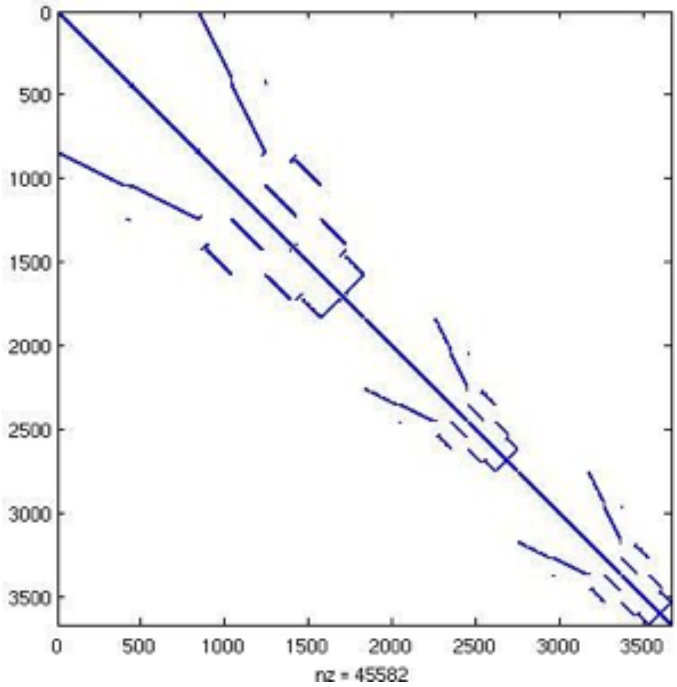
as shown in Figure 4.12b. The "Broyden" matrix approximation fills only these augmented locations using the approach described in Section 2.3.2.3 and fills the diagonal blocks with the stand-alone Jacobian matrices as in the "Jacobi" approximation. The problem is steady-state so that the number of nonlinear iterations is for a single solve and provides an indication of the robustness of the coupling algorithm employed. The number of linear iterations is the cumulative total for all nonlinear iterations for a given voltage loading and coupling algorithm and provides an indication of the computational cost of the algorithm. The results of Table 4.2 show that JFNK coupling using either preconditioning matrix performs essentially on par with strong, pure Newton coupling. The desirable qualities of both robustness and efficiency of the JFNK coupling algorithm are achieved using only weak coupling application information.

Additional insight into the improvement our intermediate coupling algorithms over weak coupling can be seen in the rate of convergence shown for each in Figure 4.13 Here again we see a linear rate of convergence for weak coupling and a quadratic rate for our Newton-based JFNK coupling algorithm.

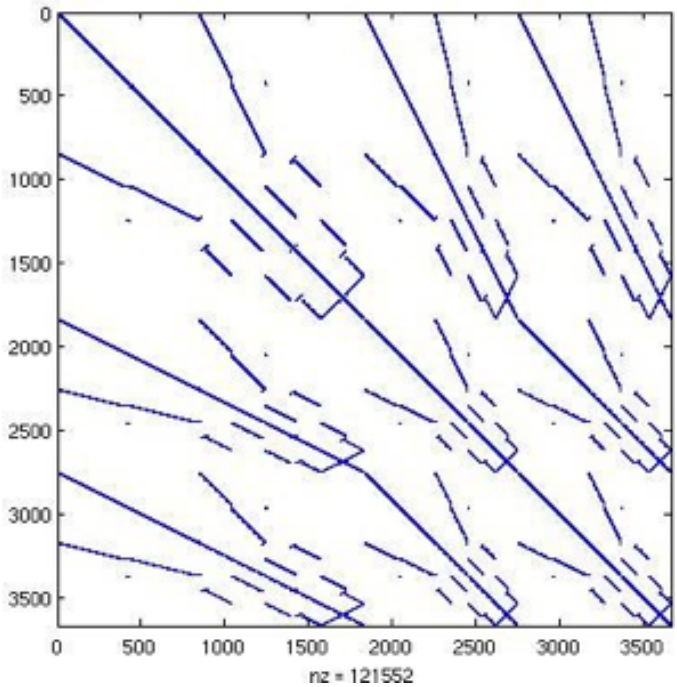
In summary for our coupling studies in Aria, we have effectively enabled Newton-based coupling using only weak coupling application data. We have approached the task of implementation with an eye toward general usefulness and applicability by incorporating our intermediate coupling algorithms into Solution Control in Sierra.

4.3 Conclusions

For non-trivial relevant multi-physics coupling scenarios encompassing the two canonical types of problem coupling, i.e. coupling across a shared interface and coupling within a shared physical domain, we have shown that Newton-type convergence quality can be achieved while requiring little to no additional application information beyond the minimal amount needed for weak coupling. We have shown that our new coupling algorithms apply well to fundamentally different physics represented by parabolic, elliptic and hyperbolic partial differential equations. Moreover, we have shown the value-added benefits of our coupling algorithms in that any additional information applications can supply beyond that needed for weak coupling can be employed to construct a better, less approximate preconditioning matrix.



(a) Diagonal block structure of "Jacobi" matrix approximation consisting of stand-alone Jacobian matrices from each Aria physics.



(b) Complete block structure coupled multi-physics MEMS problem. The additional non-zero entries beyond the "Jacobi" structure are filled using secant updates as in Section 2.3.2.3.

Figure 4.12. Non-zero structure of "Jacobi" (top) and "Broyden" (bottom) matrix approximations.

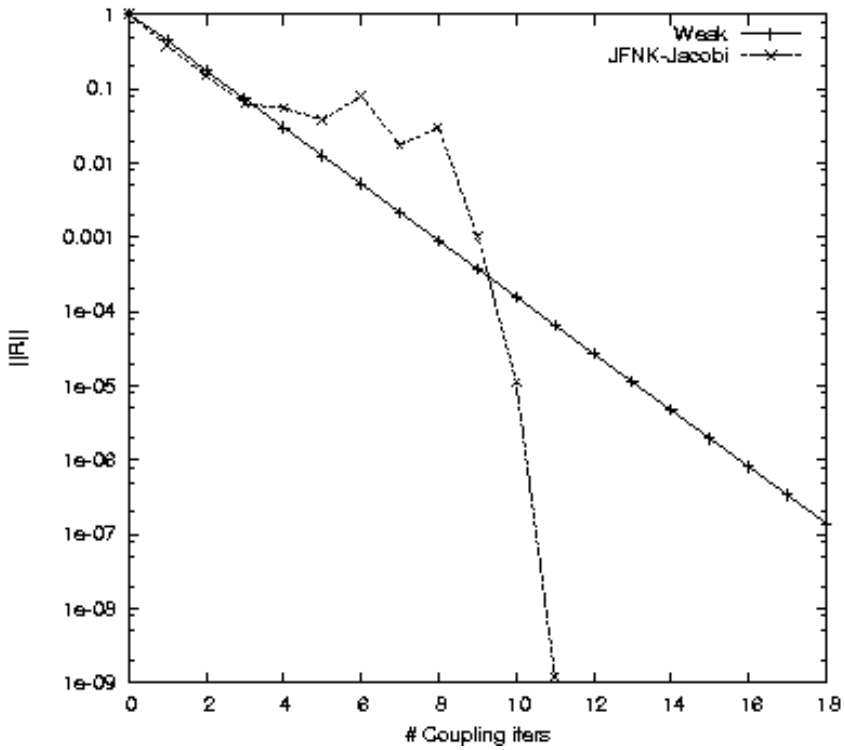


Figure 4.13. Convergence rate behavior for weak and JFNK coupling in Aria for $\Delta V = 1.7V$.

Chapter 5

Analysis and Observations

5.1 Introduction and Setup of the Theory

Here we establish an approach for analyzing the performance of various coupling algorithms. Having summarized various approaches to coupling in Chapter 2, we now wish to explore certain salient features in greater detail.

For purposes of discussion, we consider finding a solution to a multi-physics problem consisting of coupled A and B problems. As shown in Section 2.1, we may formulate Newton's method for the coupled problem as the solution of the following matrix equation.

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{J}_{AB} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^n - \mathbf{x}_A^{n-1} \\ \mathbf{x}_B^n - \mathbf{x}_B^{n-1} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_A \\ \mathbf{R}_B \end{bmatrix} \tag{5.1}$$

$\mathbf{J}_{AB}$ and $\mathbf{J}_{BA}$ represent coupling terms between the two A and B problems which typically cannot be provided by either problem being coupled and are unavailable in general.

We begin by assuming that there exists efficient strategies for solution of the separate A and B problems using Newton's method or a variant of Newton's method. In other words, the following equations may be readily solved:

$$\Delta \mathbf{x}_A^n = -\mathbf{J}_{AA}{}^{-1} \mathbf{R}_A^{n-1} \tag{5.2}$$

$$\Delta \mathbf{x}_B^n = -\mathbf{J}_{BB}{}^{-1} \mathbf{R}_B^{n-1} \tag{5.3}$$

In contrast, solution of the coupled system of eq..(5.1) encounters severe difficulties arising from such issues discussed in Chapter 2. Even when Newton-based coupling algorithms can be used, it can still be more desirable to employ weak coupling. This can be the case if the loss of sparsity of the coupled Jacobian due to the off-diagonal blocks is dramatic or if very specialized methods have been developed to solve each application stand-alone. Then, forcing a Newton-based approach can represent a loss of desirable capability or performance. Weak coupling can also be much more efficient when the physics is only loosely coupled, e.g. one-way coupling in which case it makes total sense to solve the upstream problem first followed by the downstream dependent problem next without repeating the cycle.

When using weak coupling there are two modes of iterating. When applied as a Jacobi method, weak coupling involves solving eqs.. (5.2) and (5.3) for the coupled A -B system, when they are applied successively and with no intermediate transfer of updated variables until convergence is achieved. The Gauss-Seidel mode of weak coupling for eqs. (5.4) and (5.5) involves first solving the A problem variables and then the B problem after substituting in the just-updated solution from the A problem and repeating until convergence is achieved. As predicted form fixed-point theory (cf Section 2.2) and as observed in studies involving prototype and real applications problems, convergence rates for both modes of weak coupling are only linear.

$$\Delta \mathbf{x}_A^{n-1} = -\mathbf{J}_{AA}{}^{-1} \mathbf{R}_A^{n-1}(\mathbf{x}_A^{n-1}, \mathbf{x}_B^{n-1}) \tag{5.4}$$

$$\Delta \mathbf{x}_B^n = -\mathbf{J}_{BB}{}^{-1}\mathbf{R}_B^{n-1}(\mathbf{x}_A^n, \mathbf{x}_B^{n-1}) \tag{5.5}$$

As a means of analyzing weak coupling, the problem A part of eq. (5.1) may be formally solved in terms of the delta unknowns in problem B, $\Delta \mathbf{x}_B^n$. Then, plugging the results back into the problem B part of eq. (5.1) results in the following reduced equation for the update vector, $\Delta \mathbf{x}_B$.

$$\mathbf{S}_{AA}\Delta \mathbf{x}_B = -\mathbf{R}_B + \mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{R}_A \tag{5.6}$$

where

$$\mathbf{S}_{AA} = \mathbf{J}_{BB} - \mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{J}_{AB} \tag{5.7}$$

is the Schur complement of the A problem. Looking at eq. (5.6), we may think of the update to $\Delta \mathbf{x}_B$ in the coupled problem as being formed from two contributions. The first is from $\mathbf{R}_B$, while the second is from $\mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{R}_A$. Taking norms of the vectors and matrices may lead to a metric for determining when coupling terms are important. $\mathbf{S}_{AA}$ may be further interpreted. When there is zero or one way coupling, $\mathbf{S}_{AA}$ is the same as the uncoupled case, because either $\mathbf{J}_{AB}$ or $\mathbf{J}_{BA}$ is zero. Only in the instance of two way coupling is $\mathbf{S}_{AA}$ non-zero.

We may continue from eq. (5.6) to find an expression for the update to $\Delta \mathbf{x}_A$, eq. (5.8), since we now have an expression for $\Delta \mathbf{x}_B$.

$$\mathbf{J}_{AA}\Delta \mathbf{x}_A = -\mathbf{R}_A + \mathbf{J}_{AB}\mathbf{S}_{AA}{}^{-1}\mathbf{R}_B - \mathbf{J}_{AB}\mathbf{S}_{AA}{}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{R}_A \tag{5.8}$$

We may think of the update to $\Delta \mathbf{x}_A$ as being formed from three contributions. The first contribution is from $\mathbf{R}_A$, while the second contribution is from $\mathbf{J}_{AB}\mathbf{S}_{AA}{}^{-1}\mathbf{R}_B$. This is the direct effect that residual errors in problem B have on unknowns in problem A. The third contribution is from the completed cross-coupling effect. Errors in problem A affect the variables in problem B, which in turn affect the variables in problem A. The third contribution is only active if both $\|\mathbf{J}_{AB}\| > 0$ and $\|\mathbf{J}_{BA}\| > 0$, and is only important if both of these off-diagonal matrices exhibit significant norms. Taking norms of the vectors and matrices of the three contributions lead to a metric for when the coupling terms are important.

It should also be noted here that the condition numbers of both individual problems A and B may greatly influence the convergence results of the coupled problem. For example, eq.(5.6) demonstrates that an ill-conditioned matrix for problem A will result in a large or even unbounded perturbation to the solution of problem B through it's influence on the rhs of eq.(5.6). The norm of the inverse of $\mathbf{J}_{AA}$ is proportional to the condition number, κ_{AA}, of the block-diagonal matrix $\mathbf{J}_{AA}$.

$$\left\|\mathbf{J}_{AA}^{-1}\right\| = \frac{\kappa_{AA}}{\|\mathbf{J}_{AA}\|} \tag{5.9}$$

From eq.(5.6) this implies that if $\mathbf{J}_{BA}$, i.e. the dependence of B on A, is of sufficient size, then the accuracy in the solution of B will be dominated by the condition number for problem A. Generalizing, the number of significant digits that may be accurately computed in a coupled problem will be a function of the amount of ill-conditioning in the worst of the individual problems.

5.2 Analysis of Weak Coupling

We can analyze the successive substitution strategy in terms of related quantities used in the analysis of the Schur components. We will start with Newton's method, eq. (5.1). Newton's method is expressed in terms of updates to problem variables , i.e. $\Delta \mathbf{x}_A$ and $\Delta \mathbf{x}_B$. Since we need a recursion relation to understand successive substitution, Newton's method must be rearranged into an equation for the solution variables, $\mathbf{x}_A$ and $\mathbf{x}_B$. This is easily done by expanding the definition of $\Delta \mathbf{x}_A =$

$\mathbf{x}_A^n - \mathbf{x}_A^{n-1}$ and $\Delta\mathbf{x}_B = \mathbf{x}_B^n - \mathbf{x}_B^{n-1}$, and then reformulating the problem and the rhs into a picard factorization, eq.(5.10), where we have renamed the expanded rhs's as $\mathbf{R}_A^p$ and $\mathbf{R}_B^p$.

