SciDAC
Scientific Discovery through Advanced Computing

FASTMath SciDAC Institute Overview

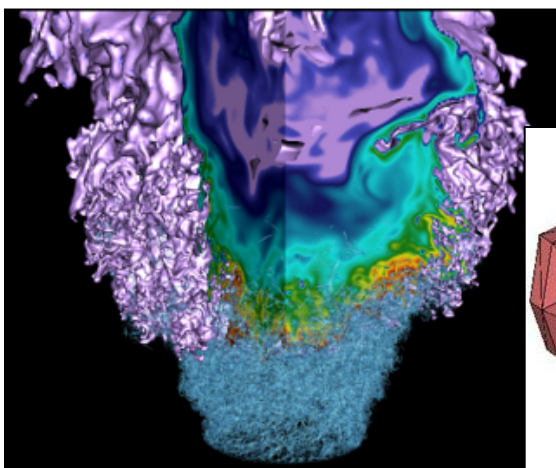
FASTMath Team

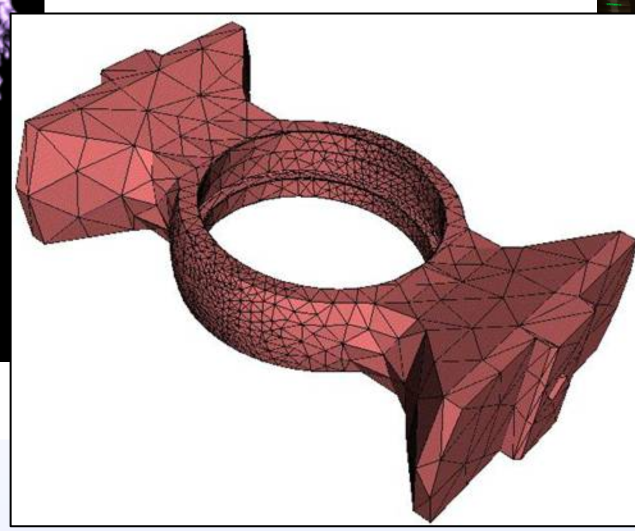
Lori Diachin, Institute Director

FASTMath SciDAC Institute

FASTMath Objectives

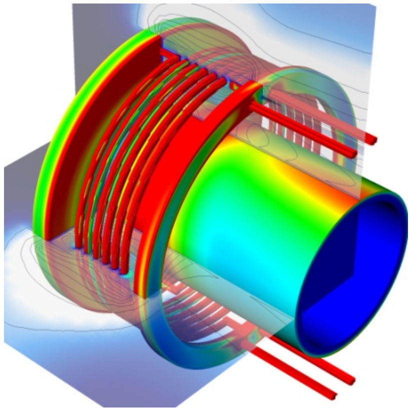
The FASTMath SciDAC Institute will develop and deploy scalable mathematical algorithms and software tools for reliable simulation of complex physical phenomena and will collaborate with DOE domain scientists to ensure the usefulness and applicability of FASTMath technologies





FASTMath will help application scientists overcome two fundamental challenges

1. Improve the quality of their simulations
   - Increase accuracy
   - Increase physical fidelity
   - Improve robustness and reliability



2. Adapt computations to make effective use of LCFs
   - Million way parallelism
   - Multi-/many-core nodes

FASTMath will help address both challenges by focusing on the interactions among mathematical algorithms, software design, and computer architectures

The FASTMath team includes experts from four national laboratories and six universities

Lawrence Berkeley National Laboratory

- Ann Almgren
- John Bell
- Phil Colella
- Dan Graves
- Sherry Li
- Terry Ligocki
- Mike Lijewski
- Peter McCorquodale
- Esmond Ng
- Brian Van Straalen
- Chao Yang

Sandia National Laboratories

- Karen Devine
- Jonathan Hu
- Vitus Leung
- Andrew Salinger

Lawrence Livermore National Laboratory

- Lori Diachin
- Milo Dorr
- Rob Falgout
- Jeff Hittinger
- Mark Miller
- Carol Woodward
- Ulrike Yang

