



Institute for Scientific Computing Research



# ISCR Subcontract Research Summaries



# Self-correcting Algebraic Multigrid

Timothy P. Chartier,  
University of Washington

## Summary

---

**W**e report on the research and development of spectral AMGe (sAMGe), including:

- 1) enhancing the operator dependent decision-making process in the algebraic agglomeration routine developed by LLNL's Jim Jones and Panayot Vassilevski,
- 2) researching less computationally costly algorithms based on sAMGe that do not demand the use of spectral decomposition to determine smoothness, and
- 3) generating code to facilitate numerical experiments that would compare the efficacy of the sAMGe algorithms.

We are interested in element-free AMGe which does not demand access to the element matrices. Instead, the algorithm requires spatial coordinates for each degree of freedom. Then, polynomials would serve as initial guesses to globally smooth vectors. After a few relaxation steps, such polynomials would satisfy boundary conditions and become estimates for globally smooth error in sAMGe. Though requiring certain assumptions on smoothness, such an algorithm could harness sAMGe's ability to choose coarse grids and interpolation simultaneously while at the same time considerably reducing the computational cost of the algorithm.

The method produced impressive 2-level multigrid results for scalar elliptic problems. However, the multilevel tests were much less satisfying. Chow and Chartier continued discussions on this area of research throughout the summer and results of this work led to advancements in self-correcting methods detailed below. Still, further investigation is needed in order to determine if changes in the algorithm might produce multilevel results similar to those in the 2-level tests.

We began researching algorithmic development of self-correcting algebraic multigrid methods that use the sAMGe framework. Such methods automatically and adaptively correct their multigrid components to accelerate convergence.

As with sAMGe, such algorithms generalize the sense of smoothness. Initially, the work concentrated on adaptively correcting the columns of interpolation needed for coarse-grid correction in multigrid cycling.

Chow's research led to an alternate self-correcting algebraic multigrid scheme that creates additional V-cycles to dampen unresolved error components. In this way, convergence is accelerated adaptively within the method. We implemented this idea within the sAMGe framework and obtained excellent 2-level results possessing invariance of

*Summary (continued)*

---

convergence factors to diagonal scaling of the discrete operator – an important feature of self-correcting methods that is not found in most AMG methods. We also completed an initial multilevel version of the algorithm. The preliminary results look promising as to the efficacy of the method. Further enhancement of the algorithm should further improve these results. Testing the self-correcting ideas on systems of partial differential operators is an important step in the research and should be the next focus of the work.

On the matter of conducting numerical experiments to compare the efficacy of the sAMGe algorithms, Chartier completed code that reformats input data for the sAMGe code developed for his dissertation into a format suitable for the sAMGe code currently utilized at LLNL. This work will facilitate needed testing between the two sAMGe methods on a variety of problems.

sAMGe is a notable development in algebraic multigrid research. The promising results from the work of this subcontract bode well for the ability of sAMGe research to widen the problems efficiently solved by algebraic multigrid methods. Continued research is needed; yet important advances have been made as a part of this research.

# Morse Complexes and Visualization of Complex Topology Phenomena

**Herbert Edelsbrunner**

Duke University

## Summary

---

**M**y LLNL sabbatical was divided between pursuing research on Morse complexes and learning about specific application problems at the laboratory. The research on Morse complexes and related topics progressed nicely. A non-exhaustive list of topics researched is

- The loop structure in Reeb graphs
- A simplification hierarchy of Reeb graphs
- An algorithm for hierarchical 3D Morse-Smale complexes
- The combination of topological and geometric simplification
- Jacobian sub manifolds of multiple Morse functions
- A definition of the similarity between continuous functions

We worked on all these problems as a group, always including Valerio Pascucci, Kree Cole-McLaughlin, Vijay Natarajan, and myself. For a week during my sabbatical, we regularly met with John Harer, mathematician (and vice-provost) at Duke University, working on the same set of problems. Topological and geometric simplification work also involved Timo Bremer, student at UC Davis. To different degrees, the projects are on going and require further mathematical work, the development of algorithms, the design of visualization tools, and implementation work.

