Scientific Discovery through Advanced Computation Solicitation 01-07
SciDAC Integrated Software Infrastructure Centers

The Terascale Simulation Tools and Technologies (TSTT) Center

Center Director

Jim Glimm
Director, Center for Data Intensive Computing, Brookhaven National Laboratory
Chair, Department of Applied Mathematics and Statistics, SUNY at Stony Brook, NY 11794
Phone: (631) 632-8355
Fax: (631) 632-8490
Email: glimm@bnl.gov

University Principal Investigator

Mark Shephard
Scientific Computation Research Center
Rensselaer Polytechnic Institute
Phone: (518) 276-6795
Email: shephard@scorec.rpi.edu

Co-Principal Investigators

David Brown, Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Phone: (925) 424-3557
E-mail: dlb@llnl.gov

Ed D'Azevedo, Computer Science and Mathematics Division
Oak Ridge National Laboratory
Phone: (865) 576-7925
Email: e6d@ornl.gov

Joe Flaherty, Computer Science Department
Rensselaer Polytechnic Institute
Phone: (518) 276-6348
Email: flaherje@cs.rpi.edu

Lori Freitag, Mathematics and Computer Science Division
Argonne National Laboratory
Phone: (630) 252-7246
E-mail: freitag@mcs.anl.gov

Patrick Knupp, Parallel Computing Sciences Department
Sandia National Laboratories
Phone: (505) 284-4565
Email: pknupp@sandia.gov

Harold Trease, Theory, Modeling and Simulation Directorate
Pacific Northwest National Laboratory
Phone: (509) 375-2602
Email: het@pnl.gov

Abstract

The primary objective of the Terascale Simulation Tools and Technologies (TSTT) center is to develop technologies that enable application scientists to easily use multiple mesh and discretization strategies within a single simulation on terascale computers. We will focus our efforts in the areas of high-quality, hybrid mesh generation for representing complex and possibly evolving domains, high-order discretization techniques for improved numerical solutions, and adaptive strategies for automatically optimizing the mesh to follow moving fronts or to capture important solution features. We will encapsulate our research into software components with well-defined interfaces that enable different mesh types, discretization strategies, and adaptive techniques to interoperate in a "plug and play" fashion. All software will be designed for terascale computing environments with particular emphasis on scalable algorithms for hybrid, adaptive computations and single processor performance optimization. To ensure the relevance of our research and software developments to the SciDAC goals, we will collaborate closely with both SciDAC application researchers and other ISIC centers. In particular, we will insert existing TSTT technologies into fusion, accelerator design, climate modeling, and chemically reacting flow applications, and thereby have both significant near-term impact on those efforts as well as receive the feedback necessary to ensure our success. We will show the potential long-term impact of TSTT center research by deploying hybrid solution strategies in SciDAC applications and demonstrate the ability of researchers to easily obtain superior simulation results.

Table of Contents

1 Introduction

Terascale computing provides an unprecedented opportunity to achieve numerical simulations at levels of detail and accuracy previously unattainable. DOE scientists in many different application areas can reach new levels of understanding through the use of high-fidelity calculations based on multiple coupled physical processes and multiple interacting physical scales. As a particular example from the SciDAC application areas, consider simulation-based accelerator design. To evaluate a proposed design, a two-stage process is used in which Maxwell's equations are first solved to determine the particle forces that are then provided as input to a PIC (particle-in-cell) code that computes the particle trajectories. Researchers have found that structured grids are best in the beam center while an unstructured mesh is best for the waveguides and other areas of complex geometry. Therefore, the simulation would ideally use a high-quality, conformal joining of structured and unstructured meshes. Furthermore, adaptive techniques are required to properly resolve the space-charge effect near the center of the beam and in areas of complex geometry. Adaptive techniques that can accommodate both structured and unstructured grids are therefore required for use with the ideal hybrid grid just described. Moreover, the difficulties associated with generating meshes on complex geometries inhibits the use of distinct, optimally configured, meshes in the Maxwell and PIC codes. This example is not unique; many DOE Office of Science applications (including those in SciDAC) have similar issues. For example, fusion and climate applications have need of both adaptive methods and advanced discretization technology; chemically reactive flows are often simulated on complex geometries, and depend on adaptive methods to achieve adequate efficiencies; and biological modeling will only be possible with advanced domain representations and interfaces. The optimal route to superior simulation in these and many other areas, and frequently the only way to obtain useful answers, is to use adaptive, composite, hybrid approaches as described above. Unfortunately, the lack of easy-to-apply, interoperable meshing, discretization, and adaptive technologies severely hampers the realization of this optimal path. The *Terascale Simulation Tools and Technologies (TSTT) Center* recognizes this critical gap, and will, as its central goal, address the technical and human barriers preventing the effective use of powerful adaptive, composite, and hybrid methods.

In today's environment, there are many tools available that generate a variety of mesh types ranging from unstructured meshes of various types to overlapping structured meshes and hybrid meshes that include both structured and unstructured components (for an extensive list, see Robert Schneiders' web page [Sch01]). Approximation techniques used on these meshes include finite difference (FD), finite volume (FV), finite element (FE) (e.g., [Hughes00]), spectral element (SE) (e.g., [BelMad94]), and discontinuous Galerkin (DG) methods (e.g., [CocKar00]). Any combination of these mesh and approximation types may be used to solve PDE-based problems. The fundamental concepts are the same for all approaches: some discrete representation of the geometry (the mesh) is used to approximate the physical domain, and some discretization procedure is used to represent approximate solutions and differential operators on the mesh. In addition, the concepts of adaptive mesh refinement for local resolution enhancement, time-varying meshes to represent moving geometry, data transfer between different meshes, and parallel decomposition of the mesh for computation on advanced computers are the same regardless of their implementation. In each case, the software tools providing these advanced capabilities are becoming increasingly accepted by the scientific community, but their application interfaces are not compatible. Thus interchanging technology is often a labor intensive and error prone code modification process that must be endured by the application scientist. This typically results in a lengthy diversion from the central scientific investigation and severely inhibits experimentation with improved mesh and discretization technologies.

The recognition of this fundamental problem leads to our proposal to develop the technologies needed to create interoperable and interchangeable meshing and discretization software. Our approach to grid software interoperability is both revolutionary and evolutionary. We will formulate a broad, comprehensive design that encompasses many aspects of the meshing and discretization process (the revolutionary part), but will work toward that goal through incremental insertions of existing and newly developed technologies into our targeted applications (the evolutionary part). We will focus our efforts in three primary areas: 1) advanced meshing technologies, 2) high order discretization techniques, and 3) terascale computing issues. In advanced meshing technologies, our emphasis is the creation of common interfaces for existing TSTT Center technologies that will allow them to interoperate with each other to provide fundamentally increased capabilities and to allow application scientists to easily switch among them. We will develop new capabilities as needed within these tools to provide compatible functionality and to support complex geometries, high order discretization techniques, and adaptive methods. In the area of high order discretization, we will use our extensive experience with a number of different discrete operators and high-level interface definitions to create a Discretization Library. This library will support commonly used operators and boundary conditions, will be extensible to provide application specific customization, and will be independent of the underlying mesh type and therefore interoperable with all TSTT meshing technology. Finally, we will develop the algorithms necessary for efficient performance on terascale architectures. Our focus will be on enhancement of parallel mesh generation capabilities, dynamic partitioning strategies for hybrid, adaptive computations, and the use of preprocessing tools to achieve optimized single processor performance.

In all cases, we will encapsulate our research into software components with well-defined interfaces that enable different mesh types, discretization strategies, and adaptive techniques to interoperate in a "plug and play" fashion. The interface design will be driven by application scientists' requirements and the need for intuitive, easy to use interfaces at multiple levels of sophistication. Thus, we will provide both high-level abstractions (e.g. representations of an entire complex mesh structure, and operations on that mesh) appropriate for new application development and low-level access functions (e.g. approximations of derivatives at a single point on a mesh) appropriate for incremental insertion of new technologies into existing applications. To ensure the relevance of our research and software developments to the SciDAC goals, we will collaborate closely with both SciDAC application researchers and other ISIC centers. In particular, we will insert existing TSTT technologies into fusion, accelerator design, climate modeling, and chemically reacting flow applications, and thereby have both significant near-term impact on those efforts as well as receive the feedback necessary to ensure our success. By working with a large number of applications, rather than just one, we will be able to abstract the user requirements with a greater level of generality to improve the likelihood of providing common interfaces that are widely applicable.

The time is ripe for such an effort in the area of meshing and discretization technology. The existence of mature software tools in these fields allows us to examine them, and their use in applications, to extract the appropriate abstractions for developing common interfaces at both high and low levels. We note that such multi-institutional activities are underway in other scientific domains. For example, the Equation Solver Interface (ESI) group [ESI] is defining a set of common interfaces for linear solvers and preconditioners and is implementing them in a number of independently developed tools including, e.g., PETSc, ISIS++, and Aztec. In a related effort, the Finite Element Interface (FEI) group is developing a high level abstraction and interface definition between (implicit) FE/FV codes and linear and nonlinear solver packages [FEI]. The Common Component Architecture (CCA) Forum [CCA] has developed a standard interface for the interactions between software components and frameworks that promotes “plug and play” experimentation. Moreover, this group has recently initiated an effort to define common interfaces for scientific data, including meshes and fields; collaboration with them in this area is an integral part of the proposed TSTT center. However, the the scope of the CCA effort is limited to providing interfaces to existing mesh technology. Our proposed work includes such interface definition development and goes beyond it, providing the additional functionality needed to create a truly interoperable environment for meshing and discretization. We are well qualified to perform this research because of our broad areas of expertise in meshing and discretization, our experience with the development interoperable software within PDE solution frameworks, and our strong commitment to applications.

2 Related Previous Work of the Investigators

We have a strong team of DOE and university researchers whose areas of expertise span mesh generation and quality control, discretization, adaptive procedures for *h*- (mesh resolution), *p*- (method order), and *r*-refinement (node point movement), mesh motion, front tracking techniques, static and dynamic partitioning strategies, and parallel computing. Our previous work in these areas displays a strong balance of basic research, software development and deployment, and collaboration with application scientists.

Mesh Generation and Quality Control. The first step in solving many PDE-based applications is the generation of a mesh that provides a discrete representation of the computational domain. There are a wide variety of mesh types commonly used ranging from logically regular, structured meshes to unstructured tetrahedral, hexahedral or mixed element meshes. The TSTT center PIs are experts in each of these areas of mesh generation and have created advanced algorithms and techniques that address the challenges associated with complex geometry, high order discretizations, and effective mesh quality assessment and control.

Structured Mesh Generation. We have pursued two primary development efforts in the area of structured mesh generation. First, Brown, Henshaw, and Quinlan at LLNL have developed algorithms to create high-quality, predominantly structured meshes on complex, CAD-defined geometries as part of the Overture project [BrCh97, He98a, He98b]. The geometry is decomposed into simpler parts and a structured grid is built for each part. The parts are then combined with either an overlapping grid (where the parts communicate through interpolation) or a hybrid grid (where the parts are joined by unstructured grid elements using an advancing front algorithm) [He98a]. A variety of techniques for building structured component grids are supported including those based on solving hyperbolic, elliptic, and algebraic equations. Second, basic research by Khamayseh, de Almeida and D'Azevedo at ORNL and by Knupp at SNL has improved many of the algorithms used to generate structured grids. For example, Khamayseh has improved elliptic structured grid generators to better enforce orthogonality and satisfy certain geometric constraints [KhaMas94, KhaHam96, KahHan00, Kha98]. De Almeida has successfully applied hyperelasticity tools for the analysis and practical construction of domain deformation mappings in variational mesh generation [deA99] and has solved the difficult problem of computing a bijective nonsingular mapping that satisfies the invertibility condition using solution continuation and nonlinear finite element techniques [deADer00, deAYec98]. Knupp has authored numerous articles on variational grid generation [KnR00] and a textbook on structured grid generation [Kn93]. To improve the quality of regular two-dimensional meshes, D'Azevedo created algorithms based on coordinate transformations for low order interpolation [DAz91, DAz99, DAz00] and, in some cases, achieved superconvergent results with his transformed bilinear quadrilaterals.

Unstructured Mesh Generation. Several of the TSTT center PIs have developed sophisticated algorithms and software for the generation and quality control of tetrahedral, hexahedral, and mixed element, hybrid meshes in the MEGA (RPI), CUBIT (SNL), and NWGrid (PNNL) projects. These tools all support the automatic (or nearly automatic) generation of unstructured meshes on complex geometries, but they differ in the element types generated, the particular algorithms used, and the advanced functionality supported. For example, the MEGA toolkit (Shephard) [Shephar00] supports isotropic meshing and mesh modification [deCShe99, Shephar00], boundary layer mesh generation [GarShe00], and the generation of curved mesh entities for higher-order discretization methods [DeyOba99]. MEGA focuses primarily on the generation of tetrahedral meshes directly from solid models using combinations of Delaunay and advancing front techniques as well as octree decompositions. In contrast, the CUBIT project (Tautges, Knupp, Leland) [CUBIT] has developed a toolkit of new algorithms (e.g., paving [TB90], weaving [Fol99], hex/tet [Mey98], and sweeping [Kn99,Kn98]) for hexahedral mesh generation on complex geometric assemblies [Tau01b]. These algorithms are interfaced to geometry using the Common Geometry Module [Tau01], allowing access to geometry in multiple formats, including alternative representations like facets and virtual geometry [Kra00]. CUBIT has been used to generate all-hexahedral meshes containing between 10 to 100 million elements and also supports tetrahedral and some hybrid meshes. NWGrid (Trease) [Tre00a,Jon00] is used to construct hybrid meshes by using a combination of Delaunay, AMR, and block structured mesh generation algorithms [Tre00a, Jon00]. To improve the quality of unstructured simplicial and quad/hex meshes, Knupp (SNL) and Freitag (ANL) have developed optimization-based, local mesh smoothing algorithms that effectively improve both the extremal and average element quality [FreKnu99,FreOll97,FreOll00,FrePla00,Knupp00]. In addition, Knupp has developed new algebraic mesh quality metrics that provide a unifying theory for assessing and improving element quality [Knupp01].

Frameworks for the solution of PDEs. Each of these mesh generation packages has been interfaced to one or more frameworks for the solution of PDEs. Particular examples of such frameworks developed by TSTT PIs include Overture (Brown, Henshaw, and Quinlan) [BrCh97,He98c], Trellis (Shephard and Flaherty) [BeaShe97,BeaShe99], NWPhys (Trease and others) [Tre96, Jon00, Nie01], and FronTier (Glimm and others) [GliGro99,GliGro99a]. These frameworks utilize advanced computer science concepts such as object oriented programming techniques (Overture, Trellis, NWPhys, FronTier), generic programming methods (Trellis), and performance optimization through preprocessing source code translators (Overture). In addition, Fischer (ANL) has developed nek5000, a high-performance, distributed memory framework for the solution of computational fluid dynamics applications using spectral elements [TufFis99]. These frameworks all aim to significantly ease the burden of new application development on distributed memory, parallel architectures by providing high level interfaces to sophisticated functionalities such as high-order discretization schemes and adaptive methods.

High-Order Discretization Techniques. The high-order discretization schemes supported by the TSTT center frameworks include finite element (Trellis) [DeyShe97, SheDey97], finite volume (NWPhys [Tre00b], Trellis, Overture[Br99]), finite difference (NWPhys [Tre00b], Overture [He98c]), discontinuous Galerkin (Trellis) [BisDev94, RemFla00], spectral elements (nek5000) [TufFis99, Fis97], and partition of unity methods (Trellis) [KlaShe00]. To illustrate the development of high level interfaces for these discretization methods we highlight two of the frameworks: Overture and Trellis. Overture contains a set of grid-function classes that represent field variables on grids and operator classes that implement a variety of finite-difference and finite-volume operators (2^{nd} order, 4^{th} order, conservative/non-conservative) on curvilinear grids [He98c, Br99]. The operators can be manipulated using high level semantics permitting the specification of complicated differential operators. An extensive selection of boundary conditions is also supported. Trellis separates the weak forms, shape functions, Jacobian and integration functions into object libraries so that new components can be added that directly use other components as desired [BeaShe99]. This allows application scientists to quickly add new models that use the existing classes to provide shape function data and perform the various differentiation, mapping and integration functions; advanced developers can easily add customized sub-classes to optimize efficiency. Trellis also employs a high-level, geometry-based problem specification to support the needs of high-order discretization methods and a full set of adaptive methods. These strategies have proven successful in their respective tools and will serve as the foundation of our proposed Discretization Library.

