Modeling Large Accelerator Structures with the Parallel Field Solver Tau3P[*]

Michael Wolf, Adam Guetz, and Cho-Kuen Ng
Stanford Linear Accelerator Center
{mwolf,guetz,cho}@slac.Stanford.edu

ABSTRACT

In this paper Tau3P, a parallel 3-D distributed-memory time domain electromagnetic solver that uses unstructured meshes to model large accelerator structures is discussed. The theoretical background upon which Tau3P is based as well as many of the key features that have been implemented in this code are described briefly. The different types of problems that Tau3P has been used to solve are presented. The parallel performance obtained with Tau3P is briefly discussed and as well as the high performance computing issues addressed in this code. Finally, the features that are currently being implemented and research that still remains to be done are expounded.

I. INTRODUCTION

The increasingly demanding design requirements of next-generation particle accelerators have placed heavy emphasis on the accuracy and reliability of electromagnetic software. There is a push to model entire accelerator structures to design tolerances. Traditional electromagnetic codes cannot solve the complicated and enormous geometries needed to model these large accelerator structures. These traditional codes have difficulties obtaining the needed accuracy in a reasonable amount of time. Therefore, it is useful to use more modern programming and modeling paradigms in order to achieve this goal.

One such paradigm is the use of unstructured mesh instead of the more traditional finite difference method with structured mesh for time domain problems (FDTD). These FDTD methods have the advantage of being accurate and fast when the geometry is very regular. However, most FDTD methods are less effective when the geometry cannot be easily approximated with an orthogonal mesh and have difficulty modeling curved geometries without using small mesh step sizes that result in more memory and longer runtime. Many accelerator structures have an irregular geometry and thus FDTD algorithms may not be the best solution. One solution is to use unstructured meshes that fit the contours of the irregular accelerator geometries. However, finding an accurate and efficient algorithm that works with unstructured meshes can be difficult. N.K. Madsen proposed a method for unstructured meshes he claims to be accurate and also computationally efficient [1]. His method was derived using a discrete surface integration (DSI) technique. This DSI method uses a dual grid as well as a primary grid and allows for unstructured meshes made up of hexahedra, tetrahedra, triangular prisms, and pyramids. The algorithm reduces to a finite difference method for the orthogonal case, which is important for accuracy and efficiency. Solving Maxwell's equations, the electric field projections are

[*] Work supported by the U.S. DOE Contract No. DE-AC03-76SF00515.

calculated on the edges of the primary mesh and the magnetic field projections are calculated on the edges of the dual mesh (Figure 1).

$$\oint E \bullet ds = -\iint \frac{\partial B}{\partial t} \bullet dA$$

$$\oint H \bullet ds^* = \iint \frac{\partial D}{\partial t} \bullet dA^* + \iint j \bullet dA^*$$

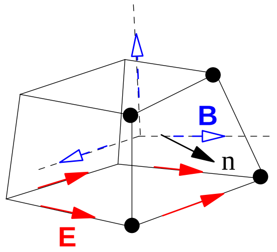


Figure 1: Maxwell's Equations and Modified Yee Grid.

This DSI method solves Maxwell's equations using a leapfrog scheme for time advancement in which the electric field projections are solved at every integral time step (t) and the magnetic field projects are solved at every half time step (t+½).

High performance computing is another important paradigm that is rapidly becoming an essential part of modeling particle accelerator structures. In order to get the accuracy in the simulations needed to design structures, the meshes must be sufficiently detailed to capture the intricacies of the geometries, resulting in more memory and longer runtimes. More ambitious simulations of entire accelerator structures would be impossible to simulate on a single processor machine, requiring terabytes of memory and years of runtime. Therefore, parallel codes are necessary to solve the more complicated and ambitious accelerator problems.

Under a DOE Grand Challenge and SCIDAC, the Advanced Computations Department (ACD) at the Stanford Linear Accelerator Center (SLAC) is developing parallel codes to solve difficult accelerator problems on high performance computing platforms. One of ACD's codes which uses these aforementioned modern paradigms to help meet these ambitious design requirements is Tau3P, a parallel 3-D distributed-memory time domain electromagnetic solver. Tau3P uses the DSI method with unstructured meshes to fit the contours of complicated accelerator structures and model large structures with a high degree of accuracy. Tau3P also uses domain decomposition and MPI to parallelize these large accelerator problems, allowing large problems to be solved that would serially be impossible due to memory and time constraints.