I also spent time with other researchers at LLNL discussing application problems for which we feel that our theoretical work could lead to new insights and developments. A non-exhaustive list of applications in this category is

- The study of breaking copper wires on an atomic level
- The behavior of gas under extreme pressure
- The state transition problem in computational chemistry
- The simulation of the heat chamber in atomic fusion
- The mixing of two liquids after a passing shock wave

In each case, we observe rather complicated topological phenomena developing over time. We made steps towards developing some of our ideas into software and to use one or two application problems in the above list as a case study to validate the tool, but primarily to serve the application research.

In summary, my sabbatical at LLNL was extremely productive for my own research. The visit has got the collaboration between Valerio Pascucci's and my own group off the ground, and I look forward to a continuation of that collaboration in the near and further future.

# Enabling Communication between Components for Scientific Computing

**Professor Rod Fatoohi**

San Jose State University

## *Summary*

---

**T**here has been a growing interest within the scientific computing community in component technologies. These technologies have shown their usefulness in the business community with examples like Microsoft's Component Object Model (COM), Sun's Java Beans, and OMG's CORBA Component Model (CCM). The Component Project at LLNL has developed a software tool called Babel that enables language interoperability among a variety of scientific programming languages, including Fortran, C, C++, Java, and Python. Babel enables the creation and distribution of software libraries specified by a Scientific Interface Definition Language (SIDL), which is similar to CORBA and COM IDLs but it focuses on the needs of scientific computing. Babel generates stubs and skeleton code, as in CORBA and others, from SIDL interfaces, but no communication code. Here we are developing the communication mechanism to enable invocation across multiple memory address spaces. Several communication protocols are considered including: ONC RPC, BSD Sockets, CORBA IIOp, and SOAP.

The goal of this project is to enable Babel users to make remote calls seamlessly, regardless to the location of the requested component. We are developing an abstract sub-layer underneath the Babel generated code to interface with multiple communication protocols. This sub-layer, similar to the abstract protocol layer of the Inter-Language Unification (ILU) system by Xerox, defines message types and parameter types between Babel runtime system and the communication protocols. The application programming interface for these protocols will be integrated with the Babel tool so that calls to them will be automatically generated by Babel.

Different communication protocols are considered including ONC RPC, BSD Sockets, CORBA IIOp, and SOAP. We have already specified how Babel data types are mapped into the ONC RPC wire protocol: eXternal Data Representation (XDR). Our next step is to integrate our interface with the Intermediate Object Representation (IOR) of Babel. We are also leveraging from the research activity at Indiana University in this area. We are aiming at implementing all four protocols (listed above), and give the user the option to select any specific protocol based on several criteria including performance, reliability, flexibility, scalability, and maintenance cost.

# Scalable Symmetric Eigenvalue Solver using *hypre*

Andrew Knyazev

University of Colorado, Denver

## Summary

---

The goal of this project was to develop Scalable Symmetric Eigenvalue Solver algorithm software for the solution of partial eigenvalue problems for large, sparse symmetric matrices on massively parallel computers, taking advantage of advances in the Scalable Linear Solvers project, in particular in multigrid technology and in incomplete factorizations (ILU) developed under the *hypre* project at the LLNL. This implementation allows the utilization of *hypre* preconditioners for symmetric eigenvalue problems, thus significantly extending their capability. LLNL plans to include this capability in the next *hypre* release.

The base iterative algorithm that has been implemented is the PI's locally optimal block preconditioned conjugate gradient (LOBPCG) method. The LOBPCG solver finds one or more of the smallest eigenvalues and eigenvectors of a positive definite matrix. The code is written in C and uses *hypre* and LAPACK/BLAS libraries. The user interface to the solver is implemented using *hypre* style object-oriented function calls. The matrix-vector multiply and the preconditioned solve are done through user supplied functions, which provide significant flexibility. The implementation illustrates that this matrix-free algorithm can be implemented successfully and efficiently using parallel libraries.