Adaptive Procedures. To increase the accessibility of adaptive methods to application scientists, *h*-, *p*-, and *r*-refinement methods are provided in the TSTT center frameworks. In particular, the Overture, Trellis, and NWPhys frameworks all support *h*-refinement. Overture provides an extension of the Berger-Collela block structured *h*-refinement methods to overlapping grids [Bri95] and includes support for grid interpolation functions, error estimation, and solution algorithms [Qu98,BrHe00]. The Trellis and NWPhys frameworks support local refinements of various types on an element-by-element basis. Trellis also supports *p*-adaptivity for both continuous and discontinuous basis functions [DeyShe97,RemFla00]. The *h*- and *p*-adaptive methods in Trellis have been used to construct *hp*-adaptive procedures that are linked to the underlying geometry so that refined elements remain true to it. Furthermore, in the area of *r*-refinement Knupp has created an optimization-based approach for Arbitrary Lagrange-Eulerian (ALE) methods that preserves mesh quality while maintaining fidelity to the underlying flow field [KnMaSh01].

The adaptivity process must be guided by error estimators or indicators, and each framework supports a number of these. In particular, Overture supports simple difference and gradient-based error estimators, Trellis supports residual- and projection-based error estimators as well as some simple gradient-based error indicators, and NWPhys supports operator scale length refinement/adaptation criteria [Tre96]. Moreover, Flaherty has developed a number of new error estimation techniques. For example, he proved that the discretization errors of one-dimensional discontinuous Galerkin methods exhibit superconvergence at the roots of Radau polynomials [AdjDev00]. These results were applied to create a very efficient *a posteriori* error estimation technique for linear and nonlinear hyperbolic conservation laws [AdjDev00, BisDev94]. He also constructed simple and accurate *a posteriori* error estimates of reaction diffusion applications by using an odd-even dichotomy discovered by Yu. The results were proven to be asymptotically correct [AdjFla99] and extended to a variety of finite element bases [AdjBel01].

An additional adaptive strategy associated with our center is FronTier, a front tracking code developed by Glimm and others at Stony Brook, Los Alamos National Laboratory, and Brookhaven National Laboratory. It has unique capabilities to eliminate mass diffusion across fluid interfaces, and the ability to track dynamic solution discontinuities of various types [GliGra98,GliGro99,GliGro99a]. FronTier has provided high quality simulations for a number of difficult fluid mixing problems. It was the first code to achieve correct simulations for the single mode impulsive mixing (Richtmyer-Meshkov) problem [GroHol94,HolFry99]. Recently, it has achieved the unique accomplishment of agreement with laboratory experiment for multimode steady acceleration (Rayliegh-Taylor) fluid mixing rates [GliGro00,CheGli00], and it has been used to model laser driven models of supernova instabilities [DraRob00].

Terascale Computing. All of the TSTT center frameworks operate on large-scale parallel computers, and the PIs have considerable experience with parallel mesh generation, static and dynamic load balancing algorithms, and performance optimization. For example, the MEGA mesh generation tool has been used to create meshes containing over 10 million elements on a 32 processor IBM SP [deCShe99, deCShe99a]. The parallel mesh generator employs an octree structure and dynamic load balancing which ensures the scalability of the algorithm and controls, without adverse effects on mesh quality, the interprocessor mesh interfaces. In addition, NWGrid has been run on a variety of computing systems ranging from one PC workstation to 6000+ processors of a multi-teraflop supercomputer. For the ASCI program, NWGrid has routinely been used to generate 10 to 100 million element unstructured, hybrid meshes and on occasion has been used to generate 100 to 900 million element AMR meshes. To build its meshes in parallel, NWGrid distributes the domain geometry to the processors and applies the mesh generation algorithms to each piece in such a way that a single, coherent mesh results.

Achieving scalable performance on terascale systems depends critically on the distribution of the mesh across the processors of the parallel computer. Typically, the partitioning strategy strives to create balanced processor workloads while simultaneously minimizing communication costs. This process is further complicated for dynamic, adaptive computations because the workload changes as the simulation proceeds. In this case, an initially balanced partitioning is unlikely to remain so, and must be updated periodically. Several TSTT PIs have worked in the areas of static and dynamic load balancing. The problem can be posed differently for structured and unstructured meshes; often graph-based algorithms are used for unstructured meshes while either graph based or geometric methods may be appropriate for structured meshes. On unstructured meshes, for example, Leland's (SNL) contributions to the static partitioning problem were released in the popular Chaco library [HenLel94, HenLel95a, HenLel95b]. Work on dynamic load balancing is currently underway in Flaherty's RPM and the Sandia Zoltan [DevHen99a, DevHen99b] libraries. The Rensselaer Partition Model (RPM) [FlaTer00, TerBea00] uses the communication and memory properties of the actual computational nodes in a model that can be used to tailor task scheduling to a particular, possibly heterogeneous, architecture. Entity weightings have been used to successfully partition adaptive time marching algorithms [FlaLoy97a] and to create a predictive load balancing procedure where the weights were based on refinement level and the mesh rebalanced before the refinement was performed [DinShe00, FlaLoy99]. Several of the RPM dynamic load balancing procedures [FlaLoy99, FlaLoy97a] and the popular METIS suite have been incorporated into the Zoltan library [DevHen99a, DevHen99b] which provides a common interface to these independently developed partitioning tools. With proper care as to provide distributions consistent with the distribution support of structured grids (e.g. rectangular in 2D), structured grid load balancing can use many of the same or similar graph based partitioning tools as are used for unstructured grids. But geometric load balancers are perceived to be much faster (though often less able to address non-uniform communication costs). The Multilevel Load Balancer [Qu97] was developed for use in Overture. As a parallel load balancer it addresses the specific needs of adaptive overlapping mesh computations. This work is part of the parallel distribution support provided in PADRE [Qu98b]. Collectively the individual PIs have substantial experience with the distribution and load balancing of applications using our respective parallel frameworks and are in excellent position to pool collective work as part of this proposal.

The interoperable software strategies we propose to develop will entail the use of advanced computer science concepts which unfortunately often perform far below advertised peak rates for single processors. To address these issues for object-oriented C++ scientific codes, Quinlan has developed the ROSE programmable source-to-source transformation tool [Qu01]. ROSE represents a mechanism to build preprocessors that read the application source code and output highly optimized C++ code that has been transformed using the semantics of the object-oriented abstractions represented within the framework. Quinlan has demonstrated that, by using the ROSE approach, full Fortran performance can be recovered for discretization stencil operations typical of computations involving complex geometry. For example, speed-up for a typical nine-point stencil operation on an array with 10,000 elements has been measured at 5.6 times improvement from the original binary overloaded operator implementation of the array class. We will use these techniques to ensure that critical kernel operations, such as those found in the Discretization Library, obtain optimal performance on terascale architectures.

Preliminary Studies in Interoperability. The TSTT center PIs have considerable experience with the development of interoperable software and have done preliminary work in many of the proposed areas of research. In the area of common interface definition there have been three primary efforts. First, Shephard (MEGA) and Tautges (CUBIT) have created generalized interfaces that allow their mesh generation tools to interact with several solid modeling tools [BeaShe97, RemKar00, Shephar00, DeyOba99, Tau01]. Second, Freitag has been active member in the CCA Forum and is coordinating the effort to define scientific data interfaces. She has done preliminary work in this area through the creation of general interfaces for mesh interactions in an explicit, discontinuous Galerkin fluid mixing problem [FreGro99]. Finally, as already mentioned above, the Zoltan dynamic partitioning library contains common interfaces to several independently developed software packages. In the area of interoperability to support multiphysics applications, Trellis has been linked to commercial finite element packages and externally developed finite difference computational fluid dynamics codes, and within a single framework, FronTier has support for front tracking in compressible gas dynamics, oil reservoir simulations, deformation and material strength of solids, manufacture of composites, and deposition and etching.

Software Deployment via Applications Research. We are committed to working with application groups to improve numerical simulations by deploying our tools and framework software to address challenging scientific problems. For example, Shephard and Flaherty have collaborated on transient flow problems including rotorcraft aerodynamics [DinShe00], crystal growth [GivFla97], bifurcated shock tubes [FlaLoy97a], chemical vapor deposition [AdjFla99a], and fabrication processes of bio-artificial tissue [OhsFla00]. Trease has used the predecessor to NWGrid, X3D (developed while Trease was at LANL) in a variety of areas including subsurface transport for Yucca Mountain [Gab95, Gab96], semiconductor modeling [Car96, Kup96, Tre96] and oil and gas simulations [Zyv96]. FronTier has been used to study jet instabilities in design of a target for a proposed muon collider [GliKir00], jet breakup and spray formation [SchRut99,SchRut99a], and instabilities of solid-solid interfaces [WalYu00,WalGli99a]. In addition, the BNL team has worked with accelerator design physicists to add and improve parallelization for particle tracking PIC accelerator design codes [LucDIm00,GalLuc00,LucDIm00a]. They also studied the grid dependent errors associated with upscaling or grid dependent parameterization for petroleum reservoir simulations

[GliHou00,GliHou00a,GliHou01]. CUBIT has been used to create large assembly meshes for dynamic structural response, structural vibration analyses, Z-Pinch and wire implosion design studies, rolling and casting of materials, and thermal response calculations. In addition, CUBIT has been used by commercial companies such as Caterpillar and Goodyear for structural analyses of heavy machinery and tire tread design. Simulation codes built with the Overture framework have recently been used to study pattern formation in non-Newtonian Hele-Shaw flow [Fa99, FaSh99, FaKo01], viscous flow around deforming bodies [FaHe01], the dynamics of tidal flow and sail performance [Ge98], the mechanisms of biological cell fertilization [Bu99, Cs99, Ke98, Wa99], and the chemistry and dynamics of plant root growth [Ki99].

3 Research Design and Methods

Our research will focus on three major areas: advanced mesh technologies, high-order discretization methods, and the related terascale computing issues. The following sections describe the research design and procedures we will use to accomplish our goal of creating interoperable and interchangeable meshing and discretization technologies. Since our research is closely tied to interactions with SciDAC applications projects, our proposed research timetable is included in Section 7 following the discussion of our interactions with applications and the other ISIC centers.

3.1 Advanced Mesh Technologies

Delivering significant capabilities associated with advanced mesh technologies to current and future Office of Science applications is central to the goals of the TSTT center. In the following sections, we discuss the design and development of interoperable mesh technology and tools. We also discuss the improvements to automated mesh generation and mesh adaptivity that are required to support the delivery of these advanced tools.

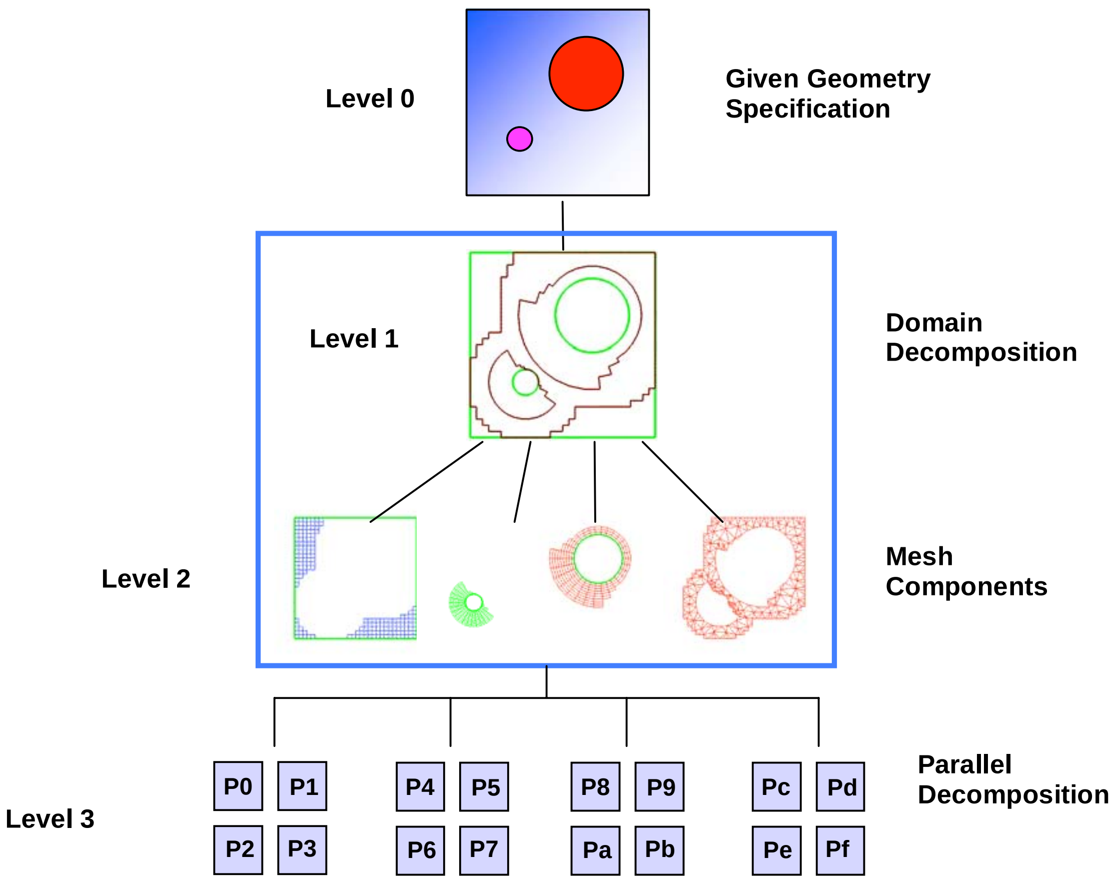

Figure 1. Conceptual geometric hierarchy.

3.1.1 Interoperable Mesh Technology

Meshing and discretization technologies must take into account a conceptual hierarchy of domain representations (see Figure 1). The original problem specification can be given in terms of a high-level geometric representation such as a CAD description, image data, or possibly a previous mesh, along with the physical attributes of the application (level 0). As an example, the electromagnetic simulations for accelerator design are based on a detailed and complex specification through CAD-generated geometries, while for other applications such as plasma physics, typical input geometries are relatively simple. This geometry has a global representation of some kind (such as a non-manifold solid model or pixel intensity information) that properly represents the physical domain. The geometric model can be decomposed into sub-regions (level 1), each of which could be discretized using a possibly different fundamental mesh type. Decompositions for the purpose of adaptive mesh refinement must also refer back to the original geometrical information, and hence are part of this level. The next level (2) consists of the meshes that are constructed within the components of the level 1 decomposition. Because the simulations will be run on terascale parallel computers, a further decomposition (level 3) to support the partitioning of the levels 0-2 entities across the 1000's of processors involved is defined (see the Terascale Computing Section). All the levels shown in Figure 1 can be considered different resolutions of the domain, and fit into a hierarchy of representations. Meshing and other tools (e.g. for partitioning, discretization, or solution interpolation) can access these representations through a common interface, simplifying the process of changing the resolution either inside a level or between levels. The use of this concept will be explored using the various representations available in the Common Geometry Module (CGM) [Tau01].