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{J}_{AB} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^n \\ \mathbf{x}_B^n \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_A - \mathbf{J}_{AA}\mathbf{x}_A^{n-1} - \mathbf{J}_{AB}\mathbf{x}_B^{n-1} \\ -\mathbf{R}_B - \mathbf{J}_{BA}\mathbf{x}_A^{n-1} - \mathbf{J}_{BB}\mathbf{x}_B^{n-1} \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_A^p \\ -\mathbf{R}_B^p \end{bmatrix} \tag{5.10}$$

Now, suppose it is necessary to solve eq.(5.10), which is a linear set of equations, via a successive substitution strategy, where for now the non-linearity in the underlying problems is ignored, i.e. we don't recompute $\mathbf{R}_A^p$ and $\mathbf{R}_B^p$. We begin by making an initial guess $\mathbf{x}_B^0$ and substituting $\mathbf{x}_B^0$ into the problem A equation system. Then, we substitute the resulting expression for $\mathbf{x}_A^1$ into the problem B equation system to obtain $\mathbf{x}_B^1$.

$$\begin{bmatrix} \mathbf{x}_A^1 \\ \mathbf{x}_B^1 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{AA}^{-1}(-\mathbf{R}_A^p - \mathbf{J}_{AB}\mathbf{x}_B^0) \\ \mathbf{J}_{BB}^{-1}(-\mathbf{R}_B^p - \mathbf{J}_{BA}\mathbf{x}_A^1) \end{bmatrix} \tag{5.11}$$

We can rewrite the B equation as

$$\begin{aligned} \mathbf{x}_B^1 = \mathbf{J}_{BB}^{-1}(-\mathbf{R}_B^p - \mathbf{J}_{BA}\mathbf{x}_A^1) &= -\mathbf{J}_{BB}^{-1}\mathbf{R}_B^p - \mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}^{-1}(-\mathbf{R}_A^p - \mathbf{J}_{AB}\mathbf{x}_B^0) \\ &= \left(-\mathbf{J}_{BB}^{-1}\mathbf{R}_B^p + \mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}^{-1}\mathbf{R}_A^p\right) + \left(\mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}^{-1}\mathbf{J}_{AB}\right)\mathbf{x}_B^0 \\ &= \mathbf{c}_B + \mathbf{G}_{BB}\mathbf{x}_B^0 \end{aligned} \tag{5.12}$$

where

$$\mathbf{G}_{BB} = \mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}^{-1}\mathbf{J}_{AB}. \tag{5.13}$$

If we apply the successive substitution algorithm recursively k times, eq. (5.14) results.

$$\mathbf{x}_B^k = \mathbf{G}_{BB}^k\mathbf{x}_B^0 + \left(\sum_{i=0}^{k-1}\mathbf{G}_{BB}^i\right)\mathbf{c}_B \tag{5.14}$$

$\mathbf{G}_{BB}$ is directly related to the modification of the Schur complement for problem B, $\mathbf{S}_{AA}$, from its strictly block diagonal form, $\mathbf{J}_{BB}$, due to the interaction of the problem A equation system:

$$\begin{aligned} \mathbf{S}_{AA} = \mathbf{J}_{BB} - \mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{J}_{AB} &= \mathbf{J}_{BB}(\mathbf{I} - \mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}{}^{-1}\mathbf{J}_{AB}) \\ &= \mathbf{J}_{BB}(\mathbf{I} - \mathbf{G}_{BB}) \end{aligned} \tag{5.15}$$

A sufficient condition for convergence of the successive substitution algorithm, eq.(5.14), for the linearized coupled system, is that the spectral radius of $\mathbf{G}_{BB}$, a dimensionless quantity both row-wise and column-wise, must be less than 1. What this means is that the all of the eigenvalues of $\mathbf{G}_{BB}$ must be within the unit circle on the complex plane. A sufficient but not necessary condition is that the L2 matrix norm of $\mathbf{G}_{BB}$ (or equivalently the maximum eigenvalue from the singular value decomposition of $\mathbf{G}_{BB}$) be less than one. In Chapter 7, we will encounter a case where the former is true, but the latter is not.

To see this, we may note that eq.(5.14) may be reduced to the following, when the spectral radius condition is satisfied and therefore the sum in eq.(5.14) may be replaced by its limiting condition.

$$\mathbf{x}_B^k = \mathbf{G}_{BB}^k\mathbf{x}_B^0 + (I - \mathbf{G}_{BB})^{-1}\mathbf{c}_B \tag{5.16}$$

This theory may be numerically verified using Matlab, where calculation of the block matrices and their norms may be carried out explicitly for small problems. The L2 matrix norm may be used as a good, but not foolproof proxy for the calculation of the spectral radius, which requires the

calculation of all of the eigenvalues. Note that the L2 matrix norm being less than one is not in fact a necessary condition for eq.(5.14) to be convergent. We will encounter such a case in Section 7.5.2, where it turns out even though the L2 matrix norm is greater than one, all of the eigenvalues are within the complex unit circle.

To verify that $\mathbf{G}_{BB}$ is in fact the fixed-point iterative matrix for eq.(5.14), we shall next derive a slightly different but equivalent form.

We may rewrite the successive substitution process via the following formulation.

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{0} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^n \\ \mathbf{x}_B^n \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_A^p \\ -\mathbf{R}_B^p \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{J_{AB}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^{n-1} \\ \mathbf{x}_B^{n-1} \end{bmatrix} \tag{5.17}$$

We note that the eventual solution of the matrix problem is eq. (5.10) and then substitute it into the equation above to yield,

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{0} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^n \\ \mathbf{x}_B^n \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{AA} & \mathbf{J_{AB}} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^s \\ \mathbf{x}_B^s \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{J_{AB}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^{n-1} \\ \mathbf{x}_B^{n-1} \end{bmatrix} \tag{5.18}$$

$\mathbf{x}_A^s$ and $\mathbf{x}_B^s$ is the steady state solutions for eq. (5.10). This may be rearranged to yield

$$\begin{bmatrix} \mathbf{J}_{AA} & \mathbf{0} \\ \mathbf{J}_{BA} & \mathbf{J}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^n - \mathbf{x}_A^s \\ \mathbf{x}_B^n - \mathbf{x}_B^s \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{J_{AB}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^{n-1} - \mathbf{x}_A^s \\ \mathbf{x}_B^{n-1} - \mathbf{x}_B^s \end{bmatrix}. \tag{5.19}$$

This equation may be inverted to yield

$$\begin{bmatrix} \mathbf{x}_A^n - \mathbf{x}_A^s \\ \mathbf{x}_B^n - \mathbf{x}_B^s \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{AA}^{-1} & \mathbf{0} \\ -\mathbf{J}_{BB}^{-1}\mathbf{J}_{BA}\mathbf{J}_{AA}^{-1} & \mathbf{J}_{BB}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{J_{AB}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^{n-1} - \mathbf{x}_A^s \\ \mathbf{x}_B^{n-1} - \mathbf{x}_B^s \end{bmatrix} \tag{5.20}$$

and then

$$\begin{bmatrix} \mathbf{x}_A^n - \mathbf{x}_A^s \\ \mathbf{x}_B^n - \mathbf{x}_B^s \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{J}_{AA}^{-1}\mathbf{J_{AB}} \\ \mathbf{0} & \mathbf{G}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_A^{n-1} - \mathbf{x}_A^s \\ \mathbf{x}_B^{n-1} - \mathbf{x}_B^s \end{bmatrix}. \tag{5.21}$$

Therefore, $\mathbf{G}_{BB}$ is in fact the fixed-point iterative matrix for eq.(5.14). In the above formulation, we assumed that problem A would be solved first and then problem B next. This created the above situation where the convergence of the fixed point algorithm is expressed in terms of the convergence of the B problem. The convergence of the A problem, however, is closely tied to the B problem. Also, the order of operations could be reversed as well, and the analysis redone. In this case, a $\mathbf{G}_{AA}$ would be formulated which would be in all cases similar to $\mathbf{G}_{BB}$ in terms of its convergence behavior. In other words, we don't change the convergence analysis in any significant way by assuming that either the A or the B problem goes first. However, in practice the breakdown of our assumptions can lead to dependence of convergence on the order of problems being solved as seen in Section 4.1.4.1.

5.3 Conclusion

This chapter has revisited weak coupling in greater detail and has identified some key quantities that may provide cheap assessment metrics of degree of dependence among coupled physics applications and accordingly suggest a "best" choice coupling algorithm. As will be shown in the next few chapters, these metrics can provide quantitative assessments of the feasibility of using weak coupling and whether a stronger coupling algorithm should be used. As opportunities arise, we hope to incorporate most or all of these into our coupling algorithms extensions to the NOX nonlinear solver library.

Chapter 6

Numerical Analysis of Coupling Using Matlab

Frequently, the choice of the most cost-effective strategy for solving nonlinear coupled multi-physics problems hinges on how tight the coupling is among the participating physics. This gives rise to the concept of "degree" of coupling between problems. In this chapter, we present an explicit analysis on a sample problem using Matlab and Aria. The analysis will show that the spectral radius of the matrix, $\mathbf{G_{BB}}$ defined in eqn.(5.13) of Section 5.2 determines the convergence behavior of the successive substitution strategy for solving the coupled problem.

To elucidate this point, we use the Aria MEMS thermal actuator problem described in Section 4.2.1. Using Matlab, we explicitly calculate and manipulate the matrices that arise in the analysis presented in Chapter 5. We show that the convergence of weak coupling may be completely determined from within Matlab by finding the spectral radius of the matrix $\mathbf{G_{BB}}$. We also document the input files used in several coupling strategies within Sierra. Examples of configuring Jacobi and Gauss-Seidel modes of weak coupling as well as JFNK coupling using multiple Aria regions via Sierra's Solution Control capability are presented. We verify that the JFNK approach using a preconditioning matrix based on the block diagonal Jacobians collected from each Aria problem being coupled is a robust and efficient strategy for solving coupled multi-physics problems.

6.1 Matrix Calculations Using MATLAB

Our numerical analysis of convergence properties of various coupling algorithms is carried out by extracting and writing to files the needed matrices from the Sierra and NOX environments and then reading them in and analyzing them from within the Matlab environment. The required matrix input-output functionality is provided through the Finite Element Interface (FEI) connecting Sierra applications such as Aria to solvers in Trilinos. In particular, we employed the FEI option "FEI_OUTPUT_LEVEL = matrix_files" to output Jacobian matrices in matrix-market format to be read into Matlab.

We used the option, " FEI_OUTPUT_LEVEL", to dump out the last newton iteration of an Aria multi-physics problem using pure Newton coupling. This gave the correct complete and accurate Jacobian matrix at the converged solution. We then extract the individual matrix blocks, $\mathbf{J}_{AA}$, $\mathbf{J}_{AB}$, $\mathbf{J}_{BA}$, and $\mathbf{J}_{BB}$ from the full coupling Jacobian matrix using Matlab scripts, which are based on the assumption that there is a constant stride in the global solution vector between temperature, voltage, and displacement unknowns.

Once in Matlab, the matrices can easily and quickly be manipulated and analyzed using a vast set of utilities. For example, we explicitly calculate the matrix $\mathbf{G}_{BB}$ given in eqn.(5.13), as well as the norms and condition numbers for $\mathbf{J}_{AA}$, $\mathbf{J}_{AB}$, $\mathbf{J}_{BA}$, and $\mathbf{J}_{BB}$. We also calculate the norm, the spectral radius, and the leading eigenvalues for the matrix $\mathbf{G}_{BB}$, results we will present in Section

6.2.1.

In passing, we note that we explicitly calculated much more than G_{BB} within Matlab. In particular we calculated values and norms for S_{AA} and for all three terms in eqn. (5.8). We will not belabor the details other than to note that the estimation of the magnitude of these terms depended greatly upon scaling issues. Both row and column scaling of eqn.(5.1) was attempted to get accurate estimates of the relative magnitudes of terms in eqn.(5.8). Column scaling involves creating a right-hand scaling vector based on the weighting function used to judge convergence in the solution unknowns (see [(10)] for an example). Creating the column scaling matrix therefore involves importing the global solution vector into Matlab. However, we encountered problems with this stemming from our use of an experimental interface to NOX, which supported the option of dumping the solution vector out in matrix market format. We therefore see the need to maintain closer concurrency between the applications environment and the algorithms environment. We anticipate revisiting this issue in the future and advocate maintenance and enhancement of capability such as this. The end result of the analysis of norms for S_{AA} is that the direct calculation of G_{BB} produced the only acceptable test for the efficiency of the weak coupling algorithm.

6.2 Example: A Coupled Voltage, Displacement, and Temperature MEMS Problem

We performed our numerical experiments in Matlab using data generated from the MEMS thermal actuator problem described in Section 4.2.1 and depicted in Figure 4.10. Figure (6.1) contains the input deck to Aria. We have left the NOX solver block, named `NOX_Solver`, out of Figure (6.1) for space reasons. However, `NOX_Solver` implements a GMRES iterative solver using an ILU preconditioner with zero fill-in.

To analyze this MEMS thermal actuator problem, we have identified problem A as the temperature equation, eqn.(4.19, and problem B as being the combined voltage and displacement equations, eqs..(4.17) and (4.20).

6.2.1 Weak Coupling Results

In this section we report attempts to identify the onset of convergence failure for Jacobi and Gauss-Seidel weak coupling algorithms using the voltage load as a dial for problem difficulty.

Gauss-Seidel and Jacobi weak coupling were driven at the Solution Control level of Sierra. Solution Control supports script-based algorithms which include Gauss-Seidel and Jacobi weak coupling. The steps needed to perform Jacobi weak coupling in the parlance of Solution control are shown in Figure (6.2).

The solution control block, a transfer region, and the Aria region that handles the separate solution of the temperature equation (with all other variables transfered in and not solved for) is shown. Regions that solve separately for the voltage (the `V_Region`) and for the displacement field (the `D_Region`) and transfer blocks that describe transfers of variables between other regions are completely analogous and are not shown.