An LOBPCG driver was developed as part of the package, which exercises the software, and provides an implementation example of how to use the LOBPCG *hypre* user interface. It can be easily run after compilation on various platforms to test different configurations. This driver is patterned after a *hypre* driver that tests `IJ_linear_solvers`. It optionally generates several Laplacians and optionally reads a matrix from a matrix market file. The following preconditioners are available and have been tested: AMG-PCG; DS-PCG; ParaSails-PCG; Schwarz-PCG (default); and Euclid-PCG. Partition of processors is determined by user input consisting of an array of parallel vectors. The testing was done on a variety of platforms using up to 60 processors.

LOBPCG has been mainly developed and tested on the Beowulf cluster at the Colorado University at Denver. This system includes 36 nodes, 2 processors per node, PIII 933MHz processors, 2GB memory per node running Linux Redhat, and a 7.2SCI Dolpin interconnect. LOBPCG has been compiled and tested on a subset of the OCF Production Systems at LLNL as well.

Source code and documentation were delivered to LLNL and a seminar on their use presented.

# Non-conforming Finite Elements, Mesh Generation, Adaptivity and Related Algebraic Multigrid and Domain Decomposition Methods in Massively Parallel Computing Environment

Raytcho Lazarov and  
Joseph Pasciak

Texas A & M University

## Summary

---

Construction, analysis, and numerical testing of efficient solution techniques for solving elliptic PDEs that allow for parallel implementation have been the focus of the research. A number of discretization and solution methods for solving second order elliptic problems that include mortar and penalty approximations and domain decomposition methods for finite elements and finite volumes have been investigated and analyzed. Techniques for parallel domain decomposition algorithms in the framework of PETSc and HYPRE have been studied and tested. Hierarchical parallel grid refinement and adaptive solution methods have been implemented and tested on various model problems. A parallel code implementing the mortar method with algebraically constructed multiplier spaces was developed.

The discretization methods of PDEs on non-matching grids provides greater flexibility in the grid generation process, increases the portability of various approximation methods and computer implementations, enhances the capabilities of coarsening strategy in parallel algebraic multigrid methods, and provides a natural and practical way for parallel domain decomposition methods and parallel adaptive methods based on a posteriori error analysis. Our research focused on the analysis, implementation, and testing of some discretization methods of PDEs on non-matching grids. This approach seems to lead to competitive algorithms that can be used in various codes for complex applications in physics and engineering.

In the area of domain decomposition methods for non-mortar approximations on non-matching grids, the interior penalty method aims at eliminating the need for additional (Lagrange multiplier or mortar) spaces and imposes (only approximately) the required continuity across the interfaces by an appropriate penalty term. In our approach, the jumps in the values of the functions along these interfaces are penalized in the variational formulation. For smooth solutions we lose the optimal accuracy due to lower approximation at the interface, but on the other hand we produce symmetric and positive definite discrete problems, which have optimal condition number.

We also address the issue of constructing preconditioners for solving the system on the composite non-matching grids. We proposed and investigated an interface domain decomposition type preconditioner that is spectrally equivalent to the reduced (on the interface) algebraic problem. We have tested both the accuracy of the method and the preconditioning technique on a series of model problems. We were able to prove an almost optimal error estimate for the interior penalty approximation of the original problem based on the partition of the domain into a finite number of subdomains. Further, an improved error analysis for the finite element approximation of the penalty formulation was derived. Numerical experiments on a series of model second order problems were performed in order to test and verify computationally our theoretical findings.