Analogous to the conceptual geometry hierarchy discussed in the previous paragraph, we also focus on a mesh data hierarchy (Figure 2). The highest level in this hierarchy (level A) again contains a description of the geometric domain. Access to this information is provided, for example, by generic interfaces such as CGM, direct functional interfaces to solid modeling kernels, e.g. [SheGeo92], or data file representations such as IGES and STEP. The lowest level (C) shown in the figure again represents

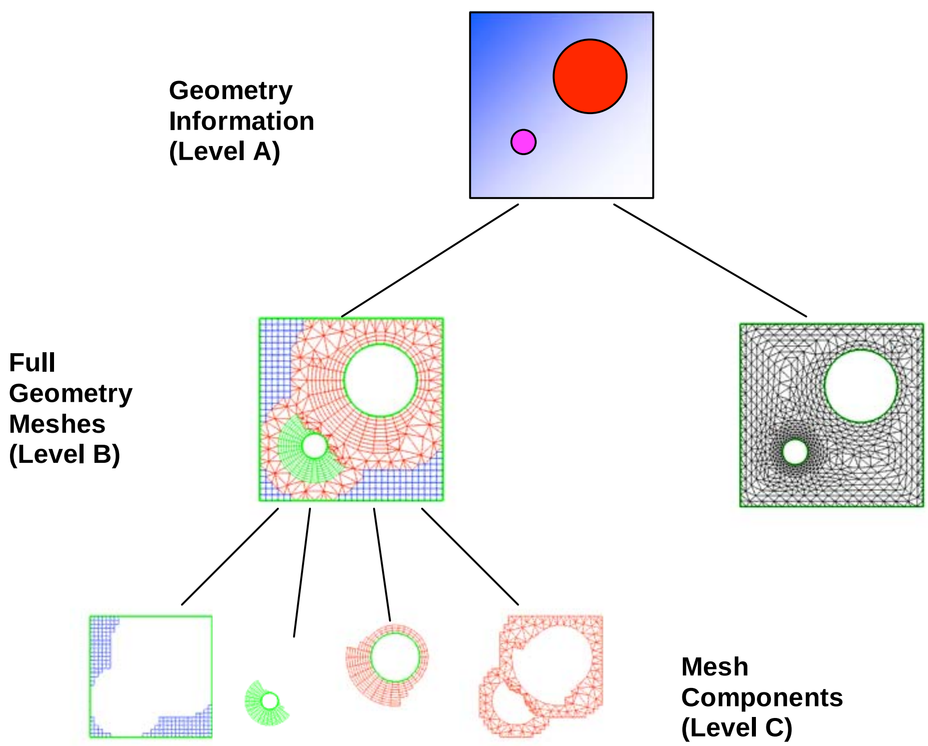

Figure 2. Mesh Data Hierarchy

the mesh components, while the intermediate level (B) represents the combination of the level C mesh components into an overall hybrid mesh, together with the communication mechanisms that couple their data together. The structured/unstructured meshes required by the Stanford Linear Accelerator Center's (SLAC) time-domain accelerator design code, *Tau3P*, are examples of such a hybrid mesh. In addition, level B allows the possibility of multiple mesh representations for the complete geometry to be used within a single application, or in coupled applications. At level B, code developers or users can access the entire grid hierarchy as a single object, and call functions that provide, e.g. partial differential operator discretizations, adaptive mesh refinement or multilevel data transfer over the entire mesh. This will be accomplished by providing an interface much like that currently provided in the Overture framework [BrCh97, He98b, He98c]. Functionality at this level will enable the rapid development of new mesh-based applications. In addition to this high-level strategy, we also provide access to the hierarchy at low levels to facilitate the incorporation of our tools into existing applications. At level C, for example, we will provide access to Fortran-callable routines that return discretization or interpolation coefficients at a single mesh point or array of points on a mesh. Initially access at these lower levels is likely to provide for more efficient implementations. However, we expect that by basing the higher-level tools on highly optimized kernels, both code development and runtime efficiencies will ultimately be realized.

Construction of this hierarchical description of a computational geometry will be accomplished by leveraging existing TSTT center tools for mesh construction. In particular, we will use graphically-based tools available in Overture [He98a] and in CGM [Tau01] to partition a complex geometry into level C subregions that can be meshed using TSTT mesh generators. A key new area of research will be the coupling of the level C sub-meshes to provide a complete discretization of the computational geometry at level B, which will be accomplished by development of tools similar to the automatic overlap and hybrid mesh-stitching algorithms currently provided by Overture [He98a]. Consideration will be given to converting powerful, but reasonably heavy, mesh data structures [BeaShe97] to dynamic structures that automatically configure themselves at run time [RemKar00]. The goal is to enable construction the specific mesh entity adjacencies needed at run time, thus providing the application the most efficient means to obtain the needed mesh related information with the minimum amount of storage. The mesh structure and data hierarchy must also facilitate the needs of the discretization and solver procedures. For example discretization procedures will employ information from levels 1 and 2 (Figure 1) to control the transfer of information. These rules are dramatically different between overlapping meshes, conforming meshes and patches of non-conforming meshes. Similarly, information on these hierarchies is needed for the most effective solution of the resulting systems. We plan to work closely with the Terascale Optimal PDE Solver (TOPS) ISIC to employ information at levels 1-3 of the hierarchy for parallel multilevel solution procedures.

To accomplish our interoperable software goal, common interfaces will be defined for accessing data at each level of the hierarchy shown in Figure 2. Initially we will focus our efforts on creating the interfaces necessary for accessing static mesh geometry and topology at level C as well as the associated application, or field, data. These low level interfaces will account for the differences in how mesh data is typically stored for structured, unstructured, and hybrid meshes. For example, structured meshes can be represented compactly with node point locations and an indexing scheme and are typically accessed using multi-dimensional arrays whereas unstructured meshes require explicit representation of the connectivity information and may be accessed using list structures. We will therefore group our interface definitions into broad categories appropriate for each mesh type; they will be general enough to support a broad range of tools, but specific enough to support high-performance, efficient implementations. To further increase performance, we will explore various caching and agglomeration strategies to minimize the overhead costs associated with using generalized interfaces.

To ensure the appropriateness of our interfaces, we will work closely with the developers of the Discretization Library and researchers in the TOPS ISIC. In addition, we will collaborate with the CCA Forum and their proposed Center for Component Technology for Terascale Simulation Software to accomplish two goals. First, we will be involved in the efforts to develop standard interfaces for scientific data components, including meshes and data fields. By cooperating with each other in this effort, a larger group of tool developers will provide input to, and eventually adopt, the proposed common interfaces that in turn will promote broader community usage and acceptance. Second, in addition to modifying TSTT technologies to comply with the interfaces just described, we will also modify them to become CCA-compliant by incorporating the component interface specifications for interacting with frameworks. Thus TSTT tetrahedral, hexahedral, hybrid, structured, and overlapping grids will be provided as CCA numerical components.

Once this initial work is complete, we will develop interfaces to support access to the entire grid hierarchy shown in Figure 2. This will include support for advanced geometric capabilities such as generalized geometry-based tensor attributes with general distributions and dependencies. This work will support our efforts to create truly interoperable meshing capabilities by allowing each mesh generation tool to have a common view of the geometry and physical attributes associated with the level A physical domain. This will be followed by defining the common interfaces needed to support various forms of adaptation. This includes accessing error estimators and directionally dependent refinement indicators, accounting for high-order curved element mesh entities, and adaptive refinement that remains true to the geometric definition.

3.1.2 Enhancement of Mesh Generation Technology

The TSTT sites have already developed significant mesh generation capability starting from geometric descriptions based on various representations. Unstructured mesh generation capabilities are provided by MEGA (RPI) [Shephar00], CUBIT (SNL) [CUBIT] and NWGrid (PNNL) [Tre00a, Jon00]; structured overlapping and hybrid mesh capabilities are provided by Overture (LLNL) [He 98a]. In order to achieve true interoperability among the different mesh types, it is important that the mesh generation tools all operate on similar infrastructure and provide similar capabilities, and we have identified three areas that require improvement (described below). In addition, we will provide parallel mesh generation support as described in the *Terascale Computing* section of this proposal.

First, it is important that we significantly reduce the amount of time required to generate all meshes from a common domain description. In particular, hyperbolic mesh generation algorithms for hybrid or overlapping grid generation from CAD data needs significant improvement. Enhancement of structured mesh quality will be accomplished by developing automated procedures for optimizing hyperbolic mesh generation parameters and by improving variational mesh optimization techniques. We will also continue to develop techniques for robust generation of hybrid meshes using stitching algorithms in which large regions of smooth structured mesh are connected with regions of high-quality unstructured mesh [He98a] and hybrid mesh generation approaches based on advancing fronts [Mey98, Mit98].

Second, many of our tools will be enhanced to support the high-order discretization techniques that will be provided as part of the Discretization Library; particularly in the case of complex geometries. Preserving the convergence rate for high-order discretizations over curved domains requires that the mesh elements themselves are curvilinear [DeyShe97]. Two functionalities needed to curve a straight-edge mesh are the geometric representation of curved mesh entities [SheOba00] and the ability to modify the mesh to maintain mesh quality as the entities are curved to represent the portion of the boundary they cover [DeyOba99]. Alternative methods to define the element geometry for use in the discretization process have been examined [DeyShe97] and methods using approximating polynomials appear to be more effective than standard Lagrange interpolants [SheOba00]. Research must be performed to determine the best ways to approximate geometry and to create the proper order curved mesh entities. Efforts in this area will specifically consider the use of high-order discretization techniques in the fusion application (see Section 4.1).

Third, to ensure good quality hybrid meshes, we will develop stand alone mesh improvement tools that will work on all TSTT mesh types . Mesh quality and smoothness is critical for both the accuracy and efficiency of low- and high-order discretization techniques (see, e.g., [FreOll97]). For example, the solvers used in electromagnetics simulations for modeling linear accelerators (Section 4.2) and the hemodynamics simulations for studying vascular disease (Section 4.5) can be significantly impacted, sometimes failing to converge, unless mesh quality has been optimized [Fol01]. Although many algorithms exist to improve mesh quality (see, e.g., [FreKnu99, Joe95, FreOll97]), there currently exists no stand alone mesh improvement component that incorporates a broad spectrum of state-of-the-art approaches and that supports all of the TSTT mesh types. Such a software component will be critical to the success of the TSTT center for several reasons. First, we anticipate that mesh quality improvement algorithms will be needed for the transition regions in conforming hybrid meshes obtained from multiple meshing codes. This is especially true for high-order methods which employ relatively few macro-elements; the conditioning of the governing systems can be impacted significantly by the deformation of just a few elements. Second, both *h*-adaptivity and ALE *r*-adaptivity can significantly degrade mesh quality, which can be mitigated through the use of local mesh quality improvement algorithms. Research areas that will be pursued in this area include (i) *a priori* quality metrics appropriate for high-order discretization techniques and general polyhedral elements, (ii) *a posteriori* methods that use error indicators (see Section 3.2.1) to drive optimization-based smoothing and flipping, and (iii) improvement algorithms for mixed/hybrid structured and unstructured meshes. All algorithms and software will use the low level common interfaces for accessing mesh geometry and topology described in Section 3.1.1; hence, this stand alone tool will be compatible with all of the existing and proposed TSTT mesh generation tools.

3.1.3 Dynamic mesh evolution

In many applications, the mesh description of geometry evolves during a computation. Reasons for mesh evolution include adaptive mesh refinement/coarsening , internally tracked interfaces or fronts due to the simulation physics, or the motion of domain boundaries either in a prescribed fashion or in response to the simulation physics. These possibilities will be accommodated by the mesh hierarchy through the evolution of the hierarchy structure and the components at each level during a computation. Interfaces for communicating algorithmic or physical information to the underlying mesh generator and properly accessing and using the evolving mesh within an application simulation will be provided.

A significant amount of basic research will be done to provide interoperable adaptive techniques within the context of the mesh hierarchy described in Section 3.1.1. We consider adaptive procedures that account for mesh modifications locally (e.g., on an element-by-element basis), in selected subdomains, or in the entire domain employing information from the mesh hierarchy (see Figures 1 and 2). To accomplish the TSTT goals, we will develop adaptive techniques that address the following issues (described in more detail below):

- true geometry of the domain to preserve the convergence rates of high order discretization techniques,
- effective abstraction of adaptive analysis procedures to promote “plug and play” interoperability ,
- enhanced front tracking algorithms that are combined with hybrid mesh generation schemes to significantly improve accuracy,
- mesh modification procedures that account for multiple criteria to extend their applicability,
- mesh adaptation for hexahedral and mixed element meshes to provide equivalent capabilities among TSTT center tools, and
- automatic selection and application of optimal adaptive strategies to improve ease of use.

Accounting for the domain geometry during adaptive processes includes operations to ensure that refinements faithfully represent the original underlying geometry. In the case of straight edged geometry this includes new mesh vertices, while in the case of curved mesh entities it can include mesh vertices, edges, and faces. The process of conforming mesh entities to the boundary can invalidate elements or degrade them to the point of being unacceptable. In such cases it is necessary to apply mesh modification operators to problem elements [LiShe01]. Procedures to perform such modifications for the full set of mesh and partition types supported by the TSTT center will be developed.

Most current adaptive procedures employ a tightly coupled error estimation and mesh adaptation strategy. To support the more effective insertion of various mesh adaptation procedures into alternative simulation procedures we will develop generalized mechanisms that accept error estimation and/or correction indication information as input and that determine the specific mesh adaptations to perform based on the available AMR procedures. . These procedures will build on knowledge of the theoretical properties of the error, its control, and the relative computational cost of alternative strategies. Adaptive procedures for the accelerator, chemically reacting flows and biology applications, including those accounting for special constraints, such as grid alignment for the fusion application, will be considered early in the program.

Many classes of simulation are characterized by the formation of sharp fronts as the simulation evolves. In these cases, the most effective adaptive procedures must include a front tracking technique [GliGra98], [GliGro99] to accurately resolve the front. Exact conservation and higher order accuracy at the front are difficulties with present algorithms. Our studies indicate a significant improvement will result from addressing these two issues, for which purpose we will introduce a front adaptive, space-time discretization (instead of the more common space time semi-discretization). Three-dimensional meshes that are “swept” through time (including possible “collapses” and “splits” associated with dynamic bifurcation of the front topology) are required. Implementation of such space-time, front- adaptive meshes will be enabled by developing interoperability between our front-tracking code, FronTier, and the hybrid TSTT structured and unstructured mesh generation codes.

Interfaces developed to control adaptive simulation processes must consider cases in which multiple criteria drive adaptive mesh modifications. An obvious example where this is critical is in evolving geometry simulations based on Lagrangian meshes where mesh modifications are needed to maintain element shape control and mesh refinements are needed to control the discretization errors. One initial effort will be to consider the definition of the best refinement strategy for an element that is marked for refinement and also has one or more large dihedral angles needing reduction [SheWan01]. We will develop generalized procedures that employ a full set of mesh modification operators (split, collapse and swaps) that account for both discretization error and shape improvement criteria.

Although effective mesh adaptation methods are known for unstructured tetrahedral meshes and overlays of structured grid sub-domains, general procedures are not available for unstructured hexahedral and mixed element meshes. For these mesh types, efforts will be taken to develop and compare adaptive refinement (both conforming and non-conforming) procedures and remeshing techniques using error size control maps.

3.2 High-Order Discretization

The complexities of discretizing new applications on unstructured and adaptively evolving grids have hampered widespread usage of many powerful tools, leading to suboptimal strategies or to lengthy implementation periods. To overcome these limitations and to complement the proposed mesh generation effort, we will develop mechanisms that will greatly simplify the discretization of PDEs on TSTT center meshes. The solution methodology typically involves (possibly discrete versions of) mathematical operations (+, -, *, /, differentiation, div, grad, curl, interpolation, *etc.*) acting on discrete approximations of the field variables relative to the mesh. Such operations are:

- complicated by a tedious low-level dependence upon the mesh geometry (e.g. Jacobians),
- repeated throughout an application,
- a source of subtle bugs during development,
- a bottleneck to the interoperability of applications with different mesh structures,
- difficult to implement in a general way while maintaining optimal performance, and
- often linked with the mesh representation of the domain.

We have an excellent opportunity to simplify the development of application codes by creating a *Discretization Library* of numerous mathematical operators that are common to PDEs arising in many applications. The operators will be applicable to several discretization technologies (as defined below) and the mesh structures addressed within this project. Thus we will dramatically lower the time, cost, and effort to effectively deploy modern discretization tools. As with mesh generation tools, we will develop common interfaces to a wide variety of discretization technologies. User interaction will occur at various levels for use by practitioners wishing to rapidly implement a new application, software developers wishing to add new operators to the library, and applications specialists seeking to incorporate a particular technology into their software. The library will be extensible to allow for user implementation of additional generic and application-specific operations. These might, e.g., include numerical flux functions for use with conservation laws. Problem-dependent functions, such as equations of state, can be evaluated through a library interface. Efficiency is a must and this can be ensured by using an optimization approach such as that offered by ROSE [Qu00] or through the concepts of generic programming [Musser96].