The solution block contains the `Nonlinear` block which creates a looping condition that must be completed with stopping criteria. The stopping criteria are supplied in the `Parameters For Nonlinear` block where the convergence criteria for the loop is specified. The convergence criteria is based either on the Sierra Global Variable named `CURRENT_STEP` reaching 80 or the Sierra global variable named `Region_Nonlinear_Residual` in the `Global_Norm_Region` region being less than 1.0×10^{-3}.

```
# SiC (Silicon Carbide) material properties.
#
# Rho    = {rho = 8.784e-15}
# YM     = {YM = 4.50e5}
# PR     = {PR = 0.16}
# Lambda = {lambda = PR*YM/((1+PR)*(1-2*PR))}
# Mu     = {mu = YM /( 2*(1+PR) )}
# Alpha  = {alpha = 4.6e-6}
# Beta   = {beta = YM*alpha/(1-2*PR)}
#
# T_0    = {T_0 = 20} Celsius
# V_0    = {V_0 = 0} V
# DeltaV = {DeltaV = 1.0} V
#
Begin Sierra Aria_MEMS_Beam
   Begin Aria Material SiC
      Density                 = CONSTANT rho    = {rho}
      Mesh Stress             = LINEAR_ELASTIC reference_frame = undeformed
      Mesh Stress             = THERMAL
      Lambda                  = CONSTANT lambda = {lambda}
      Two Mu                  = CONSTANT two_mu = {2 * mu}
      Beta                    = CONSTANT beta = {beta}
      Current Density         = Ohms_Law
      Electrical Resistance   = EXPONENTIAL variable=temperature multiplier=1.179e-10 exponent=-7.976e-3
      Electrical Conductivity = From_Resistance
      Heat Conduction         = Fouriers_Law
      Specific Heat           = EXPONENTIAL variable=temperature constant=1.5e+15 multiplier=7.625e+14 exponent=-1.167e-3
      Thermal Conductivity    = CONSTANT k=3.33e+8
   End
   Begin Finite Element Model The_Mesh
      Database Name = tapered_mesh_2d_1000.e
      Database Type = exodusII
      Begin parameters for block block_1
         Material SiC
      End parameters for block block_1
   End

   Begin Procedure My_Procedure
      Begin Solution Control Description
         Use System main
         Begin System main
            Begin Sequential mysolveblk
               Advance Global_Norm_Region
            End
         End
      End
      Begin Aria Region Global_Norm_Region
         Use Finite Element Model The_Mesh
         Nonlinear Solution Strategy = NOX
         Use NOX Nonlinear Solver NOX_Solver

         EQ current FOR voltage ON block_1 USING q1 WITH diff
         BC const dirichlet AT surface_1 voltage = {V_0}
         BC const dirichlet AT surface_3 voltage = {DeltaV}

         EQ energy FOR temperature ON block_1 USING q1 WITH diff src
         BC const dirichlet AT surface_1 Temperature = {T_0}
         SOURCE FOR energy ON block_1 = joule_heating

         EQ mesh FOR mesh_displacements ON block_1 USING q1 WITH diff
         BC const dirichlet AT surface_1 mesh_displacements_x = 0.0
         BC const dirichlet AT surface_1 mesh_displacements_y = 0.0
         BC const dirichlet AT surface_3 mesh_displacements_x = 0.0

         REFERENCE TEMPERATURE = {T_0}

         POSTPROCESS Electrical_Conductivity ON block_1

         Begin Results Output Label Aria MEMS Beam
            Database Name = out_coupled.e
            At Step 0, Increment = 1
            Title Aria MEMS Beam
            Nodal Variables = solution->Temperature as T
            Nodal Variables = solution->Voltage as V
            Nodal Variables = solution->Mesh_Displacements as D
            Nodal Variables = pp->Electrical_Conductivity as Sigma
         End
      End
   End
End
```

Figure 6.1. Aria Region and SiC Materials block, describing the setup of the Aria problem

```
     Begin Solution Control Description
        Use System main
        Begin System main
           Begin Sequential mysolveblk
              Begin Nonlinear nonlinear_loop
                 Advance V_Region
                 Advance T_Region
 Advance D_Region
                 Transfer VRegion_to_GNRegion_Transfer
                 Transfer TRegion_to_GNRegion_Transfer
 Transfer DRegion_to_GNRegion_Transfer
                 Advance Global_Norm_Region
                 Transfer GNRegion_to_VRegion_Transfer
                 Transfer GNRegion_to_TRegion_Transfer
 Transfer GNRegion_to_DRegion_Transfer
              End
           End
        End
        Begin Parameters For Nonlinear nonlinear_loop
           Converged When "CURRENT_STEP==80 || Global_Norm_Region.Region_Nonlinear_Residual < 1.0e-3"
        End
     End
     begin transfer TRegion_to_GNRegion_Transfer
        copy volume nodes from T_Region to Global_Norm_Region
        send field solution->TEMPERATURE state new to solution->TEMPERATURE state new
     end
     Begin Aria Region T_Region
        Use Finite Element Model The_Mesh
        Nonlinear Solution Strategy = NOX
        Use NOX Nonlinear Solver NOX_Solver_T
        EQ current FOR voltage ON block_1 USING q1 WITH xfer
IC const ON block_1 voltage = {V_0}
        EQ energy FOR temperature ON block_1 USING q1 WITH diff src
        IC const ON block_1 temperature = {T_0}
        BC const dirichlet AT surface_1 Temperature = {T_0}
        SOURCE FOR energy ON block_1 = joule_heating
        EQ mesh FOR mesh_displacements ON block_1 USING q1 WITH xfer
        IC const AT block_1 mesh_displacements_x = 0.0
        IC const AT block_1 mesh_displacements_y = 0.0
        REFERENCE TEMPERATURE = {T_0}
     End
```

Figure 6.2. Solution control block, and a sample transfer block and region block for the Jacobi solution method using Aria

```
Begin Aria Region Global_Norm_Region
   Use Finite Element Model The_Mesh

   Use Linear Solver Iterative_global
   Nonlinear Solution Strategy = Newton

   Maximum Nonlinear Iterations   = 1
   Nonlinear Residual Tolerance   = 1.0e+100
   Nonlinear Correction Tolerance = 1.0e+100
   Nonlinear Relaxation Factor    = 1.0

   EQ current FOR voltage ON block_1 USING q1 WITH diff
   BC const dirichlet AT surface_1 voltage = {V_0}
   BC const dirichlet AT surface_3 voltage = {DeltaV}

   EQ energy FOR temperature ON block_1 USING q1 WITH diff src
   BC const dirichlet AT surface_1 Temperature = {T_0}
   SOURCE FOR energy ON block_1 = joule_heating

   EQ mesh FOR mesh_displacements ON block_1 USING q1 WITH diff
   BC const dirichlet AT surface_1 mesh_displacements_x = 0.0
   BC const dirichlet AT surface_1 mesh_displacements_y = 0.0
   BC const dirichlet AT surface_3 mesh_displacements_x = 0.0

   REFERENCE TEMPERATURE = {T_0}

   POSTPROCESS Electrical_Conductivity ON block_1

   Begin Results Output Label Aria MEMS Beam
      Database Name = out_jacobi.e
      At Step 0, Increment = 1
      Title Aria MEMS Beam
      Nodal Variables = solution->Voltage as V
      Nodal Variables = solution->Temperature as T
      Nodal Variables = solution->Mesh_Displacements as D
      Nodal Variables = pp->Electrical_Conductivity as Sigma
   End
End
```

Figure 6.3. Special region added to the calculation, whose only purpose is to calculate the global norm. This block is used in the Gauss-Seidel and Jacobi weak coupling algorithms.

Aria's temperature region solves only for the temperature field. A source term for the temperature is specified with the model name, `joule_heating`; this source term implements the rhs of eqn.(4.19).

Figure (6.3) displays the Global_Norm_Region block referred to earlier. It's only purpose is to calculate a global norm of the entire residual for use as a stopping criteria at the Solution Control level. A very cheap iterative strategy is specified in the block. However, it's never needed, as the `Nonlinear Residual Tolerance` parameter is set so high that the initial residual always passes the convergence test. Therefore, this block never actually solves the fully coupled problem.

Figure 6.4 contains the results of this analysis. The convergence rates of the Jacobi and Gauss-Seidel methods were compared (where they converged) to the spectral radius, calculated within Matlab. Above a ΔV value of 2.021105 volts, the Gauss-Seidel and Jacobi algorithms failed to converge on the problem. The fully coupled solution algorithm managed to find converged solutions up to a ΔV value of 2.87 volts before its nonlinear solution algorithm failed.

Convergence rates, calculated by fitting a power law form to the global residual value for the last couple of iterations of the Gauss-Seidel method followed the trend of the spectral radius calculation. However, the actual convergence radius was always roughly a factor of 2 lower than the spectral radius. The Jacobi calculations had a higher radius of convergence than both the Gauss-Seidel value and the spectral radius value in almost all cases. The more robust convergence rates produced by the Gauss-Seidel method can be understood and justified by noting that those rates were calculated from the nonlinear problem, while the spectral radius values were calculated from the linear problem. Therefore, as the calculation converged towards the actual value, the Jacobians in the Gauss-Seidel calculation were getting progressively better, leading towards a greater convergence rate. As the spectral radius predicted by Matlab crosses from below 1 to above 1 (actually just a little above one), in all cases, both the Gauss-Seidel method and the Jacobi method no longer can be used to converge to the steady state solution. As can be seen, the convergence of the Gauss-Seidel algorithm can be understood and analyzed within Matlab by calculating the spectral radius of G_{BB}. This

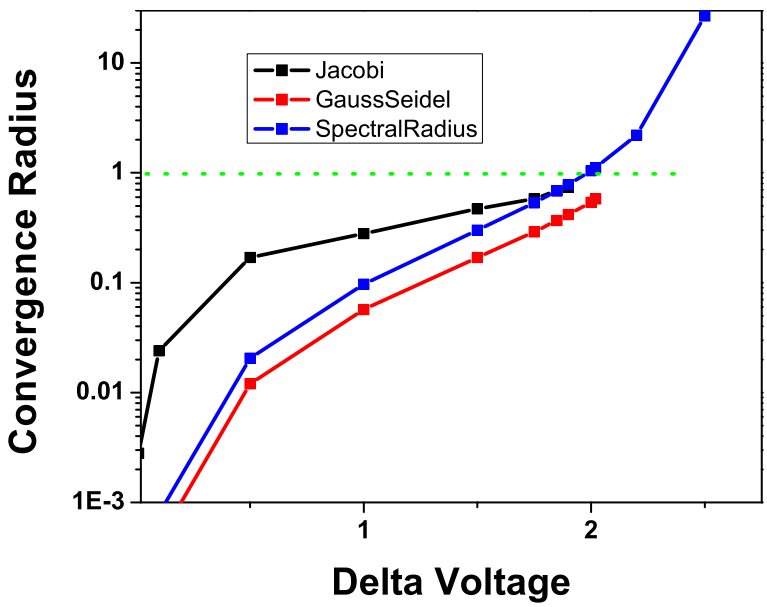


Figure 6.4. Convergence rates of the Jacobi and Gauss Seidel methods compared with Matlab calculations of the spectral radius of G.

provides numerical confirmation of the theory from the previous chapter on a real-world application in Sierra.

The next question involves how best to use these findings. One use is to assess whether we can successfully use weak coupling. In order to do this efficiently we need to have a cheaply calculated estimate of the spectral radius. It may be possible to make such cheap estimates from estimates of the norms of the four individual matrices needed to form G_{BB}.

$$\|G_{BB}\|_2 <= \left\|\mathbf{J}_{BB}{}^{-1}\right\|_2 \|\mathbf{J}_{BA}\|_2 \left\|\mathbf{J}_{AA}{}^{-1}\right\|_2 \|\mathbf{J}_{AB}\|_2 \tag{6.1}$$

However, because this involves estimating the condition number of both $\mathbf{J}_{BB}^{-1}$ and $\mathbf{J}_{AA}^{-1}$, an operation roughly equal in cost to actually carrying out the solution of one problem A and problem B step, there probably is no cheaper way to gauge an accurate estimate of the spectral radius than to actually carry out the Gauss-Seidel treatment algorithm itself and use the convergence rate of that to estimate the spectral radius of G_{BB}. In other words, by carrying out the weak coupling algorithm, we have found a cheap way to determine the spectral radius of G_{BB}, and not the other way round. It may very well be that no cheaper assessment exists to determine when weak coupling should converge than simply trying it out and calculating the convergence rate directly.

Additionally, because we have divided up a unity-ordered quantity in eqn.(6.1) into matrices, each of which have units and implying physical scales, scaling becomes a necessity for the accurate calculation of the estimate of $\|G_{BB}\|_2$. We have tried both row-sum scaling and column scaling in the evaluation of eqn.(6.1). However, the results were always never as satisfactory as the agreement we achieved in Figure 6.4.

```
Begin Nox Nonlinear Equation Solver my_nonlinear_solver_1
   # Non-Linear solver parameters:
   Nonlinear Solver Method                   = line search based
   Nonlinear Direction Method                = newton
   Nonlinear Linesearch Method               = full step
   Rescue Bad Newton Solve                   = true
   Nonlinear Absolute Residual Norm Tolerance = 1.0e-6
   Nonlinear Relative Residual Norm Tolerance = 1.0e-8
   Maximum Nonlinear Iterations              = 40
   # Linear solver parameters
   Maximum Iterations             = 300
   Residual Norm Tolerance        = 1.0e-4
   Linear Solver Output Frequency = 5
   Preconditioning Package = ifpack using myIfpack1
   Begin Teuchos Parameter Block myIfpack1
      Param-String "IfPack Preconditioner Choice" Value "ILU"
      Param-Int    "fact: level-of-fill"          Value 0
      Param-Int    "partitioner: overlap"         Value 0
   End
   Nox Output Level = high
   # Jacobian operator type -- IGNORED B/C THIS IS NEWTON-KRYLOV.
   Nonlinear Jacobian Operator = matrix free
   # Preconditioner matrix -- CURRENTLY IGNORED B/C IT USES THE ANALYTIC BLOCK DIAGONAL ONE FROM ARIA.
   Nox Nonlinear Preconditioning Method = Use Jacobian
End

Begin Procedure my_procedure
   Begin solution control description
      Use system main
      Begin system main
         Begin sequential mysolveblk
            Begin matrix free nonlinear nonlinear_loop
               Involve V_Region
               Involve T_Region
               Involve d_Region

               Transfer VRegion_to_TRegion_Transfer
               Transfer TRegion_to_VRegion_Transfer
               Transfer VRegion_to_dRegion_Transfer
               Transfer TRegion_to_dRegion_Transfer
               Transfer dRegion_to_VRegion_Transfer
               Transfer dRegion_to_TRegion_Transfer

               Use coupler my_nonlinear_solver_1
            End
         End
      End
      Begin parameters for nonlinear nonlinear_loop
         converged when "CURRENT_STEP==1"
      End
   End
   ...
End procedure
```

Figure 6.5. Sierra input blocks used for the implementation of NOX's JFNK method within Sierra. NOX's input block for the setup of its JFNK solution method is shown. The solution control block for setting up NOX's JFNK method on multiple regions is also shown.

6.2.2 JFNK Coupling Results

Figure (6.5) contains an example of a Solution Control block that implements the JFNK coupling algorithm in NOX. As before, the actual problem is divided into three Aria regions, V_Region, T_Region, and d_region, which solves the voltage problem, the energy equation, and the displacement fields, respectively. The Sierra Solution Control block adds the three regions to the JFNK block invoking the `involve` block on each region, within a `matrix free nonlinear` block. The key line `Use Coupler` tells the solution control block to invoke the NOX JFNK coupling strategy, given in the block named `my_nonlinear_solver_1` to solve the problem. Convergence of the `matrix free nonlinear` loop is handed off to NOX to determine. However, it still must be specified at the solution control level. The `parameters` block with the `converged when` key-line satisfies this requirement.

NOX's `my_nonlinear_solver_1` parameter block uses a few special commands. The key-line `Nonlinear Jacobian Operator` is set to `matrix free` instead of the more usual entries of `user supplied matrix` or `finite coloring` to indicate that no Jacobian should be formed at the NOX level. However, storage for a preconditioner matrix is still needed here if sufficient convergence of the linear problem is to be generated. The preconditioner is specified by the `Nox Nonlinear Preconditioning Method` key-line with the value, `Use Jacobian`. In this context, the line means that the block-diagonal Jacobian from the subproblems are used to populate the preconditioning matrix as described in Section 2.3.2.1.