*Summary (continued)*

---

In mortar approximations of finite element and finite volume methods on non-matching grids, we considered an algebraic extension of the local construction for the mortar multipliers based on the general 3-d dual finite element basis. The purely algebraic construction of the mortar interpolation in the present case requires the inversion of local mass matrices on the non-mortar interface. Since it is purely algebraic, it can be applied to the generalized objects (elements, faces and degrees of freedom) produced by the subdomain AMGe. A parallel code implementing the mortar method with algebraically constructed multiplier spaces was developed. The target application of this code is in the construction of parallel (including multigrid) preconditioners using element based (AMGe) coarsening in each subdomain. A general code was implemented to illustrate the behavior of the proposed method. It requires input data for each subdomain that includes the element topology, the local subdomain stiffness matrices as well as the mass matrices on the interfaces. This information is independent of the dimension and structure of the problem and is regenerated after an AMGe coarsening.

In multilevel adaptive grid refinement and error control, we worked on a problem of multilevel grid refinement and error control for both finite volume approximations and penalty domain decomposition methods. This work was in direct connection with research in CASC on developing and testing of parallel algorithms. We have developed 2-D and 3-D codes for parallel adaptive grid refinement that produces nested (and matching) grids. The new software was connected to the HyPre Preconditioner Library. A parallel mesh generation tool, called ParaGrid, was further developed. The development was a continuation of a 2-D project that was started last summer in CASC. ParaGrid is software that takes as input a coarse tetrahedral mesh, which describes suitably the domain, splits it using METIS, distributes the partitioning among the available processors and generates, in parallel, a sequence of meshes. It has internal solvers and is able to generate various Finite Element/Volume discretizations. The data structures allow ParaGrid to be easily connected to (or used to provide data to) external parallel finite element/volume solvers based on domain decomposition. Generation and solution routines for elasticity problems were added to the code. HYPRE preconditioners and solvers can be used. The connection is done through FEI 3.0. It has been successfully used by several researchers in CASC for algorithm testing purposes.

The software was developed in close collaboration with our student S. Tomov from Texas A&M University and Dr. Charles Tong from CASC. It was used for testing various ideas and strategies in the a posteriori error analysis and error control for convection-diffusion-reaction problems in 3-D domains with complex structure.

# Bioinformatics Website Discovery and Analysis

Ling Liu

Georgia Institute of Technology

## Summary

---

This report documents: (1) wrapper technology aiming at incorporating more functionality to XWRAP toolkit to provide richer content extraction and high automation, (2) initial design of a dynamic web crawler, and (3) website discovery and analysis which is built on top of (1) and (2).

Automatically extracting data from Web sites is recognized as an important task in many Web services and the Semantic Web in general. Most advances in Web information extraction are semi-automated or manual approaches, which require hand-coding information extraction logic and semantics. In addition, most existing extraction tools and wrapper generation tools limit their abilities to extracting information from single pages. They cannot extract related information from multiple pages or extracting information by following links. One example application that requires multi-page information extraction and more sophisticated extraction logics is the SciDAC SDM data integration pilot scenario .

We have observed that wrappers generated by *wrapper generators*, such as XWRAP Original or XWRAP Elite, usually perform well when extracting information from individual documents (single web pages) but are poorly equipped to extract information from multiple linked Web documents. An obvious reason for the inefficiency is due to the lack of system-wide support in the wrapper generators. For example, a wrapper in XWRAP Elite or Original can only perform information extraction over one type of Web document, while typical bioinformatics sources use several types of pages to present their information, including HTML search forms and navigation pages, summary pages for search results, and strictly formatted pages for detailed search results.

Information extraction over multiple different pages imposes new challenges for wrapper generation systems due to the varying correlation of the pages involved. The correlation can be either horizontal when grouping data from homogeneous documents (such as multiple result pages from a single search) or vertical when joining data from heterogeneous but related documents (a series of pages containing information about a specific topic). Furthermore, the correlation can be extended into a graph of workflows. A multi-page wrapper not only enriches the capability of wrappers to extract information of interests but also increases the sophistication of wrapper code generation.