The flow of the entire process from its geometrical considerations through the discretization process has been illustrated in Figure

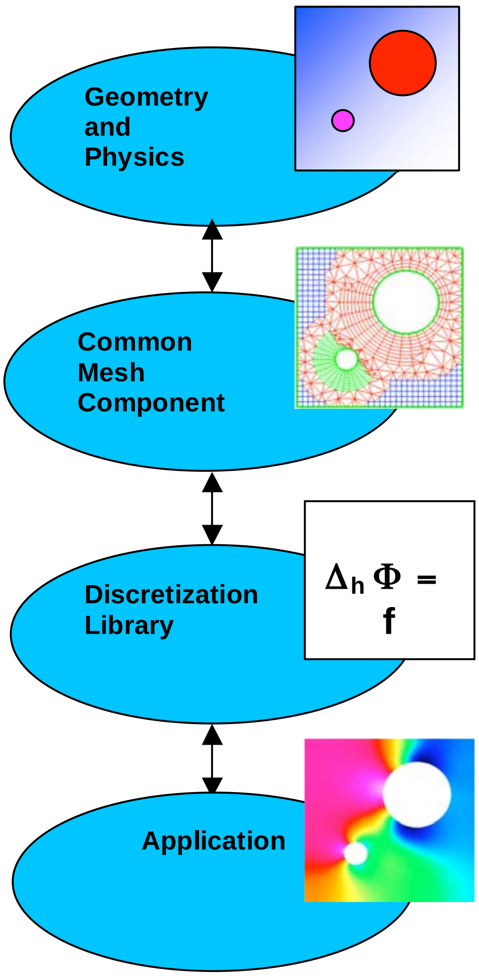

Figure 3. The connection of the Discretization Library to the Common Mesh Component and geometry

3. The mesh creation and adaptive enrichment processes will work in harmony with the Discretization Library to create discrete representations of partial differential systems in a simple and efficient manner.

3.2.1 Development of a Discrete Operator Library

The proposed operator library will provide a representation of discrete operations independent of the underlying mesh and its geometric factors. Such techniques have been developed for structured mesh topology as part of the LLNL Overture project [Hen98c] and for variational discretization (finite element, spectral element, partition of unity, etc.) as part of RPI's Trellis project [BeaShe99]. Thus, collectively we understand the advantages and limitations of this approach through experience with many application domains. Initially we will provide for the inclusion of arithmetic and differentiation operators as well as interpolation and projection operators to support multilevel solution strategies and preserve local conservation. Other common vector (div, grad, curl) and tensor operations will be added to this initial set. The design of the library would:

- permit extensibility so that users can add operators,
- allow access at mid levels (the traditional approach) or at high levels (the simplified approach), and
- permit composition where users build complex expressions by combining operators using, *e.g.*, common array operations.

This structure will allow development of application-specific discrete operators that depend on the PDEs being solved and the algorithms being used to address them. These, could include, *e.g.*, projection operators to handle incompressibility constraints and stabilization operators for convection-dominated viscous flow problems. Boundary operators to handle the myriad of boundary conditions typical of the SciDAC applications would also be included.

At lower levels of interaction the multiple level interface will be of immediate use for existing applications. This interface is simpler to construct, although it is more tedious to use since it exposes many details that will be hidden at higher levels. The higher-level interfaces greatly simplify the construction of new applications but require a much higher degree of infrastructure and optimization.

The Discretization Library will define operators for finite difference (FD), finite volume (FV), finite element (FE), spectral element (SE), discontinuous Galerkin (DG), and partition of unity (PUM) methods. Operators of all orders will be considered, but our emphasis will be on the creation of high-order and variable order methods for use with adaptive methods focusing on varying both the mesh and order (*hp*- or *hpr*-refinement). High-order methods provide rapid convergence (*p*-refinement) and high efficiency in regions where the solution is regular; however, convergence stalls near singularities and *h*-refinement offers a more optimal strategy. The optimal combination of the two (*hp*-refinement) converges at exponential rates and, hence, is remarkably efficient. As part of this project, we will determine optimal adaptive enrichment strategies for different situations and applications. Such optimality is currently available only for simple academic problems. The fusion project considered by the Center for Extended Magnetohydrodynamics Modeling (see Section 4.1), for example, has been using piecewise-linear finite elements on unstructured meshes in poloidal direction and fourth-order finite difference methods in toroidal direction of their reactor. Our discretization library would be able to handle the mixed technologies while giving them the opportunity to increase resolution to fourth order in the poloidal direction.

High-order FE, FV, and SE methods typically associate most of their unknown solution parameters (at nodes) within an element. As such, they simplify communication on parallel computers to produce additional efficiencies. This, however, introduces some complications. Curved boundaries of the regions must be approximated to an order of accuracy that is commensurate with that of the solution. The large number of solution parameters within an element can lead to ill-conditioned algebraic systems. Thus, mesh quality will be a very important consideration. We will work in harmony with the mesh generation group to address these issues.

Support for temporal discretization strategies will range from the commonly used method-of-lines (MOL) formulation, where time steps and temporal methods are spatially independent, to local refinement methods, where time steps and methods vary in space [FlaLoy97a, FlaLoy99], to space-time techniques where unstructured meshes are used in space and time.

The library would include traditional (strong) and variational (weak) forms of operators where applicable. This would permit the combining of methods and operations on mixed grids, to create very powerful new discretization technologies. The plug-and-play versatility of the high-level discretizations within the library would give users the opportunity to experiment with various combinations of discretization technologies and optimize performance for their application. Perhaps of greatest importance, the simplification afforded by hiding the complex grid-dependent details from users would shorten development times, reduce costs, enhance software reuse, and improve reliability of applications implemented and maintained in this manner.

Interfaces will be created that allow software developed partially or totally from operators in the Discretization Library to use the linear and nonlinear algebraic component capabilities developed by the *Terascale Optimal PDE Simulation (TOPS, see Section 5.2)* and the *Integrated Software Infrastructure for Unstructured Mesh Computational Simulation of Transport/Reaction Systems* (see Section 4.3) ISICs. Adaptive approaches using automatic mesh refinement/coarsening (*h*-refinement), variation of method order (*p*-refinement), and mesh motion (*r*-refinement) are best guided by *a posteriori* estimates of discretization errors. While this is an ongoing and active area of research with several unanswered questions and unresolved issues, some error estimates can be provided within the Discretization Library with relatively little difficulty. Error estimates based on *h*-refinement (Richardson's extrapolation) use the difference between solutions computed on different meshes to furnish an estimate. Since the same differential operator is used for each solution and this discrete operator can be composed from tools in the library, it is a simple matter to add this error estimate to the library. Error estimates that make use of solutions computed on the same mesh with methods of different orders (*p*-refinement) are also relatively straightforward to implement. The complexity of the procedure can be reduced by using a hierarchical basis where the solution of order $p + 1$ may be computed as a correction to that of order p. Simpler *ad hoc* error indicators, such as solution gradients and various vorticity metrics, can also be composed from tools within the Discretization Library. The work on mesh quality metrics described in Section 3.1.2 can be combined in a natural way with that on error estimators to provide a comprehensive framework for mesh adaptivity. This framework could enable rigorous control over the geometric properties of a mesh to reduce errors for *h*-, *p*-, and *r*-refinement.

In many classes of simulation it is necessary to transfer the solution fields as the mesh is adapted. In adaptive overlapping grid procedures [BrHe00], this includes the transfer between mesh patches. In other cases, the mesh adaptations are local refinement, coarsening or modifications [deCShe99a, LiShe01]. In each case, generalized procedures must be developed to determine the interacting mesh entities and to perform interpolations meeting such requirements as local conservation properties [Qu98]. The effective support of solution transfer processes when both local modification and independent meshes are combined requires specific consideration. Searching structures and parametric inversion procedures for curved domain problems using both low and high order element geometries must be considered. Particular emphasis must be given to local conservation issues when mapping between meshes [CarBic01, GulFar00]. Early efforts in this area will support the climate application.

3.2.2 Performance Optimization of Discretization Operators

There are two aspects that must be addressed in the performance optimization of discretization operators for terascale computing. The first is obtaining good single processor performance. To accomplish this goal, mechanisms for compile-time optimization of user defined high-level abstractions will be investigated as part of a collaboration with the *High-End Computer System Performance: Science and Engineering* ISIC (see Section 5.3); specifically we will investigate the ROSE preprocessing mechanisms. This work will address the hierarchical memory performance and particularly the cache usage of statements that use the operators defined within the Discretization Library and user-defined operators. The concepts of generic programming where algorithms are separated from particular data structures could certainly be an important factor in implementing distinct discretization strategies on the diverse mesh structures. This concept has produced substantial performance gains when used with the RPM parallel data management system [FlaTer00].

Second, the algorithms for creating and composing the discrete operators and (for implicit operators) assembling them into a global algebraic system must scale well on distributed memory architectures. In most cases, because the discrete operators are local in nature, we will be able to easily accomplish this goal. For example, variational operators are created by traversing geometric entities (e.g., elements) of the mesh. Interprocessor communication is only required for entities on the boundaries of distributed-memory processor partition boundaries. Finite difference operators proceed in much the same way with, perhaps, a more complicated communication due to “ghost cells” at partition boundaries.

3.3 Terascale Computing Issues

The ability to compute on a terascale with trillions of operations per second and exploiting memory capacities of trillions of bytes has been demonstrated on selected problems. However this achievement has required the use of experimental hardware and software systems, great expertise and much labor. The associated challenges have restricted terascale computing to a small fraction of its potential impact on the scientific enterprise. A primary TSTT goal therefore is to address grid related challenges in terascale computing and provide a coherent, usable, scalable and efficient set of tools enabling terascale computing for mainstream scientific applications.

3.3.1 Hierarchical Design

Comprehensive hierarchical design of both data structures and algorithms is the essential challenge in achieving efficient use of computing resources at the terascale. Interoperability alone is not sufficient as tools with this property may nevertheless have computational complexities and memory footprints that render them useless at the terascale. A single failure in this regard anywhere in the tool chain will incapacitate the over-all system. Hence the hierarchical design concepts introduced in section 3.1.1 must be applied throughout the entire software system in use. This has been recognized by the scientific simulation community, and the best-in-class tools in most of the relevant sub-disciplines rely on hierarchical design. Examples include CAD systems that organize data in different layers of resolution, adaptivity in meshing, multi-level graph partitioning schemes for static problem partitioning, multigrid solvers and visualization systems that nimbly present data at many levels of resolution. It is our belief, however, that although these tools are nearly optimized for their particular application, their union is not optimized for the over-all problem of terascale simulation. Our intention is to actively consider various design trade-offs across the whole system. We will not try to design better solver or visualization routines. Rather we will consider what hierarchical information known or easily generated within our scope of geometry and meshing technology can be exploited by downstream tools. For example, information about the geometric domain model and its decomposition may be useful in computing optimal decompositions shown in Figure 1. Similarly, hierarchical information generated previously in the process may be preserved and potentially exploited in preconditioning of iterative solvers..

3.3.2 Parallel Mesh Generation

Even in cases where adaptive mesh refinement methods are to be used, the initial meshes needed for terascale computations will often be so large (e.g., > 100,000,000 elements) that they must be generated in parallel so they are distributed over the entire system form the start. Although this has been an active area of research for several years [deCShe99b], the currently available techniques are not sufficient to meet the needs of an integrated environment. One approach that has been taken to the parallel generation of meshes is to generate a coarse mesh in serial, distribute it to the processors of the parallel system, and refine it without information allowing for the interactions of the distributed mesh with the higher level definition of the domain. In addition to not supporting the general association of analysis attributes, this approach will not support the improvement of the domain approximation as the mesh is refined. We propose a solution in which we ensure that each processor has a copy of at least that portion of the domain definition associated with the portion of the mesh it will maintain. Procedures that distribute the entire geometric model to all processors to support parallel generation of the initial mesh and the geometric interrogation needs of all mesh refinement processes have been developed [deCShe99], [deCShe99a]. Since the geometric model is typically small and its size does not change during an adaptive simulation, this approach will often prove satisfactory. However, it is clear that it does not scale with growth in geometric model complexity, and this may become important limitation in some classes of simulation. In these cases the geometric model will also need to be properly distributed.

3.3.3 Load Balancing

A critical step in parallel computation is the assignment of application data to processors. The ideal distribution assigns data so that per-processor workloads are the same over all processors (eliminating idle time) and inter-processor communication is minimized. For both structured and unstructured grids this distribution is often done through a serial pre-processing step, using a static partitioning tool such as Chaco [HenLel94] or METIS [Kar97].

For adaptive terascale computing, serial static partitioning is insufficient. Dynamic partitioning techniques must be employed from the start of the simulation process. The specific dynamic load balancing algorithms used are often different for structured vs. unstructured grids. Often graph based algorithms are used for unstructured grids while either graph based or geometric methods may be appropriate for structured grids. Our proposal includes parallel mesh generation steps [deCShe99], constructing the discrete contributions to the global system, performing global iterations, and adapting the discretization [DinShe00, FlaLoy97a]. The procedures performing these operations are responsible for the construction and dynamic control of the parallel decomposition and its interactions with solution processes. Although this decomposition may be independent of the other three levels, information on how operations and communications are performed within and between those levels is central to the control of the dynamic load balancing procedures.

For the proposed work, we will use Zoltan to redistribute data after parallel meshing and adaptive mesh refinement. The Zoltan Dynamic Load-Balancing Library [Dev99] was developed to provide parallel, dynamic repartitioning to such applications. Zoltan's design is data-structure neutral, so that it can be used in many different applications without imposing restrictions on the data structures those applications use or requiring application developers to understand Zoltan's data structures. This makes it easy to support control with respect to the entities at the various levels in the hierarchic domain decomposition and also easy to integrate with target applications. Moreover, Zoltan includes a suite of algorithms, allowing developers to experiment easily with different algorithms to see which is best for their applications. Zoltan currently includes geometric algorithms (Recursive Coordinate Bisection, Recursive Inertial Bisection), tree-based algorithms (Octree Partitioning/SFC, Refinement Tree Partitioning), and graph-based algorithms (through interfaces to the popular ParMETIS [Sch99] and Jostle [Wal00] packages). We will provide interfaces from the TSTT software to Zoltan so that applications using the TSTT software can use Zoltan seamlessly. We will add support to Zoltan for hierarchical machine models of heterogeneous parallel computers. These models will account for various processor speeds, memory capacities, cache structures, and network speeds and topologies. This capability will be incorporated using RPM [TerBea00, FlaTer00] as a prototype for unstructured grids and PADRE [Qu98b] as a prototype for structured overlapping grids. We will also add to Zoltan the specific load balancing methods MLB [Qu97] developed for parallel adaptive overlapping structured grids in Overture. Our collective research will provide crucial support for adaptive methods on advanced architectures employing both distributed memory (using MPI) and shared memory (using threads). This will facilitate investigation of optimal partitioning strategies for linear and eigen solvers and preconditioners of interest in the application domains of interest.

An aspect of parallel computing important to the effectiveness of dynamic load balancing methods are the languages and mechanisms for supporting inter-processor communications. As part of our efforts in this area we plan to examine the modifications needed for our methods to take advantage of advances in this area.

4 Relationship to SciDAC Applications and other Computational Science Efforts

The goal of the TSTT Center is to develop and deploy a set of computational tools that will significantly and positively impact the ability of SciDAC applications researchers to employ advanced terascale mesh-based simulation technology in their research. Because of the sophistication of most existing advanced tools in this area, their availability has until now been limited to only the most ambitious application developers. We will address this deficiency by offering significant capability for generating different kinds of meshes for complex geometry and employing these meshes in scientific simulations through easy-to-use application-appropriate interfaces. Since many important Office of Science applications are characterized by strongly non-uniform solution features, these tools will include capabilities for solution-adaptive mesh and solution improvement capabilities, including mesh refinement and front tracking. Interaction with a broad cross-section of SciDAC applications scientists early in the design phase of our software will be an essential step in ensuring that the interfaces we develop address the needs of a broad class of Office of Science applications.