There are two NOX parameter blocks. One of the blocks is used as a dummy solver to signal each Aria region to construct the appropriate solver objects needed to create the global problem.

Table (6.1) contains the results of the comparison between JFNK coupling and strong coupling via pure Newton method. While there is a bit of scatter in the results, it's readily seen that the number of nonlinear iterations taken by the JFNK approach is very similar to the fully coupled approach, verifying that the same basic nonlinear Newton's method is being used in the two methods. Also, the fact that the number of linear iterations is roughly equivalent in the two methods indicates that the preconditioning strategy of using each subproblem's block-diagonal Jacobian as input to a block-diagonal ilu(0) preconditioner with zero fill-in is adequate for this application. Therefore, we have achieved our goal of creating a coupling method in which the off-diagonal contributions to the Jacobian don't have to be explicitly calculated, but which behaves as though the off-diagonal contributions are fully integrated into Newton's method.

One drawback to the JFNK method is reflected in the timing results. The JFNK approach is significantly slower than the fully coupled method by roughly a factor of 5. Therefore, JFNK should only be used when necessary. While the nonlinear iterations and number of linear iterations are roughly equivalent for the two methods, the matrix-free GMRES algorithm takes many more residual evaluations during the linear solve that are not needed in the linear solve of the fully coupled method. Note, both methods calculate analytical Jacobians. The JFNK method calculates analytical Jacobians for the block diagonals, which are then used as inputs to the ilu preconditioner of the coupled problem. The coupled approach calculates a full Jacobian, analytical contribution using Aria's built-in capabilities.

Δ Volts	JFNK NonLin Its	Lin Its	Coupled NonLin Its	Lin Its
0.5	6	(66)	6	(136)
		(60)		(64)
		(44)		(52)
1.0	9	(117)	9	(67)
		(67)		(68)
		(66)		(66)
1.5	11	(67)	10	(133)
		(60)		(66)
		(68)		(65)
2.0	11	(67)	10	(133)
		(60)		(66)
		(68)		(65)
2.5	12	(145)	16	(146)
		(145)		(135)
		(67)		(84)
2.75	16	(70)	12	(147)
		(65)		(119)
		(65)		(67)
2.85	23	(85)	19	(137)
		(150)		(147)
		(150)		(126)
2.87	21	(145)	20	(136)
		(155)		(149)
		(95)		(146)

Table 6.1. Nonlinear Iterations Counts for JFNK coupling vs. strong coupling. The number of iterations in the linear solver taken by the last three nonlinear iterations are also shown.

Chapter 7

Nonlinear Elimination for Circuit/Device Coupling

This chapter demonstrates the nonlinear elimination multi-physics coupling technique as applied to a circuit/device simulation. The motivation stems from the qualification of circuits under hostile (radiation) environments. Circuits are normally modeled using a network simulator that solves a set of fully coupled lumped parameter models or what we term as the "low-fidelity" models. To adequately capture the effects of radiation damage, lumped parameter models are not adequate. High fidelity models of individual devices must be used.

The circuit simulator Xyce (11) is used to model large-scale networks of devices. The circuit is a set of coupled device models that can be as small a a few devices or as large as millions of devices with complicated interconnections. The devices are typically modeled with lumped parameter models that generate a system of high index dynamic algebraic equations (DAEs). Solving such systems can be difficult and Xyce employs such techniques as natural and artificial homotopy (12) and voltage limiting (13) to attain convergence to the steady-state conditions.

The "high fidelity" device model is implemented in the Charon finite element code. An individual device in Charon solves for the electric potential, density of holes and electrons, and defect species from radiation damage via a fully coupled set of partial differential equations (PDEs). The discretization uses a variational multi-scale stabilized Galerkin finite element formulation. A typical device will have on the order of 100's of millions of degrees of freedom. Large-scale supercomputers are needed to solve a single device, let alone the entire coupled system. Due to this size, it is impossible to model all devices in the circuit at this level. We therefore couple a few high fidelity Charon devices where the radiation damage occurs to the rest of the circuit that is described with the low fidelity models in Xyce.

The first section describes the general theory and procedure for the nonlinear elimination, the second section show results on simple test problems, and the third section summarizes the work.

7.1 Theory

The nonlinear elimination technique has been used in the past for circuit device coupling. Originally, it was called the "Two-level Newton" technique and was described specifically in terms of circuit/device coupling (14, 15). In the 90's, nonlinear elimination was put on a strong theoretical footing with convergence proofs and analysis (16, 17). It was eventually extended to systems other than circuits such as aerodynamics (18).

We begin by defining the notation. The equations for the network model equations are described by the subscript c, representing the circuit equations. The PDE device equations are described by

the subscript d for "device." The fully coupled set of discretized nonlinear equations is:

$$F(x) = \begin{bmatrix} F_c(x_c, x_d) \\ F_d(x_c, x_d) \end{bmatrix} = \mathbf{0}, \tag{7.1}$$

where F_c and F_d are the equations corresponding to the circuit and device equations respectively.

We would like to achieve the q-quadratic convergence rate of a Newton-based solve where we would normally solve the fully coupled system via Newton's method:

$$\begin{bmatrix} \mathbf{J}_{cc} & \mathbf{J}_{cd} \\ \mathbf{J}_{dc} & \mathbf{J}_{dd} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_c \\ \Delta\mathbf{x}_d \end{bmatrix} = \begin{bmatrix} \mathbf{F}_c \\ \mathbf{F}_d \end{bmatrix}, \tag{7.2}$$

where $J_{ij} = \frac{\partial F_i}{\partial F_j}$ is the Jacobian matrix. Consistent with the common theme of our coupling studies, solution of the overall coupled problem in this manner encounters some difficulties. First, if the codes are written completely independent of one another, getting off-diagonal dependencies (the $\mathbf{J}_{cd}$ and the $\mathbf{J}_{cd}$ blocks) may be prohibitively expensive if not impossible to compute. For example, a finite difference Jacobian may be the only way to get Jacobian dependencies, and as the number of equations increases, the computation becomes intractable.

A second issue that is extremely important to solving the multi-physics associated with device-circuits is the is the need for and delicate nature of globalizations for Newtons method. Globalizations typically include line search and trust region algorithms (19), homotopy methods (12, 20) and the circuit specific voltage limiting (13). The difficulty is that globalization routines do not perform well for all physics. For example, basic polynomial line search methods are extremely ineffective on certain low fidelity simulations while they work extremely well on the PDE-based high fidelity models. The explanation is that line searches scale the Newton direction, $\Delta\mathbf{x}$, to enforce a monotonic reduction of the residual during the Newton iteration sequence. Circuit models typically have discontinuous functions and/or derivatives causing the scale factor to be reduced to extremely small values to enforce monotonicity thereby causing the method to stall out. For a device to switch states, it must permit increases to the magnitude of the residual at some point in the iteration sequence. Most globalization routines assume continuous functions, which does not hold for the on-off character of certain device states. Thus, coupling the low and high fidelity models together can generate systems that require different globalizations that are mutually incompatible. Nonlinear elimination can be used a technique to preserve application specific globalizations while achieving a fully coupled solve. Optimal solution algorithms including appropriate globalizations are utilized for each physics.

The objective of nonlinear elimination is to eliminate one or more sets of physics from the solve. In this case we chose the device equations to be eliminated. This choice was made due to constraints on the coupling. At the time this was undertaken, the quantities of interest could only be computed via a one-way process. The high fidelity code could only accept voltages and return the currents. The reverse process was not available and thereby dictated our choice of problem to be eliminated.

We begin by assuming that we can solve the stand alone system of device equations

$$\mathbf{F}_d(\mathbf{x}_c, \mathbf{x}_d) = 0, \tag{7.3}$$

for the device solution, $\mathbf{x}_d$. This is done by treating the circuit solution variables, $\mathbf{x}_c$, as "fixed" parameters in the device simulation model. If such a solution exists, then we can define an implicit function

$$\mathbf{x}_d = \mathbf{h}(\mathbf{x}_c). \tag{7.4}$$

If the solution to the implicit function exists, then $\mathbf{F}_d = \mathbf{0}$ and we can rewrite equation 7.2 leveraging the implicit function theorem as the following

$$\mathbf{F}(x) = \mathbf{F}_c(\mathbf{h}(\mathbf{x}_c), \mathbf{x}_c) = \mathbf{F}(x_c). \tag{7.5}$$

The resulting Newton system is just the circuit Newton solve

$$\mathbf{J}_{cc}\Delta_{\mathbf{x}_c} = \mathbf{F}(\mathbf{x}_c). \tag{7.6}$$

From this, we construct the following iterative Newton method for the nonlinear elimination technique:

While not converged: $k = 0, 1, \ldots N$
compute $\mathbf{J}_c, \mathbf{F}_c$
solve $\mathbf{J}_c^k \Delta\mathbf{x}_c^k = -\mathbf{F}_c^k$
update $\mathbf{x}_c^{k+1} = \mathbf{x}_c^k + \Delta\mathbf{x}_c^k$

The important step is the "compute" step. During the evaluation of the Jacobian and residual, a complete nonlinear solve of the high fidelity equation set is required to evaluate the terms for the assembly of the low fidelity Newton system. This is a two-stage solve. We have an "outer Newton solve" of the low fidelity model, during which a complete nonlinear solve or "inner solve" of the high fidelity model is required. This can be very inefficient compared to a fully coupled solve, but is really dependent on the cost of evaluating the off-diagonal blocks versus the cost of full nonlinear solves of the inner iteration.

7.2 Results

Here we show results from two test circuits. We discuss critical details required for the implementation of the nonlinear elimination technique. Issues we address include time integration error control, convergence tolerance, convergence rates, and efficiency.

The first test circuit is a verification problem shown in Figure 7.1. The green inverted triangle is the diode that is simulated with the high fidelity model. This problem is a diode rectifier taken from (21). It is a transient problem where an oscillating current source is damped using a diode. To begin the simulation, a steady state solve (called the DC Operating point solve) is performed to determine the initial condition of the circuit. During the inner solve, a natural parameter homotopy run was performed using the diode contact voltage as the parameter to achieve convergence of the inner problem. Four continuation steps took the voltage from 0.1 to 0.5 volts. The outer problem did not need a globalization and used Newton's method directly.

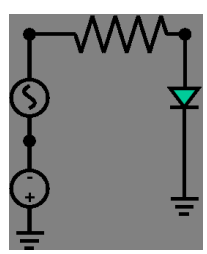


Figure 7.1. Diode rectifier circuit test model.

Once the DC Operating point was attained, the transient simulation was started. An important detail involves how to couple the time integrators between the physics. If the time scales vary widely, one can apply operator splitting techniques. In this case, each of the physics was operating on nearly the same time scales, and we chose to solve the system fully coupled. This lead to much work in getting the time integrators to synchronize with each other. The time integrator in the outer loop was directing the time step sizes for both physics problems, however the inner solve used it's own

time integrator to determine if the step sizes adequately controlled the requested error for the inner problem. The voltage values on each end of the diode is shown as a function of time in figure 7.2. Note that at 0.013 seconds the step size is reduced. This resulted from a failure of the inner iteration

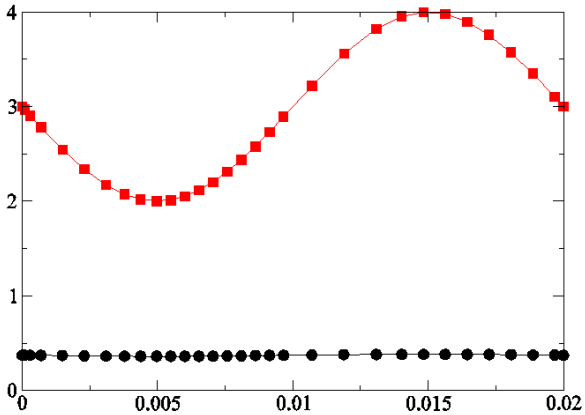


Figure 7.2. Diode rectifier results: Plot of Voltage (V) vs. time (sec).

to achieve its relative tolerance. The inner solver converged to a solution, but the time step was rejected since it failed to meet the relative error tolerance. It is critical in transient mode that the time integrators communicate errors and convergence issues correctly.

For typical transient error tolerances, we observed that the outer solve averaged two Newton steps while the inner solve averaged three Newton steps. This shows that the nonlinear elimination techniques is not as inefficient as one would expect. Two outer iterations lead to only two nonlinear solves of the inner problem.

The second test problem is a steady state npn bipolar junction transistor (BJT). Figure 7.3 depicts the circuit. The BJT device with contacts at points E, C, and B is the high fidelity device simulated

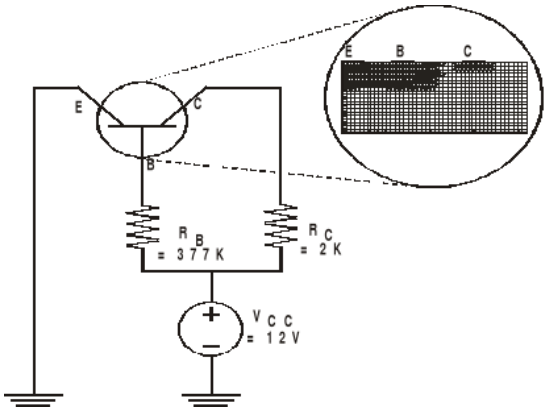


Figure 7.3. npn BJT circuit test model.

by Charon (mesh shown in inset), while the rest of the circuit is simulated with the Xyce low fidelity models. The problem converged in 10 Newton steps in the outer iteration. To achieve convergence, a

Inner Tolerance		Outer Tolerance		Number of	Time
Abs	Rel	Abs	Rel	Newton Steps	(sec)
1.0e-12	1.0e-4	1.0e-8	1.0e-6	13	525
1.0e-8	1.0e-6	1.0e-8	1.0e-6	13	400
1.0e-4	1.0e-6	1.0e-8	1.0e-6	F	F

Table 7.1. Convergence performance as a function of residual tolerances. F indicates a convergence failure of the outer iteration.

voltage limiting globalization is required on the outer iteration circuit. The inner iteration required an initial guess generated by a nonlinear poisson solve.

A plot of the electric potential for the high fidelity BJT device is shown in Figure 7.4. Note the sharp

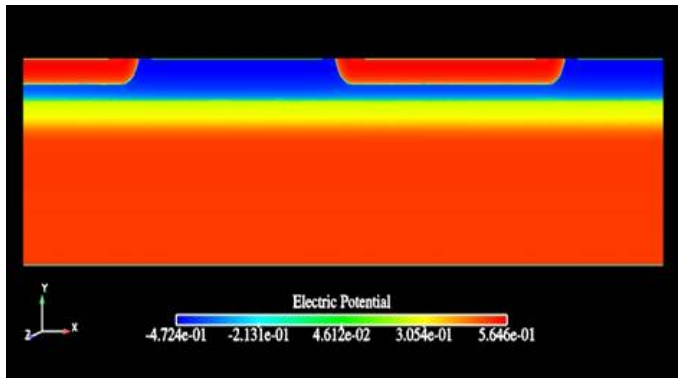


Figure 7.4. npn BJT: Plot of the electric potential.

gradients near the contacts on the top of the domain. An unstructured mesh with h-adaptivity is typically required to resolve the features.