II. TAU3P IMPLEMENTATION

Tau3P is written in C++ for portability and maintainability of the code. It runs on Linux clusters and has also been ported to the Cray T3E and IBM SP at NERSC. Tau3P is a parallel code that uses MPI for inter-process communication. In the development of Tau3P, a concerted effort has been made to use existing software libraries as often as possible when they provide the necessary functionality. Figure 2 shows the basic design overview of Tau3P and how it is used with other applications and libraries in the structure design process. The following sections discuss various aspects of the Tau3P implementation.

1. Mesh Implementation

As mentioned before, Tau3P supports an unstructured grid for conformal meshes. The code supports hexahedra, tetrahedra, triangular prisms, and pyramids. CUBIT [2] is used to generate hexahedron dominant meshes for Tau3P.

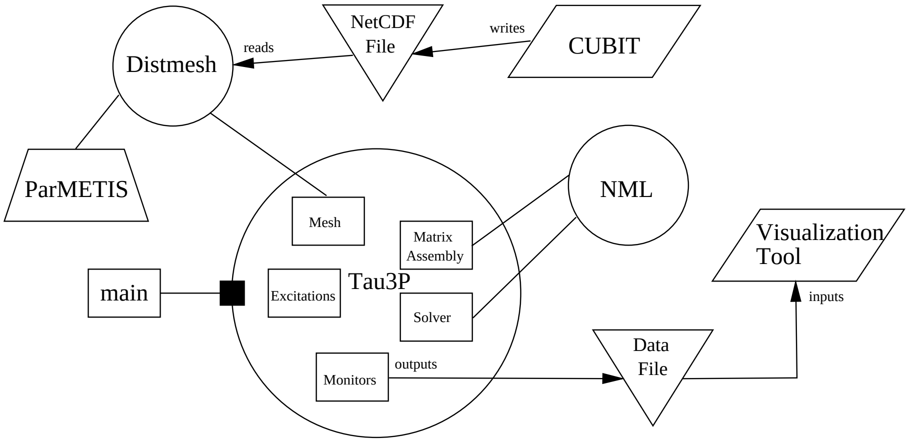


Figure 2: Tau3P Design Overview.

This meshing step is often a very difficult one. For complex structures, a well-meshed geometry is essential for long-term stability of the Tau3P DSI implementation. CUBIT outputs a mesh that is then converted into a binary mesh format, NetCDF [3]. Tau3P uses the distributed mesh library DistMesh to have each processor read in its share of the mesh and store it in the distributed mesh object DM_Mesh. After the initial naïve partitioning, DistMesh uses ParMETIS [4], a parallel mesh partitioner, to repartition the mesh in an attempt to minimize communication and evenly distribute the elements. Next the edges and faces (faces numbered to effectively number dual edges) that are to be degrees of freedom are identified based on the boundary conditions assigned in the mesh.

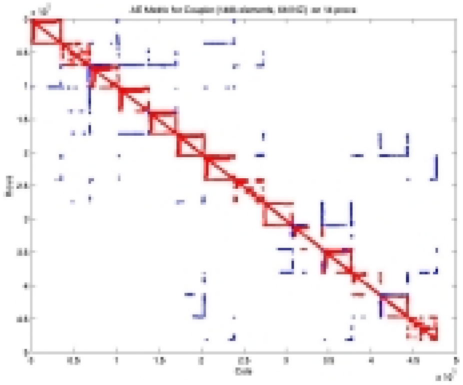


Figure 3: Non-zero Pattern for Typical AE Matrix.

2. Matrix Assembly

Matrices are assembled from the mesh information in order to efficiently solve the two equations produced by the DSI formulation of Maxwell's equations. The two matrices are stored in a matrix class belonging to the numerical methods library (NML): one (AH) to solve for the electric field projections and another (AE) to solve for the magnetic field projections. Two vectors from the NML library are also constructed to store the electric and magnetic field projections. The two matrices are both very sparse. The number of non-zeros per row can vary from 2 to 20 depending on the orthogonality of the elements surrounding the degree of freedom. A typical matrix can be seen in Figure 3.