We will initially target five application areas and address central issues within these areas. For each of these applications, we will assign a lead TSTT site and TSTT project leader to initiate one-on-one contact with the application team and to coordinate all TSTT technology delivery with that application. These one-on-one interactions will be our main method of interaction with our targeted applications. In addition, we will work with some 10 additional applications or with other aspects of the targeted applications, as a normal outgrowth of the extensive user interaction network of the TSTT team. Much of the additional applications work will be leveraged, i.e. at no cost to the TSTT center, while still contributing to its goal of broad acceptance of TSTT technology and tools.

To allow a forum for interaction with a broader user community, there will be applications representatives on the External Advisory Committee (see Section 8). Our application work will form the initial phase in the development of a common grid technology interface. This phase will be ``fire tested" through integration of TSTT tools (primarily existing tools to achieve early deliverables) with application codes. Acceptance of the common interface is central aspect to the plug and play usability of adaptive grid technologies that we are proposing. As our common interface evolves, we will assist the application codes in a migration to its use, while considering deeper uses for adaptivity and other advanced technologies within the same application areas. We will also broaden our involvement with applications, including ones not covered in the initial phases of our work, to ensure the widest attainable acceptance of these standards.

4.1 Fusion

“Center for Extended Magnetohydrodynamics Modeling”
PI S. Jardin , Application Point of Contact: S. Jardin (PPPL), TSTT point of Contact: J Flaherty (RPI)

The plasma physics simulation community has developed a simulation code framework that contains two primary 3D nonlinear simulation codes, parM3D and NIMROD. We expect to work closely with the parM3D code. The resistive and two-fluid MHD models used in parM3D are currently approximated with piecewise-linear finite elements on unstructured meshes in the poloidal sections and fourth-order finite differences in the toroidal direction.

We anticipate making a number of contributions to this application area. In particular, in the near term we will collaborate with parM3D scientists to introduce higher-order finite element schemes in the poloidal directions to better approximate the physics in this direction. In addition, we will explore the use of several existing mesh generation packages associated with the TSTT center to help automate the process of generating the flux-aligned unstructured meshes in the poloidal directions, significantly easing the burden of problem set up.

In the longer term, we will investigate the incorporation of adaptivity, mixed element meshes, and dynamic, parallel load balancing tools into the parM3D code. The fusion simulations of a certain class of "non-resonant" instabilities or of resonant instabilities at sufficiently low Magnetic Reynolds number (S) are relatively smooth, so that the gain from an adaptive approach would appear to be small. However for "resonant" instabilities at high S, corresponding more to present and future experiments, localized structure will appear around the resonant layers and there should thus be great benefit from an adaptive approach. There is particular interest in the accurate calculation of the non-linear simulation of these resonant high S instabilities in the presence of a population of energetic alpha-particles, which are created at an energy of 14 MeV by the Deuterium/Tritium fusion reaction. To address these issues, the parM3D research group has developed a hybrid fluid-particle description. Thus, in the longer term, the adaptive mesh techniques will need to interface with both the fluid and particle-in-cell algorithms.

4.2 Accelerator Design Simulations

“Advanced Computing for 21st Century Accelerator Science and Technology”
PIs: K. Ko and R. Ryne

Two issues characterize design of high energy particle accelerators: determination of forces resulting from magnets, RF chambers, etc. and integrating these forces and those resulting from mutual electrostatic repulsion in their action on particles through many accelerator turns.

Particle Forces and EM Field Calculations (Application Point of Contact: K. Ko (SLAC), TSTT Points of Contact: P Knupp (SNL) and D. Brown (LLNL)) The problem is to design RC cavities and waveguides to focus electron beam packets. For this purpose, a comprehensive set of parallel tools for solution of Maxwell's equations will be developed by this application center. The TSTT center will work closely with this effort by providing the latest advances in two- and three-dimensional mesh generation for complex geometries, adaptive techniques, higher-order discretization methods, and terascale computing. Currently, a CAD geometry description is used as input to the CUBIT unstructured mesh generation code. Two to four meshes per month are generated for design and analysis. Present solvers require conformal 3D meshes of perhaps mixed element type. TSTT will address future needs by providing technology to produce hybrid grids that match conformally orthogonal structured grids to unstructured mixed element grids and capabilities for improved hex-tet meshing. Computational experience has shown that the finite element solvers run faster when mesh quality is improved; in some cases iterative solvers do not converge unless the quality is improved. TSTT will provide access to mesh smoothing and quality optimization software that will reduce expected run times.

Particle Tracking (Application Point of Contact: A. Luccio (BNL), TSTT Point of Contact: J. Glimm (BNL)). Accurate statistical evaluation of particle orbits may involve up to 10^6 or more particles followed for 1000 or more turns. The calculations are used for design, optimized operation and accelerator control decisions. The particles are clumped nonuniformly towards the center of the chamber, but outliers (the ``halo'') are of interest also as wall collisions must be limited to prevent build up of radioactivity in the walls. The extreme aspect ratios between longitudinal and transverse dimensions and physical length scales are an additional issue. For these and other reasons, parallel decomposition tools, as proposed by the TSTT center, to cluster particles into spatially coherent load balanced domains will be essential to achieve the benefit of terascale computing. TSTT will also assist in the development of codes for adaptive (nonuniform grids and element order) solutions to Poisson's equation with realistic boundary conditions, allowing for the rapid solution of the space charge needed for practical usability of these codes through the required mapping iterations.

4.3 Chemically Reactive Flows

"A Computational Facility for Reacting Flow Science", PI H. Najm. Application Points of Contact: H. Najm and J. Ray (SNLL), TSTT Points of Contact: L. Jameson (LLNL), P. Fischer and L. Freitag (ANL)

This center has two broad goals: 1) the development and demonstration of a new software toolkit for chemically reacting flow simulations using the CCA infrastructure, and 2) the development of advanced computational tools for enhancing scientific discovery. In the near term, we will work with the application scientists to deploy the high-order spectral methods developed by Fischer (ANL) into their toolkit. Long term, we will collaborate with researchers involved in the development of a new software infrastructure using CCA-compliant tools for mesh adaptation, namely the GrACE package from Rutgers. As part of the CCA data object definition effort, we will work with the GrACE developers to define the appropriate abstract interfaces for structured, adaptive mesh computations. Using these interfaces, we will deploy our discretization libraries in the application code that uses GrACE, particularly targeting the fourth-order finite difference schemes desired by the physicists. MICS matching funds to support L. Jameson have been requested by the SNLL center for this effort.

"An Integrated Software Infrastructure for Unstructured Mesh Computational Simulation of Transport/Reaction Systems", PI J. Shadid. .Application Point of Contact: J. Shadid (SNL), TSTT Points of Contact: R. Leland (SNL)

This center proposes to apply the Salsa (SNL) and Chad (LANL) capabilities to chemically reacting flow problems using unstructured meshes and finite element and finite volume technologies. The application areas they target include biology and MEMS. In the near term we will work with this center and integrate our existing automated mesh generation technology for complex engineering and biological geometries into their applications. In the long term we will provide the results of our work in mesh quality for use in their application and collaborate with them to develop advanced adaptive and discretization approaches appropriate to their target applications. In turn, we will integrate their nonlinear iterative solution procedures with our discretization library.

Non-SciDAC application

Finally, we will collaborate with the experimental and simulation effort to model jet breakup and spray formation centered at ANL, working with C. Tzanos. The solution of this presently unsolved modeling problem is needed for spray combustion. We will use FronTier, adapted to the common TSTT interface, and extended through TSTT collaborations, to provide an accurate model of jet breakup. The sharp interface capabilities of this code are very well adapted to this problem.

4.4 Climate

Comprehensive Design and Development of the Community Climate System Model for Tera-scale Computers, PIs R. Malone and J. Drake; Application Point of Contact: J. Taylor and J. Larson (ANL), TSTT Point of Contact: L. Freitag (ANL)

The climate community is working to improve the NCAR Community Climate System Model (CCSM) code through the development of new model formulations, algorithms, parameterizations, and analysis tools. We will collaborate with climate research scientists in several aspects of their work. We will work with the developers of the Model Coupling Toolkit to expand its functionality in two ways. First, as part of our effort to define general interpolation schemes between different mesh types, we will work to develop general support for locally conservative flux interpolation schemes. This will improve the current methodology used in climate models which only supports global conservation and will lead to more accurate representations of the interface conditions between the different climate components. Second, we will help develop support in the model coupling toolkit for dynamic load balancing, particularly for the case in which the component models reside on dynamically changing sets of processors. Matching MICS funds are being requested in the Climate proposal to help support this work.

We will also work with BNL atmospheric scientists to provide adaptive capabilities for local, regional, and global transport of atmospheric species and aerosols. This effort will use the adaptive meshing capabilities, high-order discretization techniques, and general interpolation schemes developed and available as part of the TSTT Center.

4.5 Biology Simulation Models

Application Point of Contact: David Dixon (PNNL), TSTT Points of Contact: H. Trease (PNNL) and P. Fischer (ANL)

An emerging DOE application area for mesh-based modeling is in the area of computational biology. The face of biology is changing as new methods are brought to bear on the study of living systems. In fact, it is the transition from viewing biology as a set of components to treating it by a systems approach that will truly revolutionize the field. Biology does not occur at any one scale, rather it occurs from scales as small as an atomic ion, e.g., Ca^{2+} to scales as large as complex human populations and their interactions with the environment. A key element in any study of modern biology and the impact of understanding it in terms of human health or in terms of the impact of microbial systems on the environment is the ability to cross many temporal and spatial scales in order to connect diverse data and predict biological behavior at the systems level. A principle interest of DOE will be the measuring and modeling the reaction of biological systems to environmental insults such as radiation and pollution for both prokaryotic and eukaruotic cells and better utilizing the genomic data generated at a range of sites including the Joint Genome Institute. This is described in detail in *Bringing the Genome to Life* (www.science.doe.gov/ober/berac/genome-to-life-rpt.html). A key experimental component will be the generation of data from the new imaging tools that are being developed that promise to bring much higher resolution in the quest to look at more detailed cellular structures and to better map out the spatial complexity of the cell. New computational and mathematical models and tools are needed if such data is to be optimally used with the goal of actually predicting cellular behavior from a detailed understanding of the biochemistry. This is clearly a problem for the use of advanced grid technologies. The potential for applying computational mesh methods spans the range from molecular reactions and interactions, protein folding and protein-protein interactions, individual cells, tissues, organs, and whole body simulations. Creating computational models that faithfully capture the geometry and the biophysics of these biological systems relies heavily on geometry/mesh generation, adaptive meshes and discretization methods as well as the incorporation of complex boundary and initial conditions into simulations that support research and experiments. DOE has active programs in several areas, including: (1) the computational cell modeling in the Microbial Cell Project (LAB 01-20 OBER/ASCR/SC), (2) computation simulation frameworks in Advanced Modeling and Simulation of Biological Systems (LAB 01-21 ASCR/OBER/SC), and (3) the modeling in the Low Dose Radiation Research Program (LAB 01-18 OBER/SC). These programs would benefit from the work that will be performed by the members of this ETC and in the application components that address specific issues in these various programs. Other specific areas in biophysics modeling that the TSTT Center's members are actively pursuing include; cell physiology modeling, lung physiology modeling, vascular disease, electrocardiac physiology, artificial heart valves, and tissue engineering.

4.6 Other Applications

In order to assure broad applicability of TSTT tools, we will continue collaborations in a number of applications areas not called out explicitly above.

Adaptive free surface models using the FronTier front tracking code will be developed collaboratively by several TSTT sites. Liquid metal coolant for the walls of the Tokamak reactor will be studied as well as the design of a liquid mercury target for a proposed muon collider, in collaboration with physicists specialized in these areas. ICF studies of fluid instabilities and study of the jet breakup for spray modeling and spray combustion in a diesel engine will be studied in conjunction with experimental and computational physicists. The recognized capabilities of front tracking to eliminate interface diffusion make this approach to these problems attractive.

Flow in porous media requires terascale and adaptive methods, but most commercial petroleum simulations are still performed on a single processor. Error analysis for these flows is not purely numerical. The subgrid models for these simulations are grid dependent, and so error analysis must assess the accuracy of the grid dependent parameterization as well as the numerics. Petroleum and application physicists are part of our application network and will join in this work.

The multiphysics code ALEGRA will be used for plasma physics simulations and other non-defense applications in collaboration with several universities. TSTT will enhance the capabilities of ALEGRA for these studies by providing: 1) a set of interoperable meshing tools, including Overture and CUBIT, 2) a mesh smoothing, optimization, and quality improvement toolkit such as MESQUITE, 3) an automatic r-type adaptive scheme for ALE, and 4) hierarchical terascale computing infrastructures for partitioning, load balancing, and parallel mesh generation.

5 Relationship to other SciDAC ISIC efforts

Because meshing and discretization pervades many aspects of PDE solutions, we must interact with a broad spectrum of other tool providers to ensure that our software and interfaces are both flexible and efficient on terascale computers. To accomplish this goal, we are committed to working with the other SciDAC Integrated Software Infrastructure Centers. Our planned interactions with four proposed ISIC centers are given below.

5.1 Center for Component Technology for Terascale Simulation Software ISIC

PI: Robert. Armstrong (SNLC), TSTT Point of Contact: L. Freitag (ANL)

A pervading theme in the proposed TSTT center work is the creation of interoperable technologies for meshing and discretization tools. Critical to this effort is the development of common interfaces that support mesh and data field access at many different levels of sophistication, and we will collaborate with the Center for Component Technology and the CCA Forum to define these interfaces. We will deploy these interfaces, along with those necessary to be CCA compliant, within a number of our existing tools, e.g., MEGA and Trellis (RPI), Overture (LLNL), and NWGrid (PNNL), as well as in a number of new TSTT technologies such as the Discretization Library and the tools for mesh quality optimization. These tools will then be released and accessible from the CCA component repository for wide scientific use.

We note that the abstraction of the detail behind the low-level interfaces to the operators in the Discretization Library is critical if the operator libraries are to interact with several different mesh types through common interfaces while also providing performance-critical services. One particularly interesting research question is the role of the component model in the composition of numerous discrete operators. The issues associated with run-time efficiency of codes resulting from the composition of operators has been addressed by the ROSE project for the case of object-oriented frameworks [Qu00]. We will work together with the ROSE project and the CCA Forum to address these issues for component models.

5.2 Terascale Optimal PDE Simulations (TOPS) ISIC

PI: David Keyes (ODU), TSTT Point of Contact: J. Flaherty (RPI)

The solution of large-scale PDE problems typically leads to a large, sparse linear or nonlinear algebraic system. Dimensions easily range into the hundreds of millions. With the emergence of hierarchical computing systems and inexpensive clusters this number will dramatically rise in the next few years. Suboptimal algebraic solution strategies can easily dominate the computational time and often mean the difference between being able or not being able to solve a problem. Efficiency in a terascale environment is a must. Fortunately, the TOPS ISIC is addressing these issues and developing the optimal multiscale solution software and strategies needed for the applications cited in this proposal and others intending to use our mesh and discretization libraries. This is an excellent opportunity for cooperation and collaboration. Reversing the perspective, the meshing software, adaptive procedures, and discretization library to be developed by this effort provide the TOPS ISIC with a source of algebraic systems that will be obtained by the most advanced technologies. The two centers will work together to develop an interface that ensures that the discretization and solution software will be interoperable. Both groups will collaborate throughout the course of this investigation to ensure that efforts and software developments are coordinated. Each center will use the others' software as a testbed to provide immediate feedback on difficulties and needed enhancements.