The first issue of importance is that the stopping criteria of the inner and outer problems can not be chosen independently. The stopping criteria for our inner and outer solves was based on a weighted root mean square norm defined as

$$1.0 \leq \sqrt{\sum_i = 1^N \left(\frac{x_i^k - x_i^{k-1}}{\text{RelTol}|x_i^{k-1}| + \text{AbsTol}} \right)^2}, \tag{7.7}$$

where N is the number of degrees of freedom in the solution vector and k is the Newton iterate. Table 7.1 shows the performance of the problem with various the inner and outer stopping tolerances. If the norm of the residuals for the inner solve is less than the norm of the residuals for the outer solve, then the Jacobian and residual contributions for the outer solve can be inconsistent. This typically manifests as oscillations during the outer iteration sequence. This problem is due to the Jacobian contributions from the inner solve. Given a circuit solution vector $\mathbf{x}_c$, the computed Jacobian from the inner solve will be accurate to only so many digits based on both the inner solve convergence tolerances *and the initial guess*. Each evaluation was not guaranteed to use the same initial guess, so a loose inner tolerance will result in an inaccurate Jacobian for the outer problem.

A second issue to address is the problem of incompatible globalizations. In this test, the circuit simulator required a voltage limiting globalization whereas the device simulator required a specialized nonlinear Poisson initial guess with a line search to converge. The linear search algorithm is

incompatible with the voltage limiting algorithm if one were to attempt to solve the fully coupled system. In this case, it is suspect as to whether convergence could be attained at all on the fully coupled system (we did not attempt to demonstrate the fully coupled solve). As described previously, nonlinear elimination can be used as a means to address conflicting globalizations. By separating the solve, each physics can use the "best" globalization technique developed for that problem. We cannot overemphasize this critical insight.

As mentioned earlier, nonlinear elimination should produce super-linear convergence rates. In figure 7.5, we verify that the method attains super-linear rates.

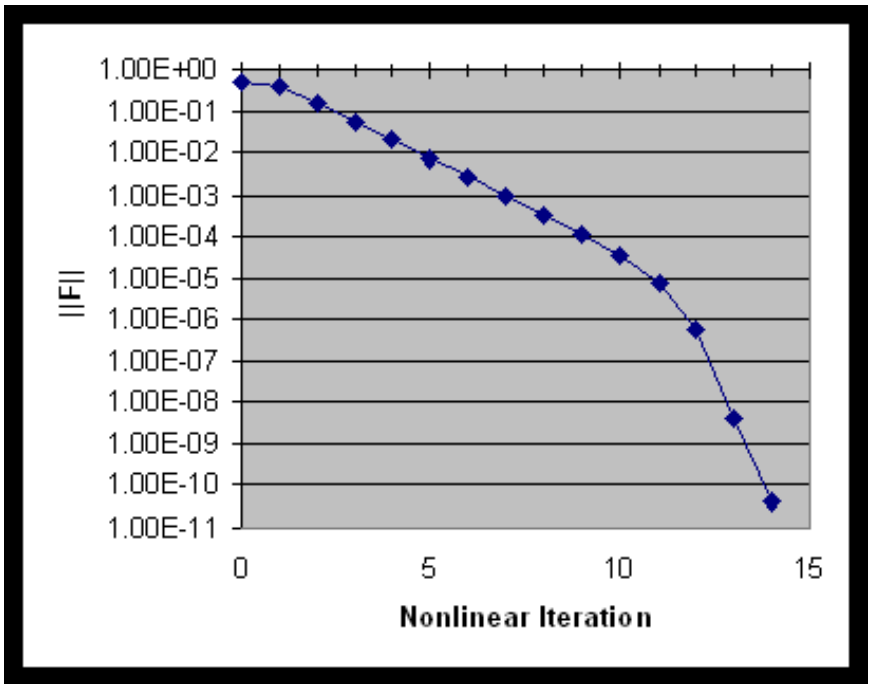


Figure 7.5. npn BJT: Norm of the error in the residual as a function of outer iteration.

7.3 Summary

We have demonstrated the use of nonlinear elimination as a valuable addition to our suite of coupling algorithms. Our unique contribution is that we are using the method to circumvent conflicting globalization routines of the different physics. The advantages of nonlinear elimination are that it produces a fully coupled solve that is robust since each physics can use the best globalization technique for each individual physics. It is well suited to multi-physics simulation codes and is less invasive than a fully coupled code. The disadvantages center around efficiency. Each Newton step of the outer iteration requires a full nonlinear solve of the inner iteration which is much more expensive than a fully coupled solve.

Future research directions on nonlinear elimination should target improving the efficiency of the solve. Some promising techniques were proposed during the course of this study, but there was insufficient time to implement and test the ideas.

Chapter 8

Example of Coupled Heat Conduction and Enclosure Radiation

8.1 Overview

This chapter presents an in-depth study of coupling issues encountered in modeling various modes of heat transfer, which is generally a problem of great importance in many engineering applications both inside and outside Sandia . The model for heat transfer can be simple, taking into account only a single mode of heat transfer, such as conduction, convection, or radiation, or it can include combinations of these in a coupled system. In this section, we consider coupling of heat conduction within a solid and thermal radiation between the surfaces of an enclosure, which is known as enclosure radiation (ER). Coupled modeling of conduction and ER is important to many applications at Sandia and within industry, where high temperatures induce significant heat transfer via ER.

Coupled conduction and ER is also a challenging coupling problem by which to test algorithms for coupling nonlinear physical models. The conduction problem is solved within a solid domain, while the ER problem is solved on a subset of the surface of the solid domain. The problems are coupled on the enclosure surface in two ways. First, the surface temperature from the conduction problem provides forcing data for the ER problem according to the Stefan-Boltzmann law

$$E \equiv \epsilon \, \sigma \, u^4, \tag{8.1}$$

where u is the temperature, ϵ is the emissivity, and σ is the Stefan-Boltzmann constant. Second, the ER problem provides a heat flux into the solid domain, which can be defined as the net difference between emitted and incident thermal radiation

$$q_n \equiv J - G, \tag{8.2}$$

where q_n is the outward normal heat flux, J is the emitted thermal radiation, called the *radiosity*, and G is the incident thermal radiation.

At Sandia, the SIERRA Mechanics codes Calore, and now Aria, have handled this coupling through weak coupling as described in Section 2.2. This approach was also the standard algorithm for the previous thermal code Coyote. The basic approach is to compute the radiation heat flux by passing the current temperature values on the enclosure surface to the ER solver Chaparral. These temperature values are then used by Chaparral through the expression (8.1) to solve the linear radiosity integral equations for J, compute G by postprocessing, and then to pass the net heat flux in eq.(8.2) back to Calore/Aria.

Consistent with previous discussions in this report of weak coupling performance this approach to the nonlinear conduction-radiation coupling exhibits only a linear convergence rate. For some problems,

the nonlinear solver can converge very slowly or even fail to converge. Because of these issues, we investigated stronger coupling algorithms in hopes of achieving the performance of full coupling via a Newton-based coupling. We did this both by using the NOX Jacobian-Free Newton-Krylov (JFNK) capability in Calore similar to our Aria multi-physics study described in Section 4.2 and Chapter 6 and by implementing a fully coupled Newton iteration directly in Calore.

The JFNK coupling via NOX is appealing in this case, since Calore eliminates the ER variables during its residual fill, by calling Chaparral to compute the radiative heat flux. In the fully coupled model, the Jacobian matrix for the temperature in the solid domain is augmented to include the surface radiosity matrix and the off diagonal coupling terms.

8.2 Problem Formulation

In this section we present the mathematical model for coupled conduction and ER, beginning with the decoupled models and concluding with the equations of the coupled model.

8.2.1 Thermal Conduction Problem

We consider a system containing a solid domain Ω in which the temperature u is the field of interest. Within Ω we consider heat transfer arising from conduction only. The surface $\partial\Omega$ is divided into several pieces upon which different boundary conditions are applied. On Γ_D we specify the temperature to be some environmental temperature

$$u = u_D \tag{8.3}$$

and on Γ_R we pose a radiative flux boundary condition

$$-(k\,\nabla u)\cdot n = \epsilon\,\sigma\,u^4 - \alpha\,G. \tag{8.4}$$

Here $E \equiv \epsilon\,\sigma\,u^4$ is the local emitted thermal radiation according to the Stefan-Boltzmann law of eq.(8.1) and G is the local incident radiation. The coefficients are emissivity ϵ and absorptivity α. The radiative heat flux is a nonlocal boundary condition, which is defined as a balance of heat emitted locally (E) and absorbed from nonlocal sources ($\alpha\,G$).

The above model can be described mathematically as a conduction–radiation heat transfer problem: find the temperature $u(x,t)$ satisfying:

$$\rho\,c_p\,\partial_t u - \nabla\cdot(k\,\nabla u) = Q, \quad x\in\Omega, t>0, \tag{8.5a}$$

$$-(k\,\nabla u)\cdot n = \epsilon\,\sigma\,u^4 - \alpha\,G, \quad s\in\Gamma_R, t>0, \tag{8.5b}$$

$$u = u_D, \quad s\in\Gamma_D, t>0, \tag{8.5c}$$

$$u = u_0, \quad x\in\Omega, t=0. \tag{8.5d}$$

Here Q is a volumetric heat source term, and k is the thermal conductivity. Thus the solution u is determined by the data $\{Q, G, u_D, u_0\}$.

8.2.2 Enclosure Radiation Problem

We now describe the radiative transfer problem in greater detail, based on several references (22, 23, 24). We assume for simplicity that Γ_R forms a complete enclosure with no holes, and consider the problem of thermal radiative heat transfer between points on Γ_R. At each point thermal radiation

is emitted according to $E = \epsilon\,\sigma\,u^4$, absorbed according to $\alpha\,G$, and reflected according to $\rho\,G$, where ρ is the local reflectivity. The coefficients are related by

$$\epsilon = \alpha = 1 - \rho. \tag{8.6}$$

A fundamental quantity for enclosure radiation is the *radiosity* or the total radiative flux leaving Γ_R, which is defined as

$$J \equiv E + \rho\,G. \tag{8.7}$$

We note that, using the definition of J in (8.7) and the energy balance in (8.4), the net flux can now be described as in eq.(8.2)

$$-(k\,\nabla u)\cdot n = J - G.$$

In order to solve for the radiosity J, an expression is needed that relates G and J. In fact the local incident radiation G is nothing more than the total radiation coming from all other points. This can be represented using an integral operator $\mathcal{F}$, which is defined by

$$(\mathcal{F}\,v)(s) \equiv \int_{\Gamma_R} K(r,s)\,v(r)\,dr. \tag{8.8}$$

Here the kernel $K(r,s)$ is the differential geometric view factor between any pair of points $r, s \in \Gamma_R$ defined by

$$K(r,s) \equiv \frac{\cos(\theta_r)\,\cos(\theta_s)}{c_d\,|r-s|^{d-1}} = \frac{[n_r\cdot(r-s)]\,[n_s\cdot(s-r)]}{c_d\,|r-s|^{d+1}}, \tag{8.9}$$

where d is the spatial dimension, n_r is the unit normal at r, θ_r is the angle between n_r and the vector $(r-s)$, and θ_s is the angle between n_s and the vector $(s-r)$. The constant c_d is equal to 2 for $d = 2$ and π for $d = 3$.

The equation that relates G and J is then

$$G = \mathcal{F}\,J. \tag{8.10}$$

We can now eliminate G and solve for J using the equation

$$J - (1-\epsilon)\,\mathcal{F}\,J = E. \tag{8.11}$$

The solution J is thus determined solely by the data E.

The solvability of the enclosure radiation problem in (8.11) is discussed in (22, 24). The main assumption on the emissivity ϵ is that it be bounded away from zero or

$$0 < \epsilon_0 \le \epsilon(s) \le 1, \quad s \in \Gamma_R. \tag{8.12}$$

It turns out that this condition is sufficient for the problem in (8.11) to have a unique solution and to vary continuously with the data E (24).

8.2.3 Variational Form of the Coupled Problem

When the temperature u is solved using eq.(8.5), the enclosure radiation problem in eq.(8.11) is coupled through the source data E. Likewise, when the incident radiative flux G is computed from the solution J of eq.(8.11), then the conduction problem in eq.(8.5) is coupled through the data G. Because of these couplings, the combined conduction-radiation problem is a two-way coupled multi-physics problem. In this section we formulate a coupled variational form of the coupled system for $\{u, J\}$ in the steady state case (the extension to the transient case is straightforward).

Multiplying by a test function v, and using eq.(8.10) to substitute for G in terms of J, the temperature u solves the variational problem: find u satisfying (8.3):

$$\int_\Omega k\,\nabla u\cdot\nabla v\,dx + \int_{\Gamma_R}\{\epsilon\,\sigma\,u^4\,v - \epsilon\,(\mathcal{F}\,J)\,v\}\,ds = \int_\Omega Q\,v\,dx, \quad \forall v, \tag{8.13}$$

for all test functions v that vanish on Γ_D. The equation for J in eq.(8.11) can be written in variational form as: find J:

$$\int_{\Gamma_R}\{J - (1-\epsilon)\,\mathcal{F}\,J\}\,\tau\,ds = \int_{\Gamma_R} E\,\tau, \tag{8.14}$$

for all test functions τ. Combining these systems we have the variational problem: find (u, J) satisfying eq.(8.3):

$$\int_\Omega k\,\nabla u\cdot\nabla v\,dx + \int_{\Gamma_R}\epsilon\,\sigma\,u^4\,v\,ds - \int_{\Gamma_R}\epsilon\,(\mathcal{F}\,J)\,v\,ds = \int_\Omega Q\,v\,dx, \quad \forall v, \tag{8.15a}$$

$$\int_{\Gamma_R} J\,\tau\,ds - \int_{\Gamma_R}(1-\epsilon)\,(\mathcal{F}\,J)\,\tau\,ds - \int_{\Gamma_R}\epsilon\,\sigma\,u^4\,\tau\,ds = 0, \quad \forall\tau. \tag{8.15b}$$

8.3 Discrete Approximation of the Coupled Problem

We now turn to the discretization by finite elements. Let $\{V_h\}$ be a family of finite dimensional approximation spaces approximating the temperature field, and let $\{X_h\}$ be another family approximating the radiosity. We assume that the domain Ω has been partitioned into a mesh $\mathcal{P}_h$ such that the set of faces of elements on Γ_R forms a surface mesh $\mathcal{G}_h$. For the space V_h, we choose the finite element space to be based on continuous, piecewise polynomial shape functions of degree p on the mesh $\mathcal{P}_h$. For X_h we will only consider the space of discontinuous piecewise constant functions on the surface mesh $\mathcal{G}_h$.