3. Solver

The Tau3P solvers use a leapfrog scheme to calculate the field projections at each time step. The most computationally expensive part of each solve step is the matrix/vector multiplication performed by the NML matrix classes. Tau3P has two different solvers: one for orthogonal meshes and one for non-orthogonal meshes. The non-orthogonal solver has a filter applied every 500 time steps. The Tau3P DSI algorithm generates two errors that are proportional to electromagnetic frequency: a truncation error when solving Maxwell's equations, and an averaging error for determining electromagnetic fields at a node. Since both errors increase with frequency, a low-pass filter, placed carefully to avoid a damping effect, can remove a large portion of the error and greatly improves stability and accuracy of the algorithm. Although this filter is computationally expensive, it is necessary for the stability of complex problems with many time steps.

4. Excitations

Different types of excitations are set in Tau3P to drive fields through the structure so that the response to these fields can be studied. Tau3P handles the following excitations with appropriate boundary conditions: dipole excitation, waveguide excitation, and beam excitation. Dipole excitation consists of a dipole being excited at some point along a structure. Waveguide excitation is when modes are excited through a port in a waveguide to study the transmission properties of that structure. With beam excitation, the fields are excited along the z-axis of structure to simulate a beam passing through the structure.

5. Other Miscellaneous Features

Tau3P also has many other miscellaneous features that are useful in doing accelerator design. Tau3P has several monitors that output electric and magnetic field data in various forms to be later processed by visualization tools to examine different aspects of the simulation. Tau3P also has built-in checkpoint and restart capabilities. These capabilities allow jobs to be restarted if a machine crashes or if the job cannot finish in time allotted by a queue. Another key feature of Tau3P is that it is built as a library with external hooks so it can be driven by external applications. This allows other applications to interact with Tau3P and extend the usefulness of the Tau3P field calculations.

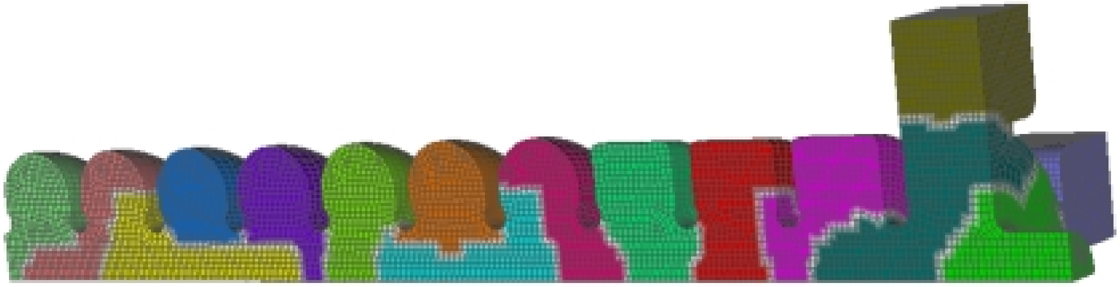


Figure 4: Mesh of RDDS Output End (10 Cells) with Coupler.

III. EXAMPLES OF TAU3P CALCULATIONS

1. Dipole Field Excitation Problem

Tau3P has been used with dipole field excitation to study part of the NLC RDDS structure (last 10 cells with coupler) (Figure 4). A dipole was excited in this structure towards the end away from the coupler and the variation in fields at a point near the coupler was monitored over time. An FFT of this time signal resulted in the dipole mode spectrum shown in Figure 5 with several pronounced resonant frequencies that compare favorably to measured frequencies (<.14% difference) (Figure 6).