5.3 High-End Computer System Performance: Science and Engineering ISIC

PI: David Bailey (LBNL), TSTT Point of Contact: Dan Quinlan (LLNL)

The High-End Computer System Performance Science and Engineering (HCSP) ISIC proposal will focus Performance Analysis and Optimization for complex memory-hierarchies. Such architectures are common to all modern parallel computers and present difficult challenges for scientific computing generally. The tools being developed within the performance ETC include tools to both analyze and report existing performance and explicitly optimize application source using source-to-source transformations. The mechanisms developed for source-to-source transformations automate the introduction of optimizations that address memory-hierarchies (particularly cache). This approach is specifically directed to the compile-time optimization of libraries such as the Discretization Library that we propose. Our development of the Discretization Library and its use of predefined mathematical operators to simplify the development of applications using complex computational meshes will have direct benefits from the research work on ROSE proposed within the HCSP ETC. Since the operators defined within the Discretization Library are used within user defined applications, the library cannot see the context of their use with one another. Only a compile-time mechanism can see the sufficient context of the use of multiple operators within the application source (within and across program statements). The work on ROSE by the HCSP ETC is explicitly a mechanism to recognize the use of high-level abstractions and automate the source-to-source transformations required to make them efficient. We expect to work closely with researchers there in the development and testing of the ROSE compile-time approach to automate optimization of application codes using our Discretization Library. Additional work will define codes using our Discretization Library. Additional work will define benchmarks specific to evaluating the performance of the Discretization Library. This proposal shares some of the same personnel with the HCSP ETC. We expect that the Discretization Library itself will be influenced by the development of this optimization research within the HCSP ETC.

5.4 The DOE SciDAC Visualization Software Infrastructure Center (VSIC)

PI: Rick Stevens(ANL), TSTT Point of Contact: L. Freitag (ANL)

We will collaborate with the Visualization Software Infrastructure Center (VSIC) to ensure interoperability between computational and visualization meshes. In particular, we will create common interfaces between the TSTT software and the visualization center's scalable visualization framework. This would provide a unified view of the numerical data to the visualization tools, providing a significant reduction in the time consuming and error prone task of creating one-to-one interfaces between the numeric and visualization tools. The visualization center also plans to leverage the TSTT infrastructure for automatically transferring data from one mesh and discretization strategy to another. In particular, and we will provide interpolation to a structured grid which will enable the TSTT datasets to utilize a wider variety of visualization algorithms. Longer term we will work with the visualization center to develop algorithms that can operate directly on the simulation results in native mesh format without the need for interpolation or sampling.

6 Software Deployment Plan

We address three primary issues associated with center software: 1) collaborative development of source code across multiple institutions, 2) software maintenance and deployment during the life of the center, and 3) continued software maintenance beyond the life of the center. The process of new source code development across multiple institutions will be facilitated by the creation of a central source code repository that will be used by all center developers. A rigorous coding standard will be developed and adhered to for code maintainability and extensibility. We will use *cvs* to manage simultaneous development, *autoconf* to manage and aid portability, and will adhere to the development plan determined by the technical leads and approved by the center executive committee (see Section 8). To ensure the robustness of our software, we will develop rigorous, testing mechanisms that will be used routinely to automatically generate and distribute error reports. In addition, the center will set up an automatic bug-tracking facility to ensure that all user requests and questions are monitored and answered.

All new software and the interface specifications developed as part of this project will be available in the public domain via a single web-based repository and as CCA components as appropriate. Incremental releases will incorporate new functionality based on both the technical plan devised to accomplish the center goals and on feedback obtained through direct interaction with our application users. In all cases, we will provide extensive documentation, test cases, portability tests, and sample programs to facilitate the use of our tools. Tutorials and workshops will be offered at leading conferences to introduce new users to the center software. Preexisting software such as CUBIT, Trellis, Overture, and NWGrid will also be made freely available to the academic and laboratory scientific community both through our repository and through the CCA repository as appropriate. These releases may be either object code or source code, depending on preexisting license agreements, but in all cases the software will comply with the interfaces defined by the TSTT center and will therefore be available for use by application scientists. These codes will continue to be maintained by the original developers at their respective institutions.

To help us manage this process, we will hire a software engineer whose primary duties will be to manage the software repository, monitor the results of the routine tests, help with code documentation, maintenance, and portability, and serve as the initial point of contact for user interactions. Detailed technical questions will be referred to the primary code developers as needed. To facilitate a deeper understanding of the center codes, the software engineer will travel to the various sites periodically to meet with the primary developers.

To ensure the viability of center software beyond the five-year life of this center, we envision some combination of the following three avenues would be pursued per DOE guidance. First, DOE could continue to support software maintenance through the continued activities of the software engineer described above. No new development of center-supported research would take place, but the central software repository and single point of contact for users could be maintained. In addition, basic activities needed to ensure code longevity, such as porting the software to new computer architectures, could be pursued. Second, an open source, community-maintained model, similar to the Linux model, could be adopted. In this scenario, the code would not be officially maintained in a single location, but successful aspects would continue to be developed and grown by the scientific community because of the inherent benefits to do so. A third option would be to encourage a private software company to continue to develop and maintain the center software, but require that the existing code remain open source. This is the model used by Kitware for its vtk software; revenues are earned through activities such as user support, and the development of extensions and better packaging.

7 Work Schedule and Deliverables

The application deliverables and the development of new technology are closely intertwined. In the first two years of the proposal we will focus on the immediate needs in the application areas, the results of which will feed into our research on interface definition. In the final three years of the proposal we will continue our work with applications, focusing on long-term goals and integration of advanced technologies developed in the center. Throughout we will focus on software development and deployment. The milestones and deliverables listed below are based on a yearly cycle beginning July 1, 2001.

Year 1

- Initial definition and deployment of low level interfaces for accessing mesh and field data in collaboration with the CCA Forum (All: ANL leads)
- Design and initial implementation of the discretization operator library; work closely with the developers of mesh, solver, and application technologies to define the interfaces necessary to ensure that all levels of access are exposed (All: LLNL, RPI lead)
- Development of a priori mesh quality assessment and improvement tools for unstructured and structured meshes (ANL, ORNL, SNL, LLNL)
- Structured mesh generation improvement to increase speed and functionality (LLNL, ORNL)
- Extraction of the essential algorithms and routines from FronTier to begin implementation of a stand alone front tracking software package, FrontierLite (BNL, SB)
- Insert existing high-order discretization and mesh generation methods into the fusion (RPI), accelerator (LLNL, SNL), and chemistry (LLNL,SNL) applications
- Begin to examine how to employ information on the mesh hierarchy (levels 0-2) for the effective construction of the parallel decomposition (level 3) (LLNL, RPI, SNL)
- Extend existing dynamic load balancing technologies to effectively account for heterogeneous parallel computer environments (RPI, SNL)
- Make a preliminary assessment on how hierarchical information we generate may be exploited in downstream applications, focusing on iterative solver technology under development in related ISIC's (SNL,ANL,RPI).
- Hire a software engineer, set up the web site repository, cvs, autoconf, and establish a coding standard (All)

Year 2

- Refine and complete definition of the low level interfaces and deploy them within the TSTT tools, e.g., Overture, Trellis, NWGrid, FrontierLite (All)
- Preliminary definition of the interfaces for the common interfaces for the geometry (CAD and image) and physics attributes. (All: RPI, SNL lead)
- Preliminary work to define the problem hierarchy and interfaces necessary for hybrid methods (All: RPI, LLNL lead)
- Deploy the a priori mesh improvement tools into the TSTT mesh generation and management tools via the low level interfaces defined in FY01 (ANL, ORNL, SNL)
- Complete interface specifications for the discretization library; begin implementation of the low-level operators and interface them to TSTT mesh technologies; work with the appropriate ISICs to define interfaces for interacting with linear and nonlinear algebraic components (All: LLNL,RPI lead)
- Continued enhancements to the mesh generation capabilities (All)
- Initial work to merge front tracking algorithms with existing codes including TSTT tools such as Overture (BNL, SB, LLNL)
- Begin implementation of the parallel decomposition (level 3) that uses information on the mesh hierarchy for effective parallel control (LLNL, RPI, SNL)
- Consider how parallel mesh generation procedures will be extended to take full advantage of the hierarchic structures (LLNL, PPNL, RPI, SNL)
- Develop software design principles facilitating downstream use of hierarchical data structures (SNL,ANL,RPI).

- Insertion of initial parallel adaptive simulation technologies into the fusion (RPI), accelerator (BNL), climate (ANL) and biology (PNNL, ANL,SNL) applications.
- Insertion of existing mesh generation and geometry capturing technologies for complex geometries in the biology and chemistry applications (PNNL, ANL, SNL)
- Documentation of new libraries, development of comprehensive test suites and automatic testing facilities, release of existing codes via the web repository (these activities continue through FY05) (All)

Year 3

- Publish the low level common interfaces in a specification document and establish a forum for soliciting and receiving community feedback (All: ANL leads)
- Continue work on the hierarchy definition and establishment of geometry interfaces, modify existing tools to conform to the common interfaces (All: RPI, SNL, LLNL lead)
- Complete implementation of low-level discretization operators; design and begin to implement the high level interfaces and develop support for parallel implementation (All: LLNL, RPI lead)
- Develop a fully space-time conservative scheme using front tracking to adapt the grid; interface with TSTT mesh generation capabilities via the low level interface specification (BNL, SB, LLNL, RPI, PNNL)
- Develop a posteriori mesh quality improvement algorithms based on error estimators (ANL, ORNL, SNL, LLNL)
- Extend the dynamic load balancing technologies of Zoltan to take full advantage of the information in the mesh hierarchy (LLNL, RPI, SNL)
- Prototype design modifications facilitating downstream use of hierarchical data structures (SNL,ANL,RPI).
- Initiate work on locally conservative flux interpolation schemes in the Climate application (RPI, LLNL lead)
- Continued work to deploy parallel adaptive techniques into the fusion (RPI), accelerator (BNL), climate (ANL) and biology (PNNL, ANL,SNL) applications
- Continued software support, documentation, testing, and release a priori mesh quality improvement tools (All)

Year 4

- Modify and extend low level common interfaces per consensus with community feedback; demonstrate the use of multiple TSTT mesh generation tools using the hierarchy interfaces developed in FY02 and FY03 in the CAD-based geometries associated with the accelerator (LLNL) and chemistry (SNL) applications (All)
- Experiment with the use of hybrid meshes and alternative discretization strategies using TSTT mesh generation capabilities and the discretization library in the fusion application (RPI)
- Develop error estimator capabilities in the discretization library; begin performance optimization work to address parallel and cache-based optimizations; develop benchmarks to characterize the performance of applications using different levels of interfaces to the discretization library operators (All: LLNL, RPI lead)
- Complete integration of front tracking algorithms with all the TSTT mesh generators and the discretization library via the low level interface definitions (SNL, SB)
- Continued work of a posteriori mesh quality improvement; integrate mesh improvement tools and discretization library tools (ANL, ORNL, SNL, LLNL)
- Continue to improve the performance of parallel dynamic load balancing procedures (ANL, RPI, SNL, LLNL)
- Improve capability to make downstream use of hierarchical data structures (SNL,ANL,RPI).
- Begin work to automate the choice of adaptive strategies used with TSTT tools based on error indicators (All: RPI leads)
- Demonstrate parallel adaptive computations in the climate and biology applications; continue to work with other applications (All)
- Release the discretization library, offer conference tutorials and workshops to facilitate use of TSTT tools, publish the common interfaces for high level problem descriptions (e.g., access to CAD geometries), continue software support and maintenance (All)

Year 5

- Continued work to analyze and improve performance characteristics of TSTT tools using both low and high level interfaces in targeted applications (All: LLNL leads)
- Complete center work on the discretization, mesh quality improvement, and front tracking libraries and release final versions into the public domain (All)
- Modify high level interfaces per community feedback and publish a specification document (All: RPI, SNL lead)
- Employ fully automated, efficient parallel adaptive simulations in terascale simulations (All)
- Continued research on automatic selection of adaptive procedures; implement various strategies in TSTT mesh management software tools; publish recommended strategies (All: RPI leads)
- Explore long-term code management and support strategies with DOE guidance (All)

The Role of Each Institution

Argonne National Laboratory. Argonne will co-lead the effort on mesh quality and optimization with SNL. We will contribute to the discretization library, interoperable meshing, and terascale computing efforts. We will lead the interactions with the CCA Forum, the Visualization ISIC, and climate and Biology applications.

Brookhaven National Laboratory. BNL provides the Director of TSTT. It will lead the application effort and provide applications contact for climate (adaptive simulation) and accelerator design (PIC codes) and several other applications. It will provide support for finite element discretization and for FronTier development.

Lawrence Livermore National Laboratory. LLNL will co-lead design and implementation of the meshing hierarchies and mesh component design with RPI. LLNL will be responsible for the enhancement of mesh generation tools for structured and hybrid meshes, the discretization operator library performance optimizations through ROSE, and will coordinate delivery of hybrid mesh technology for the Accelerator design project.

Oak Ridge National Laboratory. Oak Ridge will collaborate with SNL, ANL, and others on mesh quality, enhancement and interoperability, and will aim at applications in climate and chemically reacting flows.

Sandia National Laboratory. Sandia will collaborate with LLNL, RPI, and others on interoperable meshing, domain decomposition, and terascale computing. SNL will lead interactions with the SLAC accelerator and ALEGRA groups and co-lead the work on mesh quality and optimization with ANL.

Rensselear Polytechnic Institute. RPI will lead development of meshing and discretization technologies for high order methods, co-lead the development of the meshing hierarchies with LLNL and Sandia, co-lead the development of the discretization library with LLNL, provide key components for the terascale computing technologies and will coordinate the development of contributions to the fusion application.

Pacific Northwest National Laboratory. PNNL will collaborate on interoperable meshing by creating a CCA-compliant, parallel mesh generation and discretization component and will co-lead the terascale computing effort and coordinate the efforts in the biology applications for biology simulation models.

State University of New York, Stony Brook. Stony Brook will be the lead for interoperability of FronTier and the development of FronTier-Lite. It will lead in the development of conservative and higher order versions of FronTier. It will lead in the spray simulations and oil reservoir applications.

8 Center Management Plan

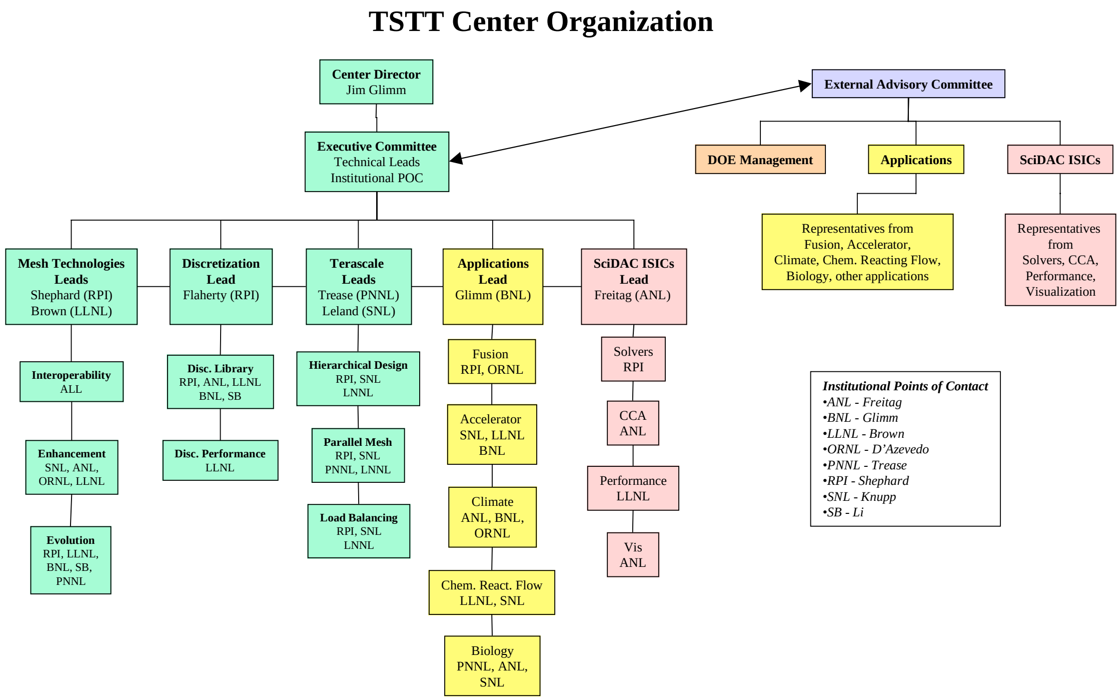

Figure 4: The TSTT Center Organization. The center director, Jim Glimm, is supported by the executive committee consisting of the technical leads and the institutional points of contact. The external advisory committee, consisting of representatives from DOE management, the application areas, and ISIC centers, will provide input to the center through annual meetings with the executive committee. The institutions listed in each box correspond to TSTT participation.