The linear integral operator $\mathcal{F}$ is also discretized using the space X_h. We define a discrete linear operator $\mathcal{F}_h : X_h \to X_h$ by Galerkin project:

$$\int_{\Gamma_R}(\mathcal{F}_h\,\tau_h)\,\chi_h\,ds = \int_{\Gamma_R}(\mathcal{F}\,\tau_h)\,\chi_h\,ds = \int_{\Gamma_R}\int_{\Gamma_R} K(r,s)\,\tau_h(r)\,dr\,\chi_h(s)\,ds, \quad \tau_h, \chi_h \in X_h. \tag{8.16}$$

We can define a basis for the space X_h given by $\{\chi_i\}$, where χ_i is one on the face A_i and zero elsewhere. Then the discrete operator $\mathcal{F}_h$ is represented in this basis by a matrix F_{ij}, such that if $J_h = \sum_j J_j\,\chi_j$, then

$$\mathcal{F}_h\,J_h = \sum_i(\sum_j F_{ij}\,J_j)\,\chi_i. \tag{8.17}$$

The matrix F_{ij} can be calculated from eq.(8.16) to be defined by

$$F_{ij} \equiv |A_i|^{-1}\int_{\Gamma_R}\int_{\Gamma_R} F(r,s)\,\chi_j(r)\,\chi_i(s)\,drds = |A_i|^{-1}\int_{A_i}\int_{A_j} F(r,s)\,drds. \tag{8.18}$$

The matrix F_{ij} is not symmetric, but does satisfy the reciprocity formula

$$|A_i|\,F_{ij} = |A_j|\,F_{ji}. \tag{8.19}$$

In Calore/Aria, the calculation of the discrete view-factor matrix F_{ij} is handled by the library Chaparral. This calculation is complex and expensive, especially when blocking surfaces are involved.

The nonlinear system of equations is determined by finding $(u_h, J_h) \in V_h \times X_h$ such that

$$\int_\Omega k\,\nabla u_h \cdot \nabla v_h\,dx + \int_{\Gamma_R} \epsilon\,\sigma\,u_h^4\,v_h\,ds - \int_{\Gamma_R} \epsilon\,(\mathcal{F}_h\,J_h)\,v_h\,ds = \int_\Omega Q\,v_h\,dx, \quad v_h \in V_h, \tag{8.20a}$$

$$\int_{\Gamma_R} J_h\,\tau_h\,ds - \int_{\Gamma_R} (1-\epsilon)\,(\mathcal{F}_h\,J_h)\,\tau_h\,ds - \int_{\Gamma_R} \epsilon\,\sigma\,u_h^4\,\tau_h\,ds = 0, \quad \tau_h \in X_h. \tag{8.20b}$$

We will abbreviate the system in eq.(8.20) using the abstract notation

$$\begin{aligned} R^u(u_h, J_h) &\equiv A\,u_h + C\,J_h + F(u_h) - Q &= 0 \\ R^J(u_h, J_h) &\equiv B\,J_h - G(u_h) &= 0 \end{aligned} \tag{8.21}$$

where

$$\begin{aligned} A_{ij} &\equiv \int_\Omega k\,\nabla v_j \cdot \nabla v_i\,dx, \\ F_i(u_h) &\equiv \int_\Gamma \epsilon\,\sigma\,u_h^4\,v_i\,ds, \\ C_{ik} &\equiv -\int_\Gamma \epsilon\,(\mathcal{F}_h \chi_k)\,v_i\,ds, \\ Q_i &\equiv \int_\Omega Q\,v_i\,dx, \\ B_{lk} &\equiv \int_\Gamma \{\chi_k\,\tau_l - (1-\epsilon)(\mathcal{F}_h\,\chi_k)\,\tau_l\}\,ds, \\ G_l(u_h) &\equiv \int_\Gamma \epsilon\,\sigma\,u_h^4\,\tau_l\,ds. \end{aligned} \tag{8.22}$$

8.4 Coupling Strategies

In this section we discuss three different solution strategies to the discrete coupled nonlinear system in eq.(8.20). These include weak coupling, full coupling, and JFNK coupling. We will use the abstract form in eq.(8.21) in order to simplify the presentation. For convenience, we drop the subscripts h on the discrete approximations u_h and J_h.

8.4.1 Weak Coupling

This algorithm is the same as that in Section 2.2. For the problem at hand we proceed as follows. Given the current temperature field u^n, the radiosity problem is solved with the temperature lagged.

$$B\,J^{n+1} = G(u^n) \tag{8.23}$$

This can be done using a radiosity solver that has been passed in the temperature field u^n or else the source data $G(u^n)$. Next, the new radiosity field J^{n+1} is combined with the old temperature u^n and substituted into the temperature equation.

$$A\,u^{n+1} = -C\,J^{n+1} - F(u^n) - Q \tag{8.24}$$

An alternate form is to linearize the source term $F(u)$ around the old temperature u^n. If we perform this partial linearization, we obtain a specialization of weak coupling which we will refer to in this chapter as "Weak Coupling".

$$\begin{aligned} B\,J^{n+1} &= G(u^n) \\ A\,u^{n+1} + F'(u^n)\,u^{n+1} &= -C\,J^{n+1} - F(u^n) + F'(u^n)\,u^n - Q \end{aligned} \tag{8.25}$$

This equation is iterated on until the convergence with respect to u^{n+1} is achieved.

8.4.2 Full Coupling

The full coupling strategy we consider in this study is nearly a pure Newton method as described in Section 2.1 except that the linear solves are done approximately using an iterative linear solver instead of a direct solver. Using the incremental formulation corresponding to eq.(8.21), this algorithm takes the form of

$$\begin{pmatrix} A + F'(u^n) & C \\ -G'(u^n) & B \end{pmatrix} \begin{pmatrix} \delta u^{n+1} \\ \delta J^{n+1} \end{pmatrix} = - \begin{pmatrix} R^u(u^n, J^n) \\ R^J(u^n, J^n) \end{pmatrix} \tag{8.26}$$

We can contrast this strategy with the Weak Coupling strategy by rewriting eq.(8.25) in incremental form

$$\begin{pmatrix} A + F'(u^n) & C \\ 0 & B \end{pmatrix} \begin{pmatrix} \delta u^{n+1} \\ \delta J^{n+1} \end{pmatrix} = - \begin{pmatrix} R^u(u^n, J^n) \\ R^J(u^n, J^n) \end{pmatrix} \tag{8.27}$$

Thus the Weak Coupling strategy differs from the Full Coupling strategy by dropping the off-diagonal operator $-G'(u^n)$ from the full Jacobian operator.

We know that the Full Coupling algorithm, since it is based on Newton-Krylov iteration, has the property of second order or quadratic convergence. The Weak Coupling algorithm has only formally first order convergence. Moreover, the rate of convergence of Weak Coupling will be strongly problem dependent, based on the physical parameters, geometry, and input data. Moreover, it may not even be convergent for some parameter regimes. These pathologies have been observed even for the prototyping studies in Chapter 3

We may analyze the convergence behavior of the Weak Coupled algorithm by applying the analysis presented in Chapter 5 to the algorithm in eq.(8.25). We performed the analysis in Section 5.2 by first linearizing the Jacobian and then transforming Newton's method for a coupled A and B system into a fixed point iteration system. We then found the iteration matrix for the fixed point scheme to determine the convergence behavior of the linearized system. The same can be done here, starting with eq.(8.26). We then obtain the result that a necessary condition for convergence of the Weak Coupling system is that the spectral radius of the following matrix, $\mathbf{G}_{BB}$,

$$\mathbf{G}_{BB} = \mathbf{J}_{BB}^{-1} \mathbf{J}_{BA} \mathbf{J}_{AA}^{-1} \mathbf{J}_{AB}, \tag{8.28}$$

is less than one. In the coupled radiation case, the elements of the Jacobian are given by the following relations.

$$\begin{aligned} \mathbf{J}_{AA} &= A + F'(u^n) \\ \mathbf{J}_{AB} &= C \\ \mathbf{J}_{BA} &= -G'(u^n) \\ \mathbf{J}_{BB} &= B \end{aligned}$$

The convergence behavior of the nonlinear system may then be related to the convergence behavior of the linear system in the following manner. At the steady state solution, the convergence of the linear system must be possible for convergence of the nonlinear system to be possible. This sets up a sufficient condition on the convergence behavior of the nonlinear system that the spectral radius of $\mathbf{G}_{BB}$ be less than one.

Using a Taylor series expansion around the exact solution, we can in fact show that the linear convergence rate of the Weak Coupling approach is bounded by the matrix operator norm of

$$\begin{aligned} & I - \begin{pmatrix} A + F'(u^n) & C \\ 0 & B \end{pmatrix}^{-1} \begin{pmatrix} A + F'(u^n) & C \\ -G'(u^n) & B \end{pmatrix} \\ & = \begin{pmatrix} (A + F'(u^n))^{-1} C \, B^{-1} \, G'(u^n) & 0 \\ -B^{-1} \, G'(u^n) & 0 \end{pmatrix}. \end{aligned} \tag{8.29}$$

We will use this result below in a simple conduction-radiation prototype example to show that this rate is correct for the Weak Coupling case. This rate is also important for determining when the Weak Coupling may fail, since the sufficient conditions for convergence of fixed-point iterations are that the rate is less than one.

8.4.3 Jacobian-Free Newton-Krylov

Another alternative to the Weak Coupling strategy is JFNK coupling available in the NOX interface in Sierra and described in detail in Section 2.3.1 and elsewhere in this report.

We could apply the JFNK algorithm to the pair of coupled residuals defined in eq.(8.21). However, since the Full Coupling case can be easily obtained once the coupled residuals have been implemented, we decided instead to apply this approach to the reduced residual that is computed by the Weak Coupling implementation in Calore.

As we described in Section 8.1, Calore/Aria passes the current temperature values on the enclosure surface into Chaparral, which then returns the net heat flux back to Calore. This is equivalent to eliminating the radiosity variable J from the coupled system in eq.(8.21), resulting in the reduced residual

$$R(u) \equiv A\,u + C\,B^{-1}\,G(u) + F(u) - Q = 0. \tag{8.30}$$

For JFNK coupling, this residual can be formed and differenced without any explicit knowledge of how the coupling works. Hence, we employ nonlinear elimination analogous to the approach described in Section 7.1.

The remaining work associated with applying JFNK coupling is to define the preconditioner. Developing an efficient preconditioner requires knowledge about how the discrete temperature solution is coupled on the enclosure. In a convex enclosure, every temperature degree of freedom (dof) is coupled to every other on the enclosure. However, when blocking surfaces exist in the enclosure, some couplings may not exist. One possibility is to simply compute a preconditioner based on the conduction matrix from Calore, which does not contain the full coupling information.

Another possibility is to assume all dofs on the enclosure are coupled and to augment the matrix sparsity graph to reflect this. Then the full Jacobian matrix can be filled using finite coloring algorithms in NOX. The drawback to this approach is that the resulting preconditioner can be very expensive to compute due to the density of the additional coupling terms. We will demonstrate these approaches in the numerical examples, but would like to emphasize that choosing optimal preconditioners was beyond the scope of the work.

8.5 Numerical Examples

We now present results to illustrate effects of nonlinear coupling on time integration and spatial mesh adaptivity. In addition, we describe a useful verification problem with analytic solution, for which we verify that we can estimate the rate of convergence of weak coupling. Finally, we illustrate the importance of the full Jacobian in the case of adjoint-based error estimation.

8.5.1 3D Coupled Conduction/ER Problem

8.5.1.1 Problem description

Our first example is a three-dimensional brick containing two enclosures in the interior. The dimensions are $1 \times 1 \times 2$ with two hexagonal enclosures of length 1.0 placed symmetrically along the z axis. The temperature solution at time $t = 0.16$ is shown in Figure 8.1, with part of the domain cut away so that the enclosures are visible.

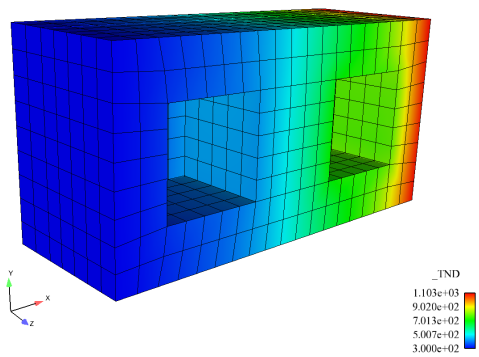


Figure 8.1. Cutaway of temperature solution at $t = 0.16$ for 3D transient ER coupling problem.

The material properties are homogeneous and isotropic, and are defined in Table 8.5.1.1. The only

Property	Value
k	$2\ W/m{-}K$
ρ	$1\ kg/m^3$
c_p	$1\ J/kg{-}K$
ϵ	0.8
u_0	$300\ K$
u_D	$300\ K$
h_c	$10\ W/m^2$

property not previously described is the convective heat transfer coefficient h_c. This parameter is used in a convective heat transfer boundary condition of the form

$$q_n = h_c\,(u - u^{\text{ref}}).$$

The reference temperature is defined by

$$u^{\text{ref}}(t) \equiv \begin{cases} 1300\text{K}, & 0 < t < 0.2, \\ 300\text{K}, & 0.2 < t \end{cases}$$

The convective boundary condition is defined on the surface $x = 1$, and the Dirichlet boundary condition is applied on the surface $x = -1$. The interior surfaces have the ER model applied, and all other exterior surfaces are adiabatic.

8.5.1.2 Coupling costs for time integration

In the transient case, the time integration algorithm is implicit Euler, which requires the solution of a nonlinear coupled conduction/ER problem for each time step. In Calore this is coupled to a forward Euler (explicit) predictor step. A baseline comparison between the three coupling strategies is to estimate the average cost per time step, which should be essentially the cost of the nonlinear solve. For the spatial discretization, we used linear Hex8 elements and solved the problem using Calore. From the data in Table 8.1, we observe that in all cases, the most expensive nonlinear solver

	time steps								
	8			15			60		
elements	WC	NOX	FC	WC	NOX	FC	WC	NOX	FC
1289	3.88	2.38	1.38	2.60	2.60	0.93	0.63	0.42	0.48
10312	23.38	15.88	13.75	15.27	12.47	10.33	6.87	6.37	5.23

Table 8.1. Comparison of average nonlinear solve times for 3D transient ER coupling problem.

was the weak coupling approach (WC). The fastest approach was almost always the full coupling approach (FC), with NOX typically performing closer to the FC case than the WC case.

The implications of this experiment are the following. First, if the time step selection is constrained to take many small implicit time steps, then weak coupling can be reliable and have comparable cost as the full coupling (consider the case of 1289 elements and 60 time steps). However, if one wishes to take much larger implicit time steps, then full coupling or JFNK coupling not only can be faster (case of 10312 elements and 8 time steps), but may be more reliable because of the possibility of failure of the weak coupling solver, even when the predictor is good.

8.5.1.3 Spatial adaptivity for the transient case

While failure of the weak coupling was not demonstrated in the previous example, failure was observed in the case of spatial mesh adaptivity within each implicit time step.