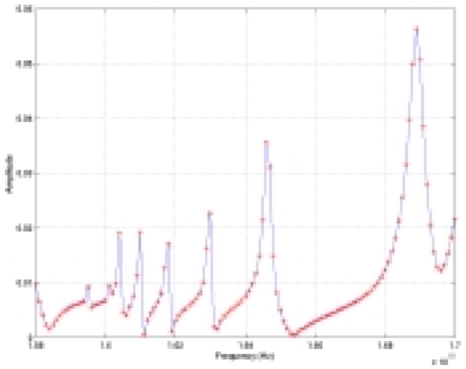
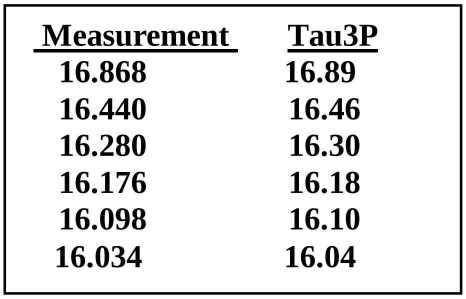


Figure 5: FFT of Dipole Mode Spectrum.



Measurement	Tau3P
16.868	16.89
16.440	16.46
16.280	16.30
16.176	16.18
16.098	16.10
16.034	16.04

Figure 6: Table of Peak Frequencies (measured and Tau3P).

2. Waveguide Excitation

Tau3P has been used with waveguide excitation in designing an input coupler to properly match the NLC RDDS (Figure 7). A wave was driven through the input coupler and the transmission and reflection through the coupler were monitored.

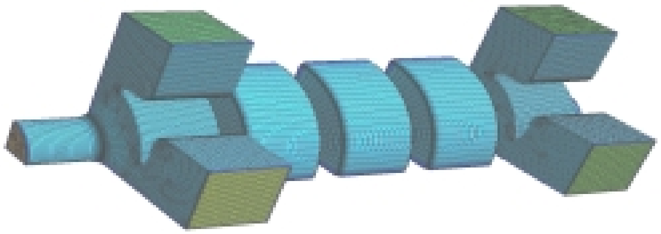


Figure 7: Mesh of NLC RDDS Input Coupler.

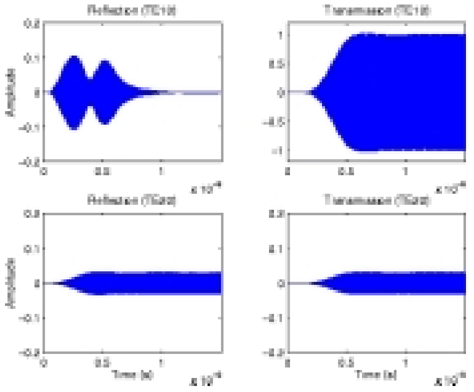


Figure 8: Transmission/Reflection Plots for NLC RDDS Input Coupler.

Figure 8 shows the transmission and reflection coefficients for two different modes for one run of this coupler. Using Tau3P with a high quality mesh realistic model to determine the coupler with the most favorable transmission properties, the Tau3P matching calculations provided accurate cavity dimensions for fabrication.

3. Beam Excitation

Tau3P with beam excitation has been used to study the standing wave (SW) detuned structure (Figure 9). Beam excitation was used to simulate a beam propagating through the structure. The fields were monitored as they varied with time to record the response of the structure to the beam. Figures 10 and 11 show dipole mode spectra resulting from the fields excited by the beam excitation obtained by taking FFTs of the field monitor time signals. Figure 10 shows frequencies inside the SW structure and Figure 11 shows the frequencies exiting through the coupler.

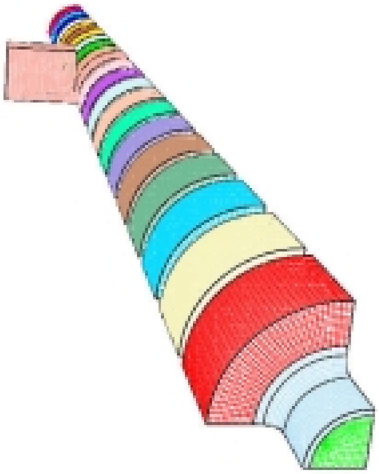


Figure 9: Mesh of Standing Wave Detuned Structure.