The proposal team consists of researchers from six DOE national laboratories (ANL, LLNL, BNL, SNL, PNNL, and ORNL) and two universities (Rensselear Polytechnic Institute and SUNY Stony Brook). The TSTT center director will be Jim Glimm. He will be supported by a strong executive committee consisting of the lead investigators responsible for cross-cutting technical activities as well as the points of contact for each participating institution. The technical leads are identified as follows: Brown and Shephard (mesh generation and adaptive techniques), Flaherty (high order discretization technologies), Trease and Leland (terascale computing), Glimm (applications liaison), and Freitag (ISIC liaison). The institutional points of contact are Freitag (ANL), Brown (LLNL), Glimm (BNL), Knupp (SNL), Trease (PNNL), D'Azevedo (ORNL), Shephard (RPI), and Li (SUNY SB). In addition, we will create an advisory committee consisting of representatives from the application centers and other ISICs to ensure the relevance of our proposed work. The executive committee will meet biannually at rotating sites and the technical groups will meet as necessary to accomplish the center goals. These meetings will be augmented by monthly teleconference calls and use of the Access Grid collaboratory facilities as they become more prevalent. The advisory committee will meet with the executive committee yearly to offer guidance and receive regular progress reports.

9 Literature Cited

[AdjBel01] S. Adjerid, B. Belguendouz, and J.E. Flaherty, "A posteriori finite element error estimation for diffusion problems," *SIAM J. Sci. Comput.*, 2001, to appear.

[AdjFla99] S. Adjerid, J. E. Flaherty and I. Babuska. A posteriori error estimation for the finite element method-of-lines solution of parabolic problems. *Mathematical Models and Methods in Applied Science*, 9:261-286, 1999.

[AdjDev00] S. Adjerid, K.D. Devine, J.E. Flaherty, and L. Krivoldonova, "A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems," *Comp. Meth. Appl. Mech. Engng.*, 2000, submitted.

[BeaShe97] M.W. Beall and M.S. Shephard. A General Topology-Based Mesh Data Structure. *Int. J. Num. Meth. Engng.*, 40(9):1573-1596, 1997.

[BeaShe99] M. W. Beall and M. S. Shephard. An Object-Oriented Framework for Reliable Numerical Simulations. *Engineering with Computers*, 15(1):61-72, 1999.

[BelMad94] F. Ben Belgacem and Y. Maday, "Nonconforming spectral element methodology tuned to parallel implementation", Comput. Meth. Appl. Mech. Engrg.,116, pp. 59-67 (1994).

[BisDev94] R. Biswas, K. Devine, and J.E. Flaherty, "Parallel adaptive finite element methods for conservation laws," Appl. Numer. Maths., 14: 255-284, 1994.

[Br99] Brown, D., "Overture Operator Classes for Finite Volume Computations on Overlapping Grids, User Guide," Lawrence Livermore National Laboratory research report, UCRL-MA-133649, 1999.

[BrCh97] D. L. Brown and Geoffrey S. Chesshire and William D. Henshaw and Daniel J. Quinlan. Overture: An Object Oriented Software System for Solving Partial Differential Equations in Serial and Parallel Environments, *Proceedings of the Eight SIAM Conference on Parallel Processing for Scientific Computing*", 1997.

[BrHe00] David L. Brown and W.D. Henshaw. Adaptive Mesh Refinement Routines for Overture. Lawrence Livermore National Laboratory, *Research Report UCRL-MA-140918*, 2000.

[Bri95] Brislawn, K.D., D.L. Brown, G. Chesshire, and J.S. Saltzman, "Adaptive Composite Overlapping Grids for Hyperbolic Conservation Laws," Los Alamos National Laboratory report LA-UR-95-257 (1995).

[Bu99] Bugrim, A., J. Keizer, R. Fontanilla,, and R. Nuccitelli, "Initiation mechanism of the fertilization Ca2+ wave in Xenopus eggs," presented at the Pacific Symposium on Bio-Computing, Hawaii, 1999.

[Car96] Cartwright, D., Albers, R., George, D., Gray, M., Kress, J., Kwan, T., Nebel, R., Soran, P., Straub, G., Trease, H., Valone, S., Modeling and Simulation for the Semiconductor and Optoelectronic Industries, LA-UR-96-1771.

[CarBic01] G.F. Carey, G. Bicken, V Carey, C. Berger and J. Sanchez, "Locally constrained projections on grids", *Int. J. Numer. Meth. Engng.*, 50:549-577, 2001

[CCA] Common Component Architecture Forum Web Page, http://www.cca-forum.org/, 2001.

[CheGli00] B. Cheng, J. Glimm, X. L. Li, and D. H. Sharp. Subgrid models and DNS studies of fluid mixing. In *Proceedings of the 7th International Conference on the Physics of Compressible Turbulent Mixing, (1999)*, 2000 submitted.

[CocKar00] B. Cockburn, G.E. Karniadakis, and G.W. Shu, (Eds.*), Discontinuous Galerkin Methods: Theory, Computation and Applications*, Springer, 2000.

[Cs99] Csutora, P., H.Y. Kim, A. Bugrim, K.W. Cunningham, R. Nuccitelli, J.E. Keizer, M.R. Hanley, and R.B. Marchase, "Calcium influx factor is synthesized by Yeast and mammalian cells depleted of organellar calcium," *PNAS*, vol. 961999, pp. 121–126.

[CUBIT] CUBIT Mesh Generation Environment, *Technical Report Number SAND94-1100*, Sandia National Laboratories, 1999.

[DAz91] E. D'Azevedo. Optimal Triangular Mesh Generation by Coordinate Transformation. SIAM Journal on Scientific and Statistical Computing, 12(4): 755-786, 1991.

[DAz99] E. D'Azevedo. On Optimal bilinear quadrilateral meshes. *Engineering with Computers*, 15(3):219-227, 1999.

[DAz00] E. D'Azevedo. Are Bilinear Quadrilaterals Better than Linear Triangles? *SIAM Journal on Scientific Computing*, 2291):198-217, 2000.

[deAYec98] V. de Almeida, A. J. Yeckel, J. J. Derby. Parallel Computing of Coupled, Incompressible Flows with 3-D Mixed Finite Elements. *Presented in AIChE annual meeting*, Miami, Florida, Nov 1998.

[deA99] V. de Almeida. Domain Deformation mapping: Application to Variational Mesh Generation. *SIAM Journal on Scientific Computing*, 20(4):1252-1275, 1999.

[deADer00] V. de Almeida, J. Derby. Construction of Solution Curves for Large 2-D Problems of Steady-State Flows of Incompressible Fluids. *SIAM Journal on Scientific Computing*, 22:285-311,2000.

[deCShe99] H.L. deCougny and M.S. Shephard. Parallel volume meshing using face removals and hierarchical repartitioning. *Comp. Meth. Appl. Mech. Engng.*, 174(3-4):275-298, 1999.

[deCShe99a] H. L. de Cougny and M. S. Shephard. Parallel Refinement and Coarsening of Tetrahedral Meshes. *Int. J. Numer. Meth. Engng.*, 46(10):1101-1125, 1999.

[DevHen99a] K. D. Devine, B. A. Hendrickson, E. Boman, M. St. John, and C. Vaughan, Zoltan: A dynamic load balancing library for parallel applications; developer's guide, Tech. Report SAND99-1376, Sandia National Laboratories, Albuquerque, NM, 1999.

[DevHen99b] K. D. Devine, B. A. Hendrickson, E. Boman, M. St. John, and C. Vaughan, Zoltan: A dynamic load balancing library for parallel applications; user's guide, Tech. Report SAND99-1377, Sandia National Laboratories, Albuquerque, NM, 1999.

[DeyOba99] S. Dey, R. M. O'Bara and M. S. Shephard. Curvilinear mesh generation in 3D. *Proc. 8th Int. Meshing Roundtable*, SAND 99-2288, pp. 407-417, 1999.

[DeyShe97] S. Dey, M. S. Shephard and J. E. Flaherty. Geometry-based issues associated with p-version finite element computations. *Comp. Meth. Appl. Mech. Engng.*, 150:39-55, 1997.

[DinShe00] M. Dindar, M. S. Shephard, J. E. Flaherty and K. E. Jansen. Adaptive CFD analysis for rotorcraft aerodynamics. *Comp. Meth. Appl. Mech. Engng.*, 189(4):1055-1076, 2000.

[DraRob00] R. P. Drake, H. F. Robey, B. A O. A. Hurricane, Remington, J. Knauer, J. Glimm, Y. Zhang, D. Arnett, D. D. Ryutov, J. O. Kane, K. S. Budil, and J. Grove. Experiments to produce a hydrodynamically unstable spherically diverging system of relevance. *Astrophysical Journal*, 2000. submitted.

[ESI] The Equation Solver Interface Group Official Web Page, http://z.ca.sandia.gov/esi/, 2001.

[Fa99] Fast, P., *Dynamics of Interfaces in non-Newtonian Hele-Shaw Flow*, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, 1999.

[FaKo01] Fast, P., Kondic, L., Shelley, M. J., Palffy-Muhoray, P., *Pattern Formation in non-Newtonian Hele-Shaw Flow*, Physics of Fluids, May 2001, in press.

[FaHe01] Fast, P., Henshaw, W. D., *Time-Accurate Simulation of Viscous Flow around Deforming Bodies using Overset Grids*, AIAA Paper 2001-2604, submitted.

[FaSh99] Fast, P., Shelley, M. J., *An Overset Grid Method for Free-Surface Flow*, LLNL technical report UCRL-JC-136484, November 1999.

[FEI] Finite Element Interface to linear solvers (FEI) Web Page, http://z.ca.sandia.gov/fei/, 2001.

[Fis97] P. F. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations, *J. of Comp. Phys.* 133 84-101 (1997).

[FlaLoy99] J. E. Flaherty, R. M. Loy, M. S. Shephard, M.L. Simone, B. K. Szymanski, J. D. Teresco and L. H. Ziantz. Distributed Octree Data Structures and Local Refinement Method for the Parallel Solution of Three-Dimensional Conservation Laws. *Grid Generation and Adaptive Algorithms*, M.W. Bern, J.E. Flaherty and M. Luskin, eds, Vol. 113, IMA Vols. in Math. & its Appl., Springer, pp. 113-134, 1999.

[FlaLoy97a] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco and L. H. Ziantz. Adaptive local refinement with octree load-balancing for the parallel solution of three-dimensional conservative laws. *J. Parallel and Distributed Computing*, 47:139-152, 1997.

[FlaLoy97b] J.E. Flaherty, R. Loy, M.S. Shephard, B.K. Szymanski, J. Teresco, and L. Ziantz, "Predictive load balancing for parallel adaptive finite element computation," in H.R. Arabnia, Ed., Proc. *Int. Conf. Parallel and Distrib. Process. Tech. and Applics. (PDPTA '97)},* Las Vegas, 460-469, 1997. [GalLuc00] John Galambos, J. Cobb, J. A. Holmes, D. K. Olsen, Alfredo Luccio, and Nicholas D'Imperio. Parallel tracking in rings with space charge.In *Proceedings European Particle Accelerator Conference*, EPAC2000 , 2000.

[FlaTer00] J.E. Flaherty and J.D. Teresco, "Software for parallel adaptive computation," in M. Deville and R. Owens, Eds., Proc. *16th IMACS World Cong. on Sci. Comput., Appl. Maths. and Simul.*, Paper 174--6, IMACS, Lausanne, 2000.

[Fol99] N. Folwell and S. Mitchell, "Reliable Whisker Weaving via Curve Contraction," pp365-378, *7th Intl. Meshing Roundtable*, Dearborn MI, 1998.

[Fol01] Nate Folwell, SLAC, personal communication, 2001.

[FreGro99] Freitag, Lori, Gropp, Bill, Hovland, Paul, McInnes, Lois and Smith, Barry, *Infrastructure and Interfaces for Large-Scale Numerical Software*, Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications,Las Vegas, Nevada, June 1999.

[FrePla00] Freitag, Lori and Plassmann, Paul, *Local Optimization-based Simplicial Mesh Untangling and Improvement*, International Journal of Numerical Methods in Engineering, Volume 49, Issue 1-2, July, 2000, pages 109-125.

[FreOll97] Freitag, Lori, and Ollivier-Gooch, Carl, *Tetrahedral Mesh Improvement Using Face Swapping and Smoothing*, International Journal for Numerical Methods in Engineering, Volume 40, J. Wiley and Sons, 1997, pages 3979--4002.

[FreOll00] Freitag, Lori, and Ollivier-Gooch, Carl, *A Cost/Benefit Analysis of Simplicial Mesh Improvement Techniques as Measured by Solution Efficiency*, International Journal of Computational Geometry and Applications, Volume 10, Number 4, 2000, pages 361-382.

[FreKnu99] Freitag, Lori and Knupp, Patrick, *Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number*, Proceedings of the Eighth International Meshing Roundtable, Lake Tahoe, California, October, 1999

[Gab95] Gable, C.W., H.E. Trease and T.A. Cherry, Automated Grid Generation From Models of Complex Geologic Structure and Stratigraphy, Proceedings volume from the Third International Conference Integrating GIS and Environmental Modeling, Santa Fe, NM, Editors: Michael Goodchild, Louis Steyaert, Bradley Parks, Michael Crane, Carol Johnston, John Wilson, Sandi Glendinning, GIS World Books, Fort Collins, CO., 1995. (also LA-UR-95-2482)

[Gab96] Gable, C.W., H.E. Trease, and T.A. Cherry, Geological Applications of Automatic Grid Generation Tools for Finite Elements Applied to Porous Flow Modeling, Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Mississippi State University, April 1996, edit by B. Soni, J. Thompson, P. Eiesman and J. Hauser, ERC-MSU Press, 1996, ISBN 0-9651627-0-2, pp. 1007-1016.

[GarShe00] R. Garimella and M. S. Shephard. Boundary layer mesh generation for viscous flow simulations in complex geometric domains. *Int. J. for Numerical Methods in Engineering*, 49(1-2):193-218, 2000. [GliGra98] J. Glimm, M. J. Graham, J. Grove, X.-L. Li, T. M. Smith, D. Tan, F. Tangerman, and Q. Zhang. Front tracking in two and three dimensions. *Comput. Math. Appl.*, 35(7):1-11, 1998.

[Ge98] Gerritsen, M., "Composite adaptive grids down under: Tidal flow modelling and sail performance," *Proceedings of the 4th Symposium on Overset Composite Grid and Solution Technology*, Aberdeen, MD, 1998

[GivFla97] D. Givoli, J.E. Flaherty, and M.S. Shephard, Analysis of InP LEC melt flows using a parallel adaptive finite element scheme, *J. Cryst. Growth*, 180: 510-516, 1997.

[GliGro00] J. Glimm, J. Grove, X. L. Li, W. Oh, and D. H. Sharp. A critical analysis of Rayleigh- Taylor growth rates. *J. Comp. Physics* , 2001. Report No. SUNYSB-99-19, State University of New York at Stony Brook.

[GliGro99a] J. Glimm, J. Grove, X.-L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comp*., 21, 2000.

[GliGro99] J. Glimm, J. Grove, X.-L. Li, and N. Zhao. Simple front tracking. In G.-Q. Chen and E. DiBebedetto, editors, *Contemporary Mathematics*, volume 238, pages 133--149. Amer. Math. Soc., Providence, RI, 1999.

[GliHou00a] J. Glimm, S. Hou, H. Kim, Y. Lee, D. Sharp, K. Ye, and Q. Zou. Risk management for petroleum reservoir production: A simulation-based study of prediction Report No. SUNYSB-00-12, State University of New York at Stony Brook, 2000. Submitted to *J. Comp. Geosciences*.

[GliHou00] J. Glimm, S. Hou, H. Kim, D. Sharp, and K. Ye. A probability model for errors in the numerical solutions of a partial differential equation.*CFD Journal* , 9, 2000. Proceedings of the 8th International Symposium on Computational Fluid Dynamics, Bremen, Germany, 1999, SUNY-SB Preprint Number SUNYSB-AMS-99-11.