*XWRAP Composer* is a semi-automated wrapper generation system that generates wrappers capable of extracting information from multiple heterogeneous Web documents. The XWRAP Composer script usually contains three types of root commands, *document retrieval*, *data extraction* and *post processing*. The document retrieval commands construct a file request or an HTTP request and fetch the document. The data extraction commands extract information from the fetched document. The post processing commands allow adding semantic filters to make the output conform to the outeface specification.

---

*Summary (continued)*

---

One design approach to Web site discovery and analysis is to use Crawler and Service Class Description. The Crawler and Service Class Description framework provides core service discovery, classification, and integration services. The heart of this system is the service class description, which outlines the relevant aspects of a service. The description provides a mechanism for encapsulating the important components of a particular service that are common to all members of the class and is the mechanism for hiding insignificant differences between individual services. We expect that service class descriptions will be crafted by scientists interested in utilizing automatic service discovery for their own application areas, so discrimination of important or significant service aspects is left to the description writer. Indeed, since Web services may fall into several classes, aspects of a service that are important to one service class may be irrelevant in another. The service class description framework has been designed with these considerations in mind and allows service class descriptions to be tailored to the particular needs of the application domain.

Service classes are described with an XML document consisting of three main components: the data types used by the service, example queries and output, and a control flow graph representing how types can interact. The type definitions describe the atomic and complex types needed by members of the service class. The service class type system is modeled after the XML Schema type system for compatibility and provides the mechanism for declaring the data types that will be needed when processing a site. For example, a simplified view of the DNA sequence BLAST service class includes a DNA sequence input type, a DNA BLAST result output type.

The second section of a service class description enumerates the expected navigational paths used by members of the service class. Nodes in the graph represent control states; these are typically pages encountered while interacting with the site, such as the query entry page; entry and exit points to the graph are stated explicitly and represent a service's interface. Edges are directed and represent possible execution paths between control states. A BLAST control flow graph will include states for the input, summary, and results as well as control features like indirection and delay pages.

The final component of the service class description is the example queries. The crawler uses these examples to test a site and see if it produces reasonable results when queried. An input example can be used to check if a site accepts input as required by the service class. The examples may also include negative or null queries that are expected to produce no results. Negative queries are useful for both validation purposes and for determining what parts of a result page are data.

The crawler provides the discovery and integration mechanics. Starting from a root set of relevant sources, the crawler searches the Web for sites providing a Web service--- e.g. those sites that support Web forms. When it finds a site, the Web crawler uses the

*Summary (continued)*

---

service class description to analyze the site; if it matches the description, we say that the site is determined to be a member of the service class described.

After classifying a site as a service class member, the crawler generates a wrapper description that maps the general concepts described in the service class description to the implementation details of the particular site. Once generated, the XWrap wrapper-generation system processes the wrapper description and produces a Java wrapper for the site. The XWrap system will also generate a Java interface definition from the service class description, which is used by the mediation and user interface subsystem and allows all members of a particular service class to be queried via a common interface.

Our crawler uses the three components of the service class description described above to both identify a site as a class member and to generate a description specific to the individual site. Once a site has been identified as a member of the service class, the system must produce a wrapper that can be used by applications interested in interacting with members of the service class. Such applications expect a single, uniform interface to all members of the class, so every generated wrapper must support the same interface signature despite differences between sites with respect to input parameter names, data values, supported features, process flow, etc. The wrapper must be able to reconcile the application's expected input parameters with the nuances of the wrapped site and, once the site returns an answer, the wrapper must reverse the process by transforming the result into the expected output format. Finally, the wrapper needs to handle exception conditions gracefully.

Some future plans for XWRAP Composer continues can be described from three aspects. First, we will finish the design of XWRAP Composer scripting language. Second, we will develop the first prototype of XWRAP Composer system. Particularly, we need to finish the script compiler that generates the wrapper code. Third, we plan to design an evaluation metrics and a set of experiments to provide experimental analysis of the performance of XWRAP Composer. Most of this work will be conducted under the SciDAC project.