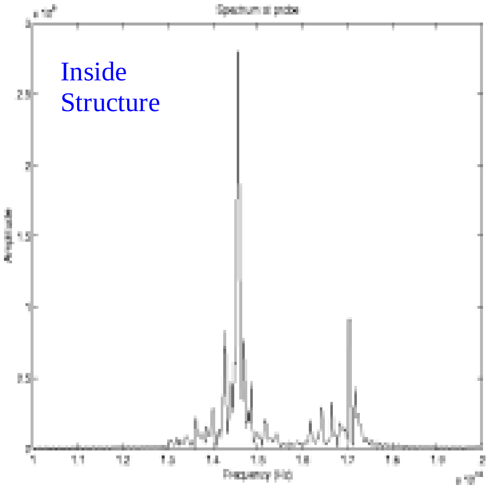


Figure 10: Beam Excited Dipole Mode Spectrum inside the SW Structure.

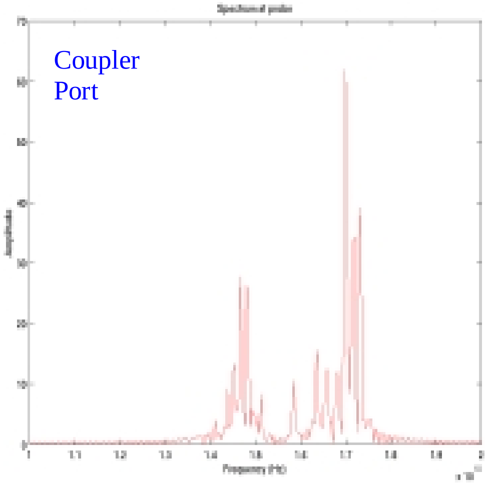


Figure 11: Beam Excited Dipole Mode Spectrum at Coupler Port.

There is a qualitative inverse relationship between the first and second band in the two FFTs. The more a mode couples out of the structure (Figure 11, second band), the weaker it will be in the SW structure (Figure 10, second band).

IV. TAU3P PERFORMANCE

Tau3P has been run on meshes containing 2 million elements with tens of millions of degrees of freedom and has been successfully used in the design and analysis of the Next Linear Collider (NLC) RDDS and the PEP-II Interaction Region (IR) structure. Figure 12 shows the parallel speedup for a typical Tau3P run. This Tau3P run for a mesh of 1.1 million hexahedra on a 16 processor Linux cluster has a speedup of only 13.7. One of the major reasons for this less than optimal parallel efficiency in Tau3P is poor load balancing of computation. ParMETIS attempts to minimize communication and evenly distribute the number of elements when partitioning the Tau3P mesh. However, distributing the number of elements equally will not guarantee that computation is load balanced since non-orthogonal elements result in more matrix non-zeros than do orthogonal elements and the orthogonality of the mesh is not uniform. Figure 13 shows a matrix created from a non-weighted ParMETIS partitioning routine. Process 10’s portion of the matrix has approximately 2.5 times the number of non-zeros contained by process 9’s portion of the matrix.

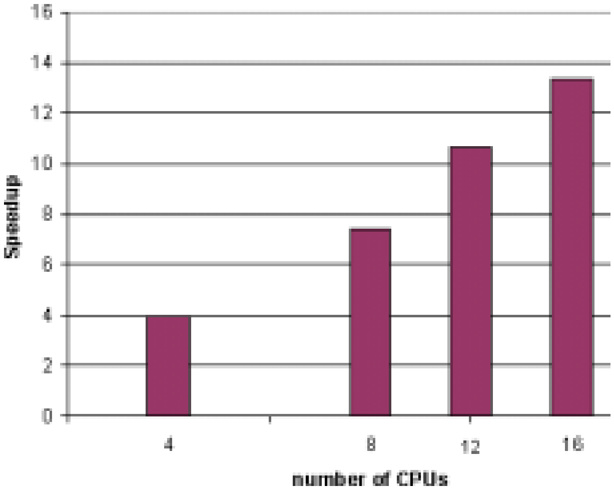


Figure 12: Parallel Efficiency.