The adaptivity is driven by a standard gradient recovery error indicator in Calore, that uses lack of smoothness in the temperature gradients to compute local element error indicators. At each time step, the mesh was adapted until the global error indicator was smaller than a fixed relative tolerance. In Figure 8.2 we plot the global error indicator versus time for a uniform mesh, and for two adaptive cases with different error tolerances. In all cases, the nonlinear solver is the Weak Coupling case. For the larger tolerance of 0.08, the weak coupling converges and the problem runs to completion. However, for the smaller tolerance of 0.04, the weak coupling strategy eventually fails. We conjecture that the reason for the failure is that the adaptivity results in a tighter coupling within the discrete system. Alternatively the problem could be that we need to reduce the time step size when attempting to adapt the spatial mesh significantly.

8.5.1.4 Spatial adaptivity for the stationary case

In the last part of this example, we consider a stationary version of the previous problem. The Dirichlet boundary condition at $x = -1$ is the same, but the convective flux boundary condition at $x = 1$ is replaced by a Dirichlet boundary condition of $u = 1300K$. In this case the solution has

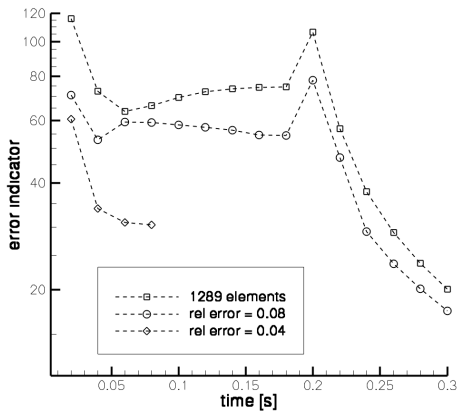


Figure 8.2. Spatial adaptivity failure from weak coupling in a 3D transient ER problem

larger gradients on the part of the domain and the enclosure nearer the surface $x = 1$, due to the larger temperature values there.

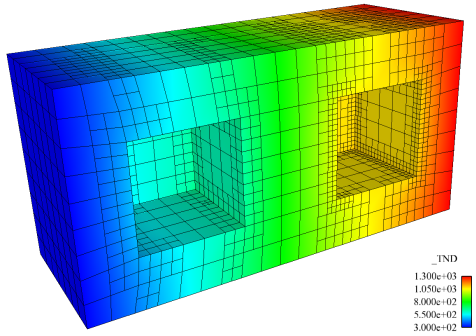


Figure 8.3. Cutaway of temperature solution after five adaptive mesh refinement steps for 3D stationary ER coupling problem.

In this case we would like to study the coupling performance under a sequence of adaptive mesh refinements of the spatial mesh. As in the previous problem, we can use computational cost as a metric by which to compare the coupling methods. In this case, we plot the global error indicator versus cumulative CPU time in Figure 8.4. Because adaptivity is often described as a method to optimize efficiency for a given error tolerance, such a comparison is natural. We could have chosen another scale such as number of elements or dofs, but due to the differing costs from the nonlinear solvers, this metric be less useful. We see in Figure 8.4 that in this case, weak coupling appears to be more efficient, reducing the global error indicator with less computational cost than JFNK coupling. This result may seem strange, since each JFNK solve typically requires only about 5–6 nonlinear iterations compared to hundreds or even thousands of iterations for the weak coupling approach. In addition, both nonlinear solvers use the same underlying linear solver (GMRES) and preconditioner (multilevel ML).

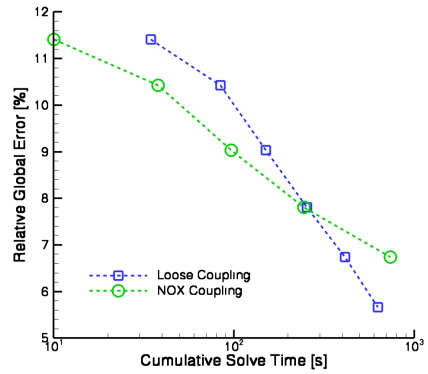


Figure 8.4. Comparison of global error indicator versus CPU time for a stationary adaptive problem using either weak coupling or JFNK coupling.

The difference is that the preconditioner for the conduction part of the weak coupling solver (Calore) has the correct sparsity pattern from the conduction problem. Thus, the linear solver typically only needs at most 10–20 iterations to converge, even on the adapted meshes. In contrast, the ML preconditioner for JFNK uses the conduction sparsity pattern to precondition the reduced coupled problem in eq.(8.30). The Jacobian for the reduced problem has many new non-zeros from the temperature coupling within the enclosure, which are omitted from the conduction sparsity pattern.

An interesting final comment is that the weak coupling solver actually fails on the next adaptive step, while JFNK coupling is able to solve it, although with a very large computational cost. This indicates the need to develop nonlinear solvers that are both robust and efficient.

8.5.2 1D Verification Problem

Our next problem is a radially symmetric heat transfer problem with an steady state analytic solution. The problem consists of a (infinite) cylinder inside of a cylindrical shell. Within the cylinder is a uniform heat generation source that drives the heat transfer. Heat is conducted through the cylinder, radiated to the outer shell, and conducted to the outer surface where a Dirichlet boundary condition is applied. The temperature profile computed by Calore using a fine mesh of linear triangle elements is shown in Figure 8.5.

8.5.2.1 Problem description

The one-dimensional analytic solution can be defined in terms of the radial coordinate r. Let the domain be $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 \equiv (0, r_1)$ and $\Omega_2 \equiv (r_2, r_3)$ with $0 < r_1 < r_2 < r_3$. We define the heat flux by

$$q(r) \equiv -k \, \frac{du}{dr}(r). \tag{8.31}$$

Within Ω the steady state heat conduction problem is

$$\frac{1}{r} \, \frac{d}{dr} \, (r \, q) = Q, \quad r \in \Omega. \tag{8.32}$$

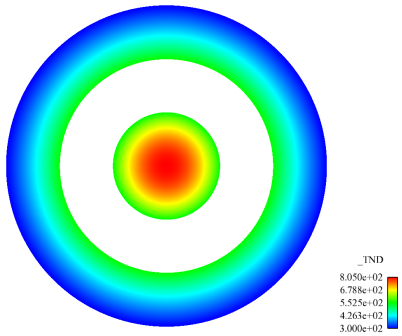


Figure 8.5. Temperature solution using a fine spatial grid for ER verification problem.

The source term Q is a positive constant in Ω_1 and zero in Ω_2. The left (inner) boundary condition is the standard zero flux condition

$$q(0) = 0, \tag{8.33}$$

and the right (outer) boundary condition is constant temperature

$$u(r_3) = u_3. \tag{8.34}$$

To close the problem, we need to specify the nonlinear radiative boundary conditions on the enclosure surfaces at $r = r_1$ and $r = r_2$. If G_i is the incident flux to r_i, for $i = 1, 2$, then the first set of conditions are

$$\begin{array}{rcl} q_1 & = & \epsilon_1\, \sigma\, u_1^4 - \epsilon_1\, G_1, \\ -q_2 & = & \epsilon_2\, \sigma\, u_2^4 - \epsilon_2\, G_2, \end{array} \tag{8.35}$$

where the subscripts denote evaluation at the point r_i. The incident radiation is defined in terms of the radiosity variable J_i by

$$\begin{array}{rcl} G_1 & = & F_{11}\, J_1 + F_{12}\, J_2 \\ G_2 & = & F_{21}\, J_1 + F_{22}\, J_2 \end{array} \tag{8.36}$$

The final equation couples the radiosities to the surface temperatures

$$\begin{array}{rcl} J_1 - (1-\epsilon_1)F_{11}J_1 - (1-\epsilon_1)F_{12}J_2 & = & \epsilon_1\sigma u_1^4 \\ J_2 - (1-\epsilon_1)F_{21}J_1 - (1-\epsilon_2)F_{22}J_2 & = & \epsilon_2\sigma u_2^4 \end{array} \tag{8.37}$$

The view factors can be found in many textbooks (23) and are defined as

$$F_{11} = 0, \quad F_{12} = 1, \quad F_{22} = 1 - r_1/r_2, \quad F_{21} = r_1/r_2. \tag{8.38}$$

This nonlinear problem can for solved analytically for the temperature field $u(r)$ in Ω, the incident fluxes G_i and the radiosities J_i in terms of the data $r_i, k_i, \epsilon_i, Q, u_3$. The solution to eqs.(8.31)-(8.34) can be written in terms of the unknowns $u_1 \equiv u(r_1)$ and $u_2 \equiv u(r_2)$ as

$$u(r) = \left\{ \begin{array}{ll} u_1 + \frac{Q}{4\,k_1}\,(r_1^2 - r^2), & 0 < r < r_1, \\ u_3 + (u_2 - u_3)\,\frac{\ln(r/r_3)}{\ln(r_2/r_3)}, & r_2 < r < r_3 \end{array} \right. . \tag{8.39}$$

with the associated fluxes on the enclosure

$$q_1 \equiv \frac{Q\, r_1}{2} > 0, \quad q_2(u_2) \equiv \frac{k_2}{r_2}\,\frac{(u_2 - u_3)}{\ln(r_3/r_2)} > 0. \tag{8.40}$$

The analytical expressions for u_1 and u_2 can then be computed as

$$u_2 \equiv u_3 + \frac{Q\, r_1^2}{2\, k_2}\, \ln(r_3/r_2), \quad u_1 \equiv \left(u_2^4 + (\rho_2\, (r_1/r_2) + \epsilon_2/\epsilon_1)\, q_1/(\epsilon_2\, \sigma)\right)^{1/4} \tag{8.41}$$

Using these values we can also solve for the radiosity variables

$$J_1 \equiv \sigma\, u_1^4 + (1 - 1/\epsilon_1)\, q_1, \quad J_2 \equiv J_1 - q_1. \tag{8.42}$$

In order to study this problem numerically, we select a set of nominal parameters, given in Table 8.2

Parameter	nominal value
r_1	$1\ m$
r_2	$2\ m$
r_3	$3\ m$
k_1	$0.1\ W/m{-}K$
k_2	$0.08\ W/m{-}K$
ϵ_1	0.8
ϵ_2	0.7
Q	$100\ W/m^3$
u_3	$300\ K$

Table 8.2. Nominal parameter values for verification problem.

8.5.2.2 Weak Coupling Convergence using Analytical Solution

A convenient feature of this problem is that we have the exact coupled analytic solution. One way to study the coupling is to use the analytic solutions to the weak coupled problem to study the convergence of the coupling algorithms in the absence of numerical discretization error. To this end, we wrote a Matlab script in order to iterate on the analytic temperature solution using the same weak coupling algorithm employed by Calore/Aria. In this case, we pass the current temperature values on the enclosure $\{u_1, u_2\}$ to the ER "solver," which solves the 2 by 2 system for the radiosities $\{J_1, J_2\}$ and returns the net fluxes

$$\epsilon_i\, \sigma\, u_i^4 - \epsilon_i\, G_i, \quad i = 1, 2.$$

Then the thermal conduction "solver" computes a new temperature from these fluxes including a linearization of the nonlinear u_i^4 terms. This new temperature is a function only of the previous choice of parameters $\{u_1, u_2\}$.

We analyzed the weak coupling error as was done in eq.(8.29) and derived an analytical expression for a bound on the linear error rate, defined in terms of the operator blocks of the Jacobian in eq.(8.26)

$$|(A + F'(u^n))^{-1}\, C\, B^{-1}\, G'(u^n)|. \tag{8.43}$$

Because the analytic model only iterates on the two temperature values $\{u_1, u_2\}$, this product is computed as the operator norm of a 2 by 2 matrix. In Table 8.3 we report the results of a simple parametric study obtained by varying the heat source Q while keeping the other parameters in Table 8.2 constant. The estimated rate was computed using eq.(8.29) evaluated at the exact solution, while the measured rate was computed by taking the ratio of the last two errors in the

Q	estimated rate = $\|G_{BB}\|$	measured rate	error in rate	iterations
10	0.988701923052248	0.988701850821153	7.223e-08	985
50	0.994947114469730	0.994947039953165	7.451e-08	2202
100	0.997674723966611	0.997674655519477	6.844e-08	4504
250	0.999514293801377	0.999514218102668	7.569e-08	18429
500	0.999897259132588	0.999897176360744	8.277e-08	72062

Table 8.3. Verification of weak coupling analytic rate estimate. In all cases, the nonlinear tolerance was $1E{-}8$.

nonlinear iteration sequence. We see that the estimated rate is exact up to the error in the nonlinear convergence.

As a final test, we compute the approximate finite element solutions from Calore for the same problem as in Table 8.3. Using a sequence of successively finer meshes of linear triangular elements, we compare both the nonlinear convergence rates and the number of iterations to the weak coupling analytic results. In Table 8.4 we see that as the mesh is refined (increasing DOFS), the nonlinear convergence rate approaches the same rate as we observed in Table 8.3 for the analytic coupled model. Interestingly, the trend is for the convergence rate to be smaller for the coarser meshes, suggesting that they exhibit faster convergence than for the finer meshes.

We also compare the total number of iterations for convergence to the tolerance of $1E{-}8$ in Figure 8.5. Here we see that the approximate solution typically requires more nonlinear iterations, with fewer iterations needed for finer meshes. This reasons for this trend are not clear, since the larger rates for the finer meshes would suggest that the finer meshes would require greater, not fewer nonlinear iterations.

	Source term Q				
DOFS	10	50	100	250	500
22	0.987952	0.994008	0.996979	0.999249	0.999120
39	0.987923	0.993918	0.996933	0.999264	0.999601
121	0.988544	0.994745	0.997534	0.999465	0.999832
382	0.988655	0.994890	0.997636	0.999502	0.999900
1501	0.988691	0.994933	0.997665	0.999512	0.999897
5799	0.988699	0.994944	0.997672	0.999514	0.999897
Analytic	0.988702	0.994947	0.997675	0.999514	0.999897

Table 8.4. Convergence rates of weak coupling computed using 2D finite element code Calore compared to rates for the weak coupling analytic model.

	Source term Q				
DOFS	10	50	100	250	500
22	1421	3047	6233	27369	123823
39	1349	2866	5811	25179	112983
121	1304	3053	6666	32159	154277
382	1188	2853	6328	31215	151887
1501	1042	2540	5674	28274	138581
5799	893	2210	4963	24911	122814
Analytic	985	2202	4504	18429	72062

Table 8.5. Nonlinear iterations for weak coupling computed using 2D finite element code Calore compared to iterations for the weak coupling analytic model.

8.5.3 Adjoint-based Error Estimator and Coupling

In our final example, we illustrate the importance of including all coupling effects into the Jacobian matrix, in the context of adjoint-based error estimation. For this type of error estimator, a linearized adjoint solve is needed, in order to compute the adjoint solution, which measures the local sensitivity of the solution to a given global quantity. While the adjoint Jacobian could be computed either by via NOX, or by directly implementing the fully coupled Jacobian, we chose the latter approach for simplicity.

8.5.3.1 Formulation of the error estimator

In this section we use the same notation as in eq.(8.26), abusing it to represent the continuum differential and integral operators and residuals.