# Enhancing the TBEPI Electronic Structure Code

Professor Calvin Ribbens

Virginia Polytechnic Institute

## Summary

---

**W**e have collaborated with Patrice Turchi and Ben Torralva from the Materials Science and Technology Division of CMS at LLNL to support the development large-scale codes for simulating the electrical and transport properties of metallic alloys, by developing parallel algorithms and implementations for these codes, so that LLNL parallel computing resources can be used to tackle much larger problem sizes, yielding increased accuracy in the simulations.

The recent focus has been on enhancing the modeling capabilities and the performance of the *TBEPI* electronic structure code. The goal was to develop one code that will execute efficiently on a wide range of high-end machines, including ASCI Blue-Pacific, ASCI White, and the Compaq and Linux clusters. Significant progress was made in three areas:

Incorporation of new physics into the model. The so-called “off-diagonal disorder” case was included in the parallel simulation code. This required a nontrivial modification to the communication strategy used in a major phase of the computation.

Improvement of memory-scaling. Previous versions of the parallel code required each thread to have a workspace proportional to  $nthreads * natoms$ , where  $nthreads$  is the number of threads used in each process and  $natoms$  is the total number of atoms in the simulation. The implementation strategy was modified so that each thread only needs memory proportional to the number of atoms assigned to that thread. Previously, memory constraints became a problem on ASCI Blue for problem sizes larger than about 2000 atoms. With this modification we are now able to do simulations with over 5000 atoms.

Allowing variable number of threads per process. Initial parallel versions of the code were developed for the ASCI Blue-Pacific machine, which has four processors per compute node. Hence, the implementation was tuned to perform significantly better with four threads per process. The parallelization strategy has now been changed so that essentially any number of threads can be used effectively. This will be important for future simulation runs on ASCI White, for example, where each compute node has 16 processors.

The final version of *TBEPI* is now in production use by Turchi, Torralva, and colleagues in CMS. A journal article is in preparation, reporting on parallel scalability up to 732 processors on ASCI Blue.

# Spectral AMGe, ALE3D, and the FOSPACK Codes

**John W. Ruge**

Front Range Scientific Computations, Inc.

## Summary

---

In this project, we extended the FO SPACK code and applied it to problems of elasticity. Our first task was to extend the 3D code to allow for general trilinear functions over arbitrary hexahedral meshes. The internal mesh generation capability, while limited to logically rectangular hexahedral meshes, was enhanced. The capability for reading TrueGrid files and converting the data to the internal FO SPACK data structures was added. In addition, the capability was added to be able to easily modify grids, mark subdomains, and mark domain boundaries, whether the grids were generated internally or read from files. This latter part is important since FOSLS formulations require explicit addition of boundary conditions even on free boundaries where no conditions are needed in more standard formulations.

An option was added for “strong” handling of boundary conditions (imposing them on the finite element space) in addition to the existing “weak” handling, where they are incorporated as functional terms (similar to a penalty method). A related task was the treatment of boundaries as true curved surfaces where necessary, rather than as piecewise bilinear element surfaces. This was found to be important in 2D test problems. That is, the capability was added to find normal and tangential vectors to the true surface at a given node, rather than the normal or tangential vectors to the finite element surface.

The most extensive task was the modification of the code to allow for either full solution of the FOSLS problems by AMG, or a block-diagonal preconditioning for lowering storage requirements, extremely important in the solution of large 3D problems, since these requirements are much higher than standard formulations on the same mesh. This required partial matrix assembly, as opposed to the full assembly previously used. Previously, the full form of the matrix was determined before actual assembly (which unknowns connect to which ones and where, what form will it have after elimination of boundary conditions and slave nodes, etc.) This has been overhauled, and now the main connectivity (using only the desired connections from the internal equations is determined beforehand. Additional auxiliary storage is used for any desired entries that do not fit the predetermined form, and these connections are assembled into the matrix afterwards. This also allows for flexible handling of boundary conditions, and which other connections may be included in the preconditioner (for example, keeping connections between strongly coupled unknowns).