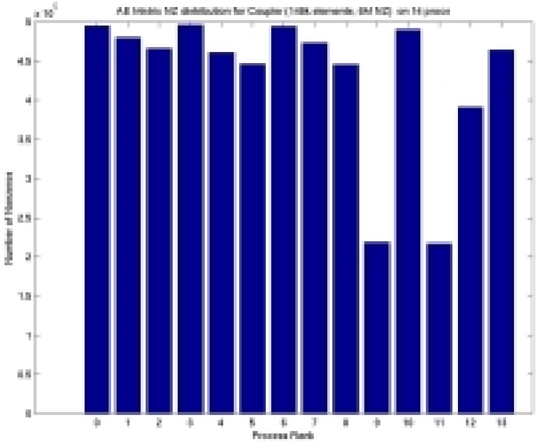


Figure 13: Number of AE Matrix Non-zeros on each Processor.

This computational load imbalance can be mostly corrected by passing weights to ParMETIS based on the orthogonality of the elements. However, the resulting partition does a less effective job of minimizing communication so the increased communication cost mitigates the expected improvement from the load balanced computation. A different communication scheme with these weights may

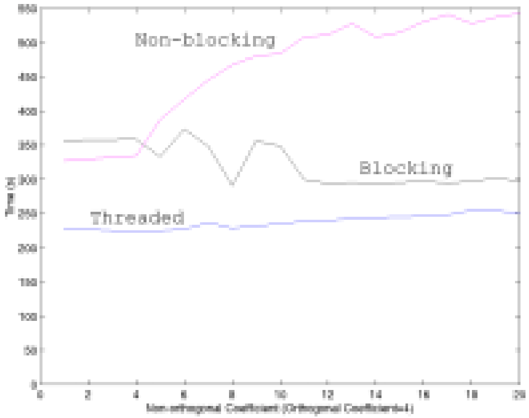


Figure 14: Runtime of Different Communication Algorithms and Non-orthogonal Weights.

lead to a more optimal solution. Figure 14 shows three different communication schemes non-blocking (MPI_Isend, MPI_Irecv), blocking(MPI_Send, MPI_Recv), and threaded (MPI calls are threaded) for different non-orthogonal weights (orthogonal weights are set to 4). These schemes show varied success but none seem to improve significantly as the computation becomes load balanced. However, some scheme may be found that will greatly improve performance.

V. DISCUSSION

Tau3P is currently a useful parallel time domain application for studying and designing various aspects of accelerator structures. However, many improvements still need to be made in order to solve the really large accelerator structure problems and to increase the variety of problems Tau3P can help solve. The main aspect of Tau3P that needs to be improved is the performance. Although the performance is adequate on small Linux clusters, it is very lacking once scaled past 64 processors. An optimal load balancing solution needs to be found, which can balance the computation and also minimize communication. Another area for improvement is the stability of the code. There are intrinsic instabilities in the DSI method that are mitigated for most problems by using filters the described above. However, there are certain problems that are very difficult to build a high enough quality mesh to run with stability. Additional features are being added that will allow Tau3P to solve more problems. Tau3P will soon support dielectric materials and there are plans to implement lossy materials in Tau3P. Tau3P is also starting to be used to calculate fields for particle tracking simulations [5]. When used in particle simulation, Tau3P is used as a library that is driven by the particle tracking code. The particle tracking code calls an initialization step in Tau3P and then asks Tau3P for fields at certain time steps. Tau3P calculates the fields until that time step and returns the fields that have been requested.

REFERENCES

[1] N.K. Madsen. Divergence Preserving Discrete Surface Integral Methods for Maxwell’s Curl Equations Using Non-orthogonal Unstructured Grids, Journal of Computational Physics, 119, pp.34-45, 1995.

[2] CUBIT, Version 5.0, Sandia National Laboratories, (2001),
URL: http://endo.sandia.gov/cubit

[3] NetCDF, Version 3.4, Unidata Program Center, (1998),
URL: http://www.unidata.ucar.edu/packages/netcdf/

[4] ParMETIS, Version 2.0, University of Minnesota, (1998),
URL: http://www-users.cs.umn.edu/~karypis/metis/parmetis/

[5] V. Ivanov et al. Particle Tracking Algorithms for 3D Parallel Codes in Frequency and Time Domains. To appear in Proceedings of ACES 2002, Monterey, 2002.