Berkeley University

- Jim Demmel

University of British Columbia

- Carl Ollivier-Gooch

Columbia University

- Mark Adams

Argonne National Laboratory

- Mihai Anitescu
- Lois Curfman McInnes
- Todd Munson
- Barry Smith
- Tim Tautges

Rensselear Polytechnic Institute

- Mark Shephard
- Onkar Sahni

Southern Methodist University

- Dan Reynolds

Colorado University at Boulder

- Ken Jansen

FASTMath encompasses three broad topical areas

Tools for problem discretization	Solution of algebraic systems	High-level integrated capabilities
▪ Structured grid technologies ▪ Unstructured grid technologies ▪ Adaptive mesh refinement ▪ Complex geometry ▪ High-order discretizations ▪ Particle methods ▪ Time integration	▪ Iterative solution of linear systems ▪ Direct solution of linear systems ▪ Nonlinear systems ▪ Eigensystems ▪ Differential Variational Inequalities	▪ Adaptivity through the software stack ▪ Management of field data ▪ Coupling different physical domains

FASTMath brings a spectrum of software tools in these areas to the SciDAC Program

FASTMath Toolset

Geometry Tools
- CGM (Tim Tautges)

Structured Mesh Tools
- BoxLib (John Bell)
- Chombo (Phil Colella)

Linear Solvers
- Hypre (Rob Falgout)
- PETSc (Barry Smith)
- SuperLU (Sherry Li)
- ML/Trilinos (Jonathan Hu)

Unstructured Mesh Tools
- MeshAdapt (Mark Shephard)
- MOAB (Tim Tautges)
- Mesquite (Lori Diachin)
- PHASTA (Ken Jansen)

Nonlinear Solvers/Differential Variational Inequalities
- PETSc (Barry Smith)
- NOX/Trilinos (Andy Salinger)

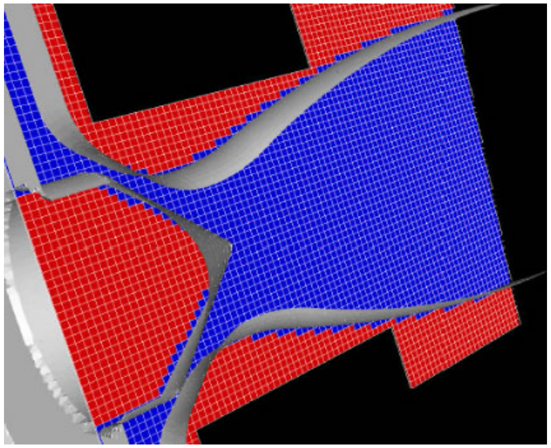
Partitioning Tools
- Zoltan (Karen Devine)

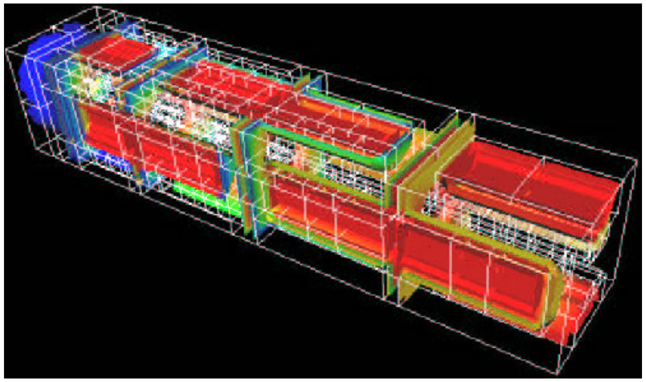
Eigensolvers
- PARPACK (Chao Yang)

Time Integrators
- SUNDIALS (Carol Woodward)
- PETSc (Barry Smith)

Structured Mesh Discretization Tools

- **Goal:** Provide tools and frameworks for efficient solution of a large variety of problems in science and engineering using block structured mesh approaches
- **Approach/Technical Details**
  - Block structured adaptive mesh refinement for continuum PDEs and particle methods
  - Volume of fluid methods and mapped multi-block techniques for complex geometry
  - Massively parallel implementations used in a variety of applications
- **Applications**: accelerator modeling, astrophysics, combustion, porous media, CFD
- **Associated Tools**: BoxLib, Chombo
- **For more information**: John Bell (jbbell@lbl.gov) and Phil Colella (pcolella@lbl.gov)