Such a preconditioning approach (either in a straightforward defect-correction approach or in conjunction with conjugate gradient acceleration) required addition of code for the computation of the full residual when the full matrix is not assembled. The needed routines were first implemented and tested in the 2D code, then extended to the 3D version.

*Summary (continued)*

---

Some modifications to the AMG solver were introduced to reduce the storage and computational requirements, particularly for large 3D problems. These included the use of long range interpolation, which allows for faster coarsening.

For FOSLS approaches, especially in 3D, storage can be a concern, since 12 unknowns (9 when only the first stage is solved) are required. This, combined with increased complexity of the current AMG algorithm in 3D, can require much more storage than standard finite element methods. The use of partial matrix assembly as noted above can cut down storage requirements by a factor of 3-4 (with some loss of linear convergence) while modifications to the AMG coarsening can cut complexity (and thus storage requirements) by another factor of 2-3.

Many numerical tests were performed in both 2D and 3D, with comparisons made between FOSLS and standard Galerkin formulations. While standard Galerkin finite elements appear to be more efficient for “nice” problems, FOSLS formulations allow for solution of elasticity problems approaching the incompressible limit with no degradation of convergence, while linear solvers for the standard approaches degrade badly. FOSLL\* formulations are also useful in the presence of singularities, while providing direct approximations to the stresses.

# Numerical Methods and Studies of High-Speed Reactive and Non-reactive Flow

Donald W. Schwendeman

Rensselaer Polytechnic Institute

## Summary

---

The work carried out under this subcontract involved the development and use of an adaptive numerical method for the accurate calculation of high-speed reactive flows on overlapping grids. The flow is modeled by the reactive Euler equations with an assumed equation of state and with various reaction rate models. A numerical method has been developed to solve the nonlinear hyperbolic partial differential equations in the model. The method uses an unsplit, shock-capturing scheme, and uses a Godunov-type scheme to compute fluxes and a Runge–Kutta error control scheme to compute the source term modeling the chemical reactions. An adaptive mesh refinement (AMR) scheme has been implemented in order to locally increase grid resolution. The numerical method uses composite overlapping grids to handle complex flow geometries. The code is part of the Overture–OverBlown framework of object-oriented codes, and the development has occurred in close collaboration with Bill Henshaw and David Brown, and other members of the Overture team within CASC.

During the period of this subcontract, a number of tasks were accomplished, including

- an extension of the numerical method to handle “ignition and grow” reaction models and a JWL equations of state
- an improvement in the efficiency of the AMR scheme and the error estimator
- an addition of a scheme of numerical dissipation designed to suppress numerical oscillations/instabilities near expanding detonations and along grid overlaps
- an exploration of the evolution to detonation in an annulus and of detonation failure in an expanding channel

Our method has been extended to handle an ignition and growth model of reactive flow. In this model, the explosive mixture consists of two components, a reactant and a product. Each component is assigned its own equation of state, typically of JWL form, from which is constructed a mixture equation of state under certain closure assumptions such as pressure and temperature equilibrium between the two components. A single variable measures the progress of the reaction. Unlike a simple one-step Arrhenius reaction rate that was implemented in the original code, the ignition and growth model considers a rate function that undergoes sudden changes in form as the progress variable crosses certain values (although the overall rate function remains continuous). The model involves many parameters that require experimental calibration, but our aim is the development of the numerical method to handle the model and the mathematical issues surrounding the model itself. An added difficulty in the implementation of the model is that the assumed equilibrium between states must be maintained throughout the calculation. This is done in a new separate section of code that is designed to handle mixture equations of state.