Suppose that we are interested in the error for a global quantity that depends on the temperature u. For simplicity, we will assume that the quantity is represented by a weighted integral

$$\int_\Omega \psi\, u\, dx.$$

The associated adjoint problem is the solution to linearized coupled PDE: find (z, Φ):

$$\begin{pmatrix} A + F'(u_h) & -G'(u_h)^* \\ C^* & B^* \end{pmatrix} \begin{pmatrix} z \\ \Phi \end{pmatrix} = \begin{pmatrix} \psi \\ 0 \end{pmatrix} \tag{8.44}$$

The solution to this problem provides an error estimate for the error in the quantity of interest that is accurate up to higher order linearization error

$$\int_\Omega \psi\,(u - u_h)\, dx \approx \langle R^u(u_h, J_h), z - z_h\rangle + \left\langle R^J(u_h, J_h), \Phi - \Phi_h\right\rangle.$$

Here z_h and Φ_h are finite element approximations computed from the same spaces as u_h and J_h.

Because eq.(8.44) is infinite dimensional, we approximate it using the same finite element spaces as the original coupled problem. As a first implementation, the weighted residual of the ER problem $R^J(u_h, J_h)$ is ignored. Next, the weighted temperature residual $R^u(u_h, J_h)$ is replaced by the energy inner product of the primal and adjoint temperature errors. Finally the exact primal/adjoint temperature gradients are approximated by applying a gradient recovery operator G to the finite

element gradients. The resulting computable error estimator is then

$$\eta \equiv \int_{\Omega} k(G(\nabla u_h) - \nabla u_h) \cdot (G(\nabla z_h) - \nabla z_h)\, dx \tag{8.45}$$

8.5.3.2 Application of the error estimator

In this example, we illustrate the dramatic improvement in the error estimator computed using the coupled adjoint system (8.44) over solving the single-physics decoupled adjoint problem given by

$$(A + F'(u_h))^* \, z = \psi. \tag{8.46}$$

The 2D problem consists of a square domain within a hollow square, similar to the previous example problem. On the right exterior boundary, a heat flux is applied, and on the remaining boundary the temperature is held constant. As a result of the heat flux, the outer domain heats up and radiates energy to the inner domain, causing it to heat up. The steady state temperature distribution is shown in Figure 8.6. The strong coupling between the outer an inner domains by means of the ER

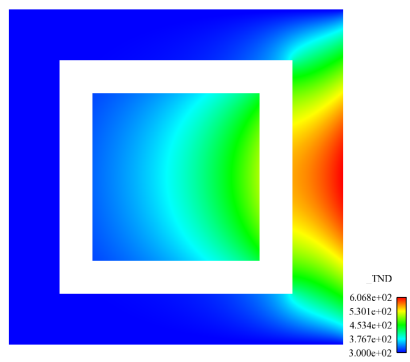


Figure 8.6. Temperature solution for the adjoint ER problem.

physics is clearly evident in Figure 8.6.

We choose the quantity of interest to be the average temperature on the inner domain, represented by a simple piecewise constant function ψ.

In Figure 8.7 we compare the temperature component of the fully coupled adjoint problem of eq.(8.44) with the adjoint temperature obtained from the decoupled adjoint problem of eq.(8.46). There are clear differences in the solutions within the inner block, with the fully coupled adjoint solution having its maximum near the right side of the inner domain. More striking is the difference between the adjoint solutions in the outer domain. The weak coupled adjoint solution is zero in the outer block, since there is no coupling between the blocks in eq.(8.46). In contrast, the fully coupled adjoint solution has a nonzero component in the outer block that determines the effect of local temperature values, and corresponding local errors, in the outer block on the average temperature in the inner block.

In Figure 8.8 we plot the corresponding temperature solutions on meshes that have been adapted using the error indicator in eq.(8.45) defined using either the full coupling or weak coupling adjoint solve. The estimator using full coupling adjoint solves produces dramatically different mesh adaptivity, that properly adapts the mesh to resolve the error that has been propagated from the outer block into the inner block via the ER coupling.

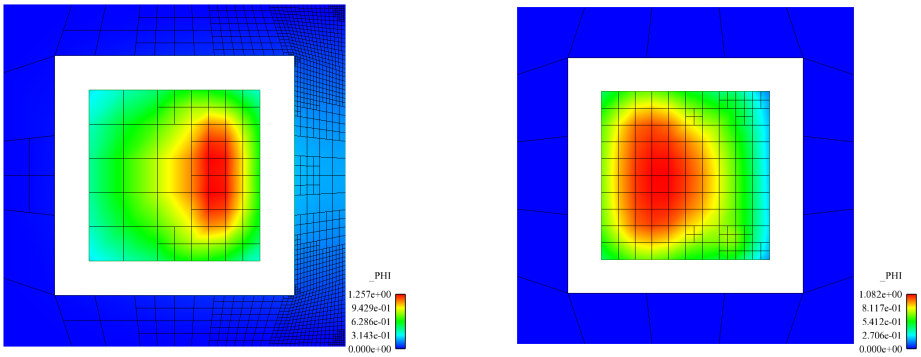


Figure 8.7. Adjoint temperature field. (left) fully coupled adjoint (right) weak coupled adjoint.

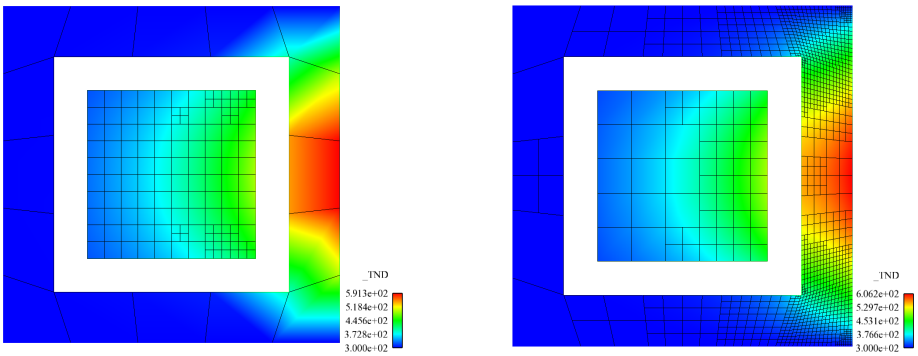


Figure 8.8. Temperature field. (left) fully coupled adjoint (right) weak coupled adjoint.

8.6 Conclusions

Based on the results in this chapter, we make the following conclusions:

1. JFNK coupling can be very efficient when compared to weak coupling and can approach the performance of full Newton coupling.
2. Efficient and appropriate preconditioners are essential for NOX, especially in the context of spatial adaptive mesh refinement.
3. When using large implicit time steps, efficient nonlinear solvers such as NOX or a full Newton coupled approach can perform much faster than successive substitution.
4. We have identified a good verification test problem that has a 1D analytic solution, suitable for testing coupling algorithms.
5. Finally, we have demonstrated the importance of a fully coupling adjoint Jacobian operator to the success of adjoint-based error estimators.

Chapter 9

Project Summary and Future Work

We feel the stated goals of this project have been met by our development, deployment and demonstration of stronger Newton-based coupling algorithms which require no application data beyond weak coupling but which can make use of any additional data to improve algorithm performance.

Our work in Chapter 8 is continuing into the areas of Uncertainty Quantification (UQ) and Error Estimation, the success of which is intimately connected to the quality of coupled multi-physics solves. Moreover, our ability to numerically compute quantities such as complete Jacobian matrices will prove invaluable to the quality of this ongoing work.

Our interactions with other communities at Sandia interested in robust coupled multi-physics simulations has led to the adoption of our algorithms by the Burner Reactor Integrated Safety Code (BRISC) LDRD. We are actively involved in the BRISC project, providing guidance from lessons learned as well as tuning and specializing our core Newton-based coupling algorithms.

Our presence in the Sierra applications framework is beginning to be utilized more, and we anticipate this to increase.

Our work on this project also identified valuable future studies in the areas of mathematically smooth and physically conservative data transfers across shared interfaces, improved software practices, and the beginnings of cheap metrics to assess and guide the selection of coupling algorithms. We could easily envision the latter functionality comprising the core of adaptive coupling algorithms that change during the simulation to optimize the overall algorithm efficiency, robustness, or some user defined weighting of each.

Finally, we recognize that weak coupling is inherently ill-suited to advanced Verification and Validation of coupled multi-physics simulation capabilities. This is due largely to the inability to treat the coupled problem as a whole which makes characteristics such spatial and temporal order of accuracy ambiguous and methods such as manufactured solutions difficult or impossible to apply at the coupled problem level. We anticipate our Newton-based coupling algorithms, which recover a complete view of the coupled problem using weak coupling data, to prove valuable in future V&V activities.

References

[1] Hermann Matthies. private communication. 1

[2] D. A. Knoll and D. E. Keyes. Jacobian-free newton-krylov methods: a survey of approaches and applications. *J. Comp. Phys.*, 132(2):357–397, 2004. 2.3.1

[3] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996. 2.3.2.2

[4] "A. Yeckel; A. Pandy and J.J. Derby. Fixed-point convergence of modular, steady-state heat transfer models coupling multiple scales and phenomena for melt-crystal growth . *Int. J. Num. Meth. Engng.*, 67(12):1768–1789, 2006. 3.1, 3.1.1, 3.1.1

[5] J. N. Shadid D. L. Ropp and C. C. Ober. Studies on the accuracy of time integration methods for reaction-diffusion equations. *J. Comp. Phys.*, 194(2):544–574, 2004. 3.2

[6] A. Haselbacher and J. Blazek. Accurate and Efficient discretization of Navier-Stokes Equations on Mixed Grids. *AIAA Journal*, 38(11):2094–2102, November 2000. 4.1.2

[7] H. Luo, J. D. Baum, and R. Lohner. Edge-Based Finite Element scheme for the Euler Equations. *AIAA Journal*, 32(6):1180–1190, June 1994. 4.1.2

[8] T. M. Smith, R. W. Hooper, C. C. Ober, A. A. Lorber, and J. N. Shadid. Comparison of Operators for Newton-Krylov Method for Solving Compressible Flows on Unstructured Meshes. *AIAA Paper 2004-0743*, January 2004. 4.1.2

[9] T. M. Smith, R. W. Hooper, C. C. Ober, and A.A. Lorber. Intelligent Nonlinear Solvers for Computational Fluid Dynamics. *AIAA Paper 2006-1483*, January 2006. 4.1.2

[10] H. K. Moffat. Cads: Cantera aerosol dynamics simulator. Sandia Report SAND2007-4216, Sandia National Laboratories, Albuquerque, NM, 2007. 6.1

[11] E. R. Keiter, S. A. Hutchinson, R. J. Hoekstra, T. V. Russo, E. L. Rankin, R. P. Pawlowski, R. L. Schiek, D. A. Fixel, D. N. Shirley, P. M. Campbell, C. W. Bogdan, and S. D. Wix. Xyce parallel electronic simulator: Users' guide version 3.1. Technical Report SAND2006-XXXX, Sandia National Laboratories, 2006. 7

[12] R. C. Melville, L. Trajkovic, S. Fang, and L. T. Watson. Artificial parameter homotopy methods for the dc operating point problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(6):861–877, 1993. 7, 7.1

[13] E. R. Keiter, S. A Hutchinson, R. J. Hoekstra, E. L. Rankin, T. V. Russo, and L. J Waters. Xyce parallel electronic simulator design: Mathematical formulation, version 2.0. Technical Report SAND2004-2283, Sandia National Laboratories, 2004. 7, 7.1

[14] N. B. G. Rabbat, A. L. Sangiovanni-Vincentelli, and H. Y. Hsieh. Multilevel newton algorithm with macromodeling and latency for the analysis of large-scale nonlinear circuits in the time domain. *IEEE Transactions on Circuits and Systems*, 1979. 7.1

[15] K. Mayaram and D. O. Pederson. Coupling algorithms for mixed-level circuit and device simulation. *IEEE Transactions on Computer Aided Design*, 11(8):1003–1012, 1992. 7.1

[16] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes. An analysis of approximate nonlinear elimination. *SIAM J. Sci. Comput.*, 17(2):538–559, 1996. 7.1

[17] James T. Wilkes. A new method for solving systems of nonlinear equations in circuit simulation. Technical report, CiteSeer [http://cs1.ist.psu.edu/cgi-bin/oai.cgi] (United States), 1994. 7.1

[18] D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson. Nonlinear elimination in aerodynamic analysis and design. In L. T. Biegler, O. Ghattas, and M. Heinkenschloss, editors, *Large-scale PDE-Constrained Optimization*, volume 30 of *Lecture Notes in Computational Science A*, pages 17–43. Springer-Verlag, 2003. 7.1

[19] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996. 7.1

[20] A. G. Salinger, E. A. Burroughs, R. P. Pawlowski, E. T. Phipps, and L. A. Romero. Bifurcation tracking algorithms and software for large scale applications. *INTERNATIONAL JOURNAL OF BIFURCATION AND CHAOS*, 15(3):1015–1032, 2005. 7.1

[21] Introduction to Device Modeling and Circuit Simulation. *T. Fjeldy and T. Ytterdal and M. Schur*, chapter 3, pages 67–69. Wiley Interscience, 1998. 7.2

[22] Aristide Mbiock and Roman Weber. *Radiation in enclosures*. Springer-Verlag, New York, 2000. 8.2.2, 8.2.2

[23] Robert Siegel and John R. Howell. *Thermal radiation heat transfer*. Taylor & Francis, New York, 2002. 8.2.2, 8.5.2.1

[24] Naji Qatanani and Monika Schulz. Analytical and numerical investigation of the fredholm integral equation for the heat radiation problem. *Appl. Math. Comp.*, 175:149–170, 2006. 8.2.2, 8.2.2, 8.2.2

DISTRIBUTION:

1 MS 1318
D. G. Bader, 1416

1 MS 1318
R. A. Bartlett, 1416

1 MS 0382
D. G. Baur, 1543

1 MS 0382
S. W. Bova, 1541

1 MS 0382
B. Carnes, 1543

1 MS 0826
D. K. Gartling, 1500

1 MS 0382
S. E. Gianoulakis, 1543

1 MS 0382
M. W. Glass, 1541

1 MS 0836
R. E. Hogan, Jr., 1516

10 MS 1318
R. Hooper, 1416

1 MS 0836
M. M. Hopkins, 1514

1 MS 0836
J. S. Lash, 1514

1 MS 0372
C. S. Lo, 1524

1 MS 0836
H. K. Moffat, 1514

1 MS 0380
H. S. Morgan, 1540

1 MS 0836
D. R. Noble, 1514

1 MS 0836
P. K. Notz, 1514

1 MS 1318
R. P. Pawlowski, 1416

1 MS 1318
E. T. Phipps, 1416

1 MS 0824
M. R. Prairie, 1510

1 MS 1318
A. G. Salinger, 1416

1 MS 0836
P. R. Schunk, 1514

1 MS 0316
J. N. Shadid, 1437

1 MS 0316
T. M. Smith, 1433

1 MS 1318
J. R. Stewart, 1411

1 MS 0382
S. R. Subia, 1541

1 MS 0382
A. B. Williams, 1543

1 MS 1318
D. E. Womble, 1410

1 MS 1070
C. C. Wong, 1526

2 MS 0619
Review & Approval Desk, 4916

1 MS 0123
D. L. Chavez, 1011

1 MS 0899
Technical Library, 9536

Sandia National Laboratories