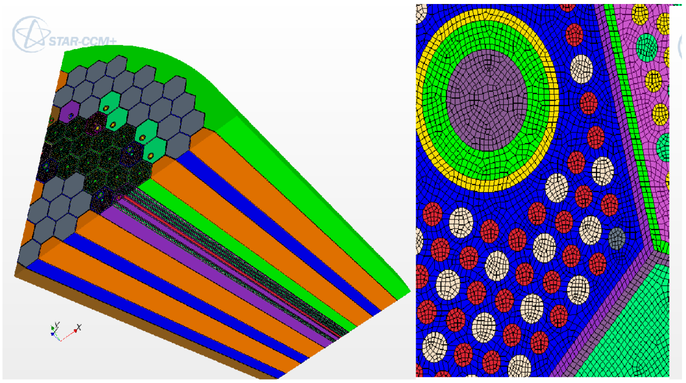


FASTMath will continue the development of structured grid technologies

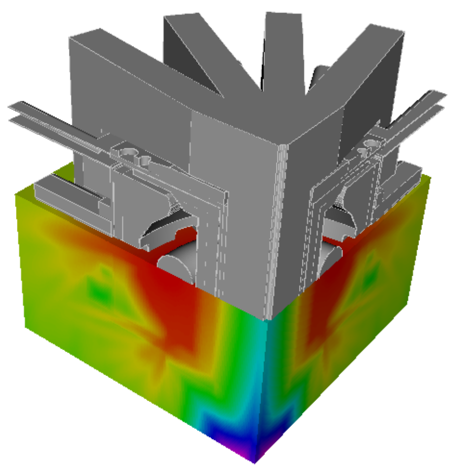
- **Structured adaptive mesh refinement**
  - Coupling to linear solvers
  - Node-level parallelism and scalability for explicit methods and pointwise physics
    - Challenge of achieving reuse of software while maximizing data locality
  - High order (>4th) finite volume methods
  - New solver capabilities for Vlasov-Maxwell and Vlasov-Poisson
- **Mapped multi-block methods**
  - Improved performance of ghost cell values at block boundaries for scalability
  - Support for anisotropy; particularly in AMR setting
- **Volume of fluid approaches**
  - Dynamic load balancing to improve scalability
  - Scalable multicomponent capabilities for fixed or moving boundaries and jump relations imposed at the boundaries
  - High order discretizations at irregular boundaries

Tools for representing complex geometrical domain and unstructured meshes

- **Goal**: provide robust components to support all phases of parallel mesh-based simulations
- **Approach/Technical Details**
  - **Common Geometry Module (CGM)**
    - Query/modification of CAD geometry
    - Supports ACIS, Open.CASCADE, SolidWorks, Catia, Pro/E engines
  - **Unstructured Mesh Management Tools:**
    - **MOAB:** Represents mixed structured/ unstructured mesh and data on the mesh
    - **FMDB:** unstructured mesh implementation focused on evolving meshes
    - **GRUMMP:** unstructured mesh implementation focused on efficient edge based computaions
  - **Lasso:** Library for querying geometry-mesh associativity
- **For more information**: Tim Tautges (ANL), Mark Shephard (RPI), Carl Ollivier-Gooch (UBC)