In a previous subcontract, an automatic mesh refinement (AMR) scheme was implemented in the code. The scheme is a patch-type refinement scheme similar in flavor to that developed originally by Berger. For a particular domain, a composite overlapping

---

*Summary (continued)*

---

grid is generated using Ogen and this grid becomes the base grid on which AMR grids may be built. As the solution evolves, AMR grids are created, modified, or removed based on an error estimate of the solution. The error estimate is based on first and second differences of the solution, as was done in the first implementation of the method, and on a measure of the error due to the source term that has been improved. The method handles the source modeling the chemical reactions using a Runge–Kutta error control scheme. (There are separate chemistry time steps in the method corresponding to an integration of a set of ordinary differential equations.) The error-control scheme estimates the truncation error and this is now used in the AMR error estimator in order to determine whether a grid cell should be tagged for refinement based on the fast scales due to the chemical reaction model.

In addition to improvements in the way the source term contributes to the error estimate for AMR, the AMR grids themselves are now handled more efficiently in the implementation of the method. These improvements involve, in particular, better memory management and more efficient interpolation. This has led to a significant improvement in the computational time spent managing the AMR grids in a typical reactive flow calculation.

A scheme of numerical dissipation is now included. This was done to suppress numerical oscillations/instabilities that have been observed in various reactive flow calculations. A typical case in which the oscillations were observed involves detonation diffraction. If the detonation is expanding along a wall, numerical oscillations in the state variables (in the density for example) occur in the direction normal to the wall. For problems of this type, there is very little numerical dissipation provided by the calculation of the numerical fluxes in the Godunov scheme so that an additional scheme of numerical dissipation was considered to be appropriate. A similar problem has been observed near shocks in non-reactive flows. The problem is worse for reactive flows where small numerical oscillation can lead to large errors in the solution due to the state-sensitivity of the solution. Numerical oscillations have also been observed for problems in which a detonation crosses a grid overlap, and our scheme of numerical dissipation helps in these situations as well.

Our code has been used to study a number of problems involving high-speed reactive flow, including evolution to detonation in an annulus and detonation diffraction and failure in expanding geometries.

# Numerical Methods for Electromagnetic Radiation and Scattering Problems

**Karl F. Warnick**

Brigham Young University

## *Summary*

---

The purpose of this project is to develop a multilevel fast multipole algorithm (MLFMA) implementation that can be integrated into LLNL's EIGER electromagnetic radiation and scattering solver, in order to increase the computational efficiency of numerical simulation of electromagnetics problems. For large and complex problems, hundreds of thousands or millions of unknowns can be required. Since the boundary element method (BEM) used by EIGER leads to dense matrices, filling and solving such large systems is not feasible. MLFMA reduces the computational complexity of BEM from at least  $O(N^2)$  to  $O(N \log N)$ , making it competitive with sparse-matrix methods and dramatically increasing the range of problems that can be solved.

Over the period of this project, an MLFMA test code was developed. A derived version of this code will be integrated into EIGER's parallel, object-oriented framework. From a research point of view, MLFMA is fairly well understood and has been implemented in commercial codes. The accuracy and efficiency of the algorithm are strongly sensitive to physics-dependent parameters, and many aspects of low-level implementation required to obtain a working algorithm have not been published. Part of the effort of this project was devoted to working through these implementation details so that a more complete, tutorial-level technical document on the method can be produced.

During the development of the MLFMA code, we addressed several method and algorithm issues. MLFMA requires the segmentation of a mesh into a hierarchical, multi-grid-like group structure. We implemented the standard oct-tree approach and an irregular method that equalizes the number of scatterers in each group. The underlying multipole expansion on which MLFMA is based is a divergent series, so the algorithm is sensitive to parameter choices and numerical precision. We found, for example, that some commonly used special function packages are not sufficiently accurate for MLFMA. The algorithm also requires interpolation on a sphere, which is another source of numerical error. We compared simple bilinear interpolation with tensor product Lagrange interpolation, and found that the less accurate scheme fails. Special care was given in examining how the interpolation methods should handle points clustered at the poles of the sphere, where the coordinate system is singular.