[GliHou01] J. Glimm, S. Hou, Y. Lee, D. Sharp, and K. Ye. Prediction of oil production with confidence intervals. SPE 66350, 2001. SPE Reservoir Simulation Symposium held in Houston, Taxas, 11-14 Feb.

[GroHol94] J. Grove, R. Holmes, D. H. Sharp, Y. Yang, and Q. Zhang. Quantitative theory of Richtmyer - Meshkov instability: Reply. *Phys. Rev. Lett.*, 73(23):3178--3178, 1994.

[GliKir00] J. Glimm, H. Kirk, X. L. Li, J. Pinezich, R. Samulyak, and N. Simos. Simulation of 3 D fluid jets with application to the muon collider target design. In *Advances in Fluid Mechanics III*, volume 26, pages 191--200.WIT Press, Southampton, Boston, 2000.

[GulFar00] H. Guillard and C. Farhat, "On the significance of the geometric conservation law for flow computations on moving meshes", *Computer Meth. Applied Mechanics and Engng*., 190:1467-1482, 2000.

[He98a] W.D. Henshaw. Ogen: An Overlapping Grid Generator for Overture, Lawrence Livermore National Laboratory, *Research Report UCRL-MA-132237*, 1998.

[He98b] W.D. Henshaw. Grid Functions for Overture, User Guide, Lawrence Livermore National Laboratory, *Research Report UCRL-MA-132231*, 1998.

[Hen98c] W.D. Henshaw. Finite Difference Operators and Boundary Conditions For Overture, User Guide, Lawrence Livermore National Laboratory, *Research Report UCRL-MA-132231*, 1998.

[HenLel94] The Chaco user's guide, version 2.0, B. Hendrickson and R. Leland, Tech. Rep. SAND 94-2692, Sandia National Laboratories, Albuquerque, NM, October 1994.

[HolFry99] R. L. Holmes, B. Fryxell, M. Gittings, J. W. Grove, G. Dimonte, M. Schneider, D. H. Sharp, A. Velikovich, R. P. Weaver, and Q. Zhang. Richtmyer-meshkov instability growth: Experiment, simulation, and theory. *J. Fluid Mech*., 389:55--79, 1999. LA-UR-97-2606.

[HenLel95a] An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations Bruce Hendrickson and Robert Leland, SIAM J. Sci. Stat. Comput., 16(2):452-469, 1995.

[HenLel95b] A Multilevel Algorithm for Partitioning Graphs, Bruce Hendrickson and Robert Leland, in Proc. Supercomputing '95.

[Hughes00] T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Pubns., ISBN 0486411818, 2000.

[Joe95] Joe, Barry. Construction of three-dimensional improved quality triagulations using loal transformations, *SIAM Journal of Scientific Computing*, 16, 1995, pp. 1292-1307.

[Jon00] Jones-Oliveira, J.B., Oliveira, J.S., Trease, H.E., Trease, L.L., Hybrid Grid Generation Using NWGrid, Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field

Simulations 2000, Whistler, B.C., Canada, Edit by B. Soni, J. Hauser, J. Thompson, and P. Eiesman, ERC-MSU Press, 2000, ISBN 0-9651627-3-7, pp. 479-486.

[Kar98] Karypis, G., and Kumar, V., *Multilevel Algorithms for Multi-Constraint Graph Partitioning*, University of Minnesota, Department of Computer Science / Army HPC Research Center, Technical Report #98-019, 1998.

[Ke98] Keizer, J., A. Bugrim, R. Fontanilla, R. Nuccitelli, "Quantitative studies of the fertilization Ca2+ wave in Xenopus laevis eggs," presented at the American Society for Cell Biology annual meeting, San Francisco, CA, 1998.

[KhaHam96] A. Khamayseh, B. Hamann. Elliptic grid generation using NURBS surfaces. *Computer Aided Geometric Design*, 13(4): 369-386, 1996.

[Kha98] A. Khamayseh. Non-parametric volume conserving smoothing. *Sixth International Conference on Numerical Grid Generation in Computational Field Simulations*, London, England, July 1998.

[KhaHan00] A. Khamayseh, G. Hansen. Quasi-orthogonal grids with impedance matching. *SIAM Journal on Scientific Computing*, 22(4):1120-1237, 2000. [LucDIm00] A. U. Luccio and Nicholas D'Imperio. Accelerator simulation and parallel computing. In *ICNSP 2000 (Proceedings of Banff Conference)*, 2000.

[KhaMas94] Khamayseh A, Mastin CW Surface Grid Generation Based on Elliptic PDE Models, *Appl Math Comput*, 65, pp. 253-64 1994

[Ki99] [Ki99] Kim, T.K., W.K. Silk and A. Y. Cheer, "A mathematical model for pH patterns in the rhizosphere of growth zones," Plant, Cell and Environment, (1999) 22, 1527-1538

[KlaShe00] Klaas, O. and Shephard, M.S., "Automatic generation of octree-based three-dimensional discretizations for partition of unity methods", *Computational Mechanics*., 25(2-3):296-304, 2000.

[Kn93] P. Knupp, The Fundamentals of Grid Generation, CRC Press, Boca Raton FL, 1993.

[Kn98] P. Knupp, Next-Generation Sweep Tool: A Method for Generating All-Hex Meshes on Two-and-one-half Dimensional Geometries, pp505-513, *7th International Meshing Roundtable*, Detroit MI, October 1998.

[Kn99] P. Knupp, Applications of Mesh Smoothing: Copy, Morph, and Sweep on Unstructured Quadrilateral Meshes, *Intl. J. Numer. Meth. Engr*., Vol. 45, Issue 1, pp37-45, 1999.

[Knupp00] P. Knupp, Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities, *Intl. J. Numer. Meth. Engr*., Vol. 48, pp1165-1185, July 2000..

[Knupp01] P. Knupp. Algebraic Mesh Quality Metrics, *SIAM J. Sci. Comp*., 2001, accepted.

[KnMaSh01] P. Knupp, L. Margolin, and M. Shashkov. Reference Jacobian Optimzation-Based Rezone Strategies for Arbitrary Langrangian Eulerian Methods, *J. Comp. Phys*., 2001, submitted.

[KnR00] P. Knupp and N. Robidoux, A Framework for Variational Grid Generation: Conditioning the Jacobian Matrix with Matrix Norms, *SIAM J. Sci. Comput*., Vol 21 No. 6, pp2029-2047, 2000.

[Kra00] J. A. Kraftcheck, Virtual Geometry: A Mechanism For Modification of CAD Model Topology For Improved Meshability, Master's Thesis, University of Wisconsin-Madison, 2000.

[Kup96] A. Kuprat, D. George, E. Linnebur, R. K. Smith, and H. Trease, Moving Adaptive Unstructured 3-D Meshes in Semiconductor Process Modeling Applications, VLSI Journal (to be published in special issue), LA-UR-961116, 1996.

[LiShe01] X. Li, M.S. Shephard and M.W. Beall, "Accounting for Curved Domains in Mesh Refinement", to appear Sixth U.S. Nat. Cong. on Computational Mechanics, U.S. Association for Computational Mechanics, 2001

[LucDIm00a] A. U. Luccio and Nicholas D'Imperio. Tracking in circular accelerators with parallel computing. In *Proceedings of the 6th International Compuational Accelerator Physics Conference*, 2000. Submitted to Physical Review Special Topics

[Mey98] R. Meyers and P. Tuchinsky, "The Hex-Tet Hex-Dominant Meshing Algorithm," pp151-158, *7th Intl. Meshing Roundtable*, Dearborn MI, 1998.

[Mit98] Scott A. Mitchell, "The All-Hex Geode-Template for Conforming a Diced Tetrahedral Mesh to any Diced Hexahedral Mesh", pp295-305, *7th Intl. Meshing Roundtable*, Dearborn MI, 1998.

[Musser96] D.R. Musser and A. Saina, *STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library*, Adison-Wesley, 1996

[Nei01] Nieplocha, J., Trease, H., Ju., J., Rector, D., and Palmer, B., Building an Application Domain Specific Programming Framework for Computational Fluid Dynamics Calculations on Parallel Computers, Tenth SIAM Conference on Parallel Processing for Scientific Computing, March 12-14, 2001.

[OhsFla00] T.K. Ohsumi, J.E. Flaherty, V.H. Barocas, S. Adjerid, and M. Aiffa, "Adaptive finite element analysis of the anisotropic biphasic theory of tissue-equivalent mechanics, Comp. Meths. in Biomech. And Biomech. Engrg, 3: 215-229, 2000.

[Qu95] D. Quinlan. A++/P++ Class Libraries, Los Alamos National Laboratory, *Research Report LA-UR-95-3273*, 1995.

[Qu97] D. Quinlan, M. Berndt, MLB: Multilevel Load Balancing for Structured Grid Applications, *Proceedings of the SIAM Parallel Conference*, Minneapolis, MN. March, 1997

[Qu98] Quinlan, D. AMR++: Object-Oriented Design for Adaptive Mesh Refinement, *Proceedings of the HPC'98 Conference*, Boston, Mass. April 5-9, 1998.

[Qu98b] K. Davis and D. Quinlan, The Parallel Asynchronous Data Routing Environment PADRE, Workshop on Parallel Object-Oriented Scientific Computing (POOSC'98), at 12th European Conference on Object-Oriented Programming (ECOOP'98), Brussels, Belgium, July 1998. LNCS 1543, Springer-Verlag, 1998.

[Qu00] Quinlan, D. ROSE: Compiler Support for Object-Oriented Frameworks, Proceedings of Conference on Parallel Compilers (CPC2000), Aussois, France, January 2000. Also published in special issue of Parallel Processing Letters (available soon).

[RemFla00] J.-F. Remacle, J. E. Flaherty and M. S. Shephard. An adaptive order discontinuous Galerkin formulation with orthogonal basis for solving a Rayleigh-Taylor instability. *SIAM Review*, submitted 2000.

[RemKar00] J.-F. Remacle, B. K. Karamete and M. S. Shephard. Algorithm oriented mesh database. *Proc. 9th Int. Meshing Roundtable*, SAND2000-2207, pp. 349-359, 2000. [WalGli99a] John Walter, James Glimm, John Grove, Hyun-Cheol Hwang, Xiao Lin, Bradley J. Plohr, David H. Sharp, and Dahai Yu. Eulerian front tracking for solid dynamics. In Kailasam Iyer and Shun chin Chou, editors, *Proceedings of the15th US Army Symposium on Solid Mechanics* , pages 343--366, Myrtle Beach, SC,1999. Battelle Memorial Institute, Battelle Press (www.battelle.org/bookstore).published on CD-ROM (ISBN 1-57477-083-7).

[Sch99] Schloegel, K., Karypis, G., and Kumar, V., *Parallel Multilevel Algorithms for Multi-Constraint Graph Partitioning*, University of Minnesota, Department of Computer Science / Army HPC Research Center, Technical Report #99-031, 1999.

[Sch01] R. Schneiders, Mesh Generation and Grid Generation on the Web, http://www-users.informatik.rwth-aachen.de/~roberts/meshgeneration.html, 2001.

[Shephar00] M. S. Shephard. Meshing environment for geometry-based analysis. *International Journal for Numerical Methods in Engineering*, 47(1-3):169-190, 2000.

[SheDey97] M. S. Shephard, S. Dey and J. E. Flaherty. A straightforward structure to construct shape functions for variable p-order meshes. *Comp. Meth. In Applied Mechanics and Engng.*, 147:209-233, 1997.

[SheGeo92] M.S. Shephard and M.K. Georges, “Reliability of automatic 3-D mesh generation”, *Comp. Meth. Appl. Mech. Engng*., 101:443-462, 1992.

[SheOba00] M.S. Shephard, R.M. O’Bara, R.M., X. Luo and X. Li, “Algorithm to generate p-version meshes for curved domains”, p and hp Finite Element Methods: Mathematics and Engineering Practice, Washington University, St. Louis, MO, pp. 103-104, 2000.

[SheWan01] M.S. Shephard, J. Wan and X. Li, “Mesh Enrichment Using Local Mesh Modification”, to appear Sixth U.S. Nat. Cong. on Computational Mechanics, U.S. Association for Computational Mechanics, 2001.

[SchRut99a] D. P. Schmidt, C. J. Rutland, and M. L. Corradini. Cavitation in two-dimensional asymmetric nozzles. SAE 1999-01-0518, February 1999. SAE International Congress.

[SchRut99] D. P. Schmidt, C. J. Rutland, and M. L. Corradini. A fully compressible, two-dimensional, model of small, high speed cavitating nozzles. *Atomization Spray*, 9(3):255--276, 1999.

[Tau01] T. J. Tautges, CGM: a Geometry Interface for mesh generation, analysis and other applications, to appear, *Engineering with Computers*, 2001.

[Tau01b] Timothy J. Tautges, "The generation of hexahedral meshes for assembly geometry: survey and progress", *Int. J. Numer. Meth. Engng*, 50:2617-2642 (2001).

[TB90] T.Blacker and M. Stephenson, Paving: a new approach to automated quadrilateral mesh generation, *SAND90-0249*, Sandia National Laboratories, 1990.

[TerBea00] J. D. Teresco, M. W. Beall, J. E. Flaherty and M. S. Shephard. A hierarchical partition model for adaptive finite element computations. *Comp. Meth. Appl. Mech. Engng.*, 184(2-4):269-285, 2000.

[ThoSoni99] J.F. Thompson, B.K. Soni and N.P. Weatherill, (Eds), *Handbook of Grid Generation*, CRC Press, 1999.

[Tre00a] Trease, H.E., Trease, L.L., NWGrid: A Multi-Dimensional, Hybrid, Unstructured, Parallel Mesh Generation System, http://www.emsl.pnl.gov/nwgrid, 2000.

[Tre00b] Trease, H.E., Trease, L.L., NWPhys: A Multi-Dimensional, Hybrid, Unstructured, Parallel Computational Simulation Framework, http://www.emsl.pnl.gov:2080/nwphys, 2000

[Tre85] Trease, H.E., Three-Dimensional Free Lagrangian Hydrodynamics, Proceedings of the first Free-Lagrange Conference, Lecture Notes in Physics, Springer-Verlag, Vol. 238, pp. 145-157, 1985.

[Tre90] Trease, H.E. and Dean, S.H., Thermal Diffusion in the X-7 Three-Dimensional Code, Proceedings of the Next Free-Lagrange Conference, Jackson Lake Lodge, Wyoming, Springer-Verlag Press, Vol. 395, pp. 193-202, June 3-7, 1990.

[Tre96] Trease, H.E., Denise George, Carl Gable, John Fowler, Eldon Linnebur, Andrew Kuprat, Ahmed Khamyaseh, The X3D Grid Generation System, *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Mississippi State University, April 1996, edit by B. Soni, J. Thompson, P. Eiesman and J. Hauser, ERC-MSU Press, 1996, ISBN 0-9651627-0-2, pp. 239-244.

[TufFis99] H. M. Tufo and P. F. Fischer, Terascale Spectral Element Algorithms and Implementations, *in Proc. of the ACM/IEEE SC99 Conf. on High Performance Networking and Computing*. IEEE Computer Soc., CDROM (1999). http://www.sc99.org/proceedings/tufobell.pdf

[Wa99] Wagner, J., A. Bugrim, J. Keizer, R. Fontanilla, R. Nuccitelli, “Quantitative studies of the fertilization Ca2+ wave initiation mechanism in Xenopus laevis eggs,” presented at the Biophysical Society annual meeting, Baltimore, MD, 1999.

[Wal00] Walshaw, C., and Cross, M., Parallel Optimisation Algorithms for Multilevel Mesh Partitioning, Parallel Comput., 26(12):1635-1660, 2000.

[WalYu00] J. Walter, D. Yu, B. Plohr, J. Glimm, and J. Grove. An algorithm for Eulerian front tracking for solid deformation. Report No. SUNYSB-AMS-00-25, State University of New York at Stony Brook, 2000.

[Zyv96] Zyvoloski, G., Auer, L., Dendy, J., Eggert, K., Gable, C., Kelkar, S., Kendall, R., Killough J., Kocberber, S., Trease, H., High Performance Computing for Domestic Petroleum Reservoir Simulation, LA-UR-96-1769.