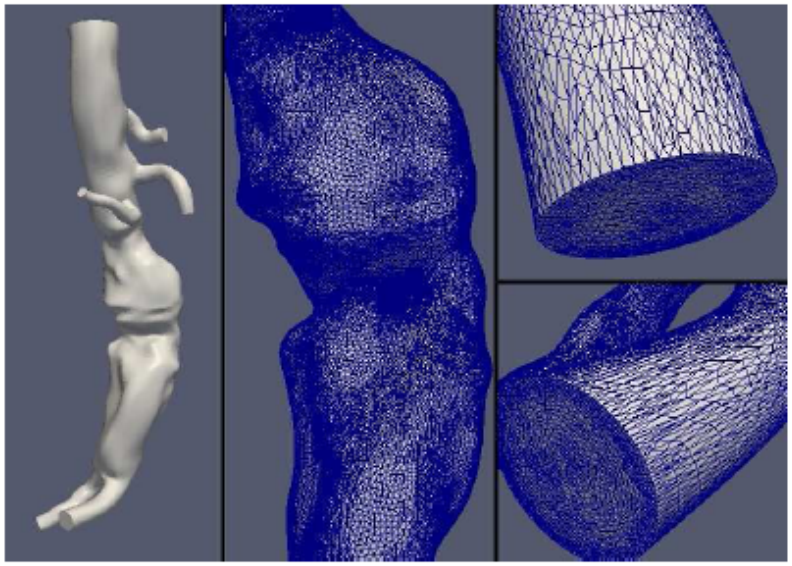
VHTR Core, 5k volumes, 53M hexes



Geometry/mesh, FDF Experiment

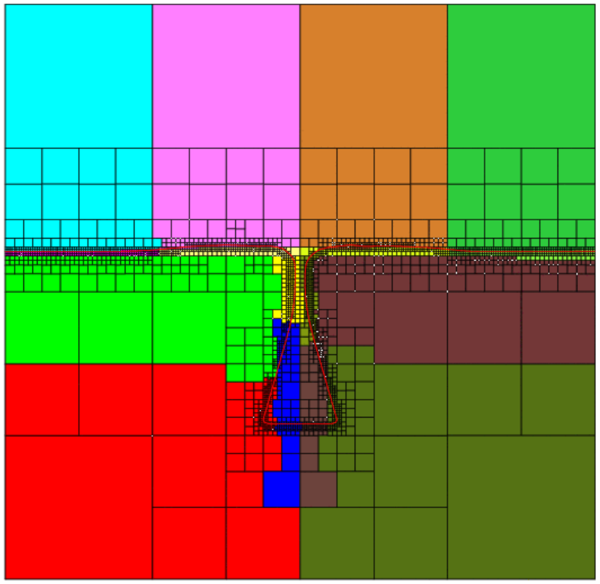
Parallel Adaptive Unstructured Mesh Refinement

- **Goal/Objectives of the technology**: Provide tools to support parallel adaptive simulations
- **Approach/Technical :**Tools for parallel mesh modifications with dynamic load balancing
- **FASTMath Components**:
  - MeshAdapt – parallel mesh adaptation based on anisotropic mesh size field
  - iField – to support local solution transfer during mesh adaptation
  - Predict – predictive load balancing
  - IPComMan – message packing taking advantage of neighborhoods



- **Applications**: CFD, Electromagnetics, MHD, solid mechanics and heat transfer
- **For more information**: Mark Shephard, shephard@scorec.rpi.edu

Parallel Partitioning and Dynamic Load Balancing



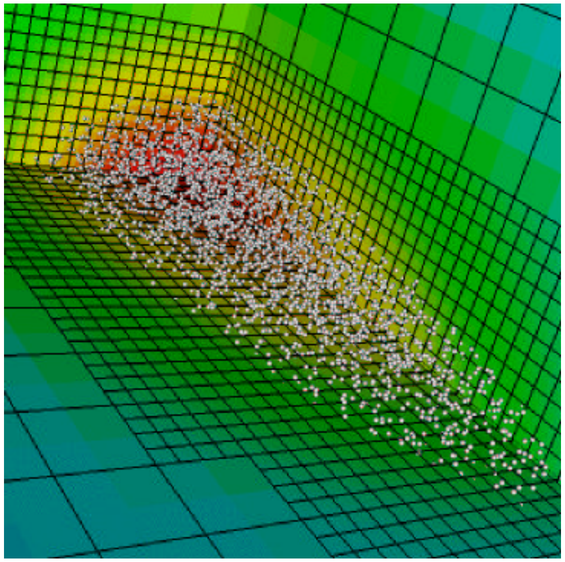
- **Goal/Objectives of the technology**: Balanced, low-communication distributions of work to processing cores
- **Approach/Technical Details:** geometric, graph-based, and hypergraph-based algorithms supporting a wide range of applications
- **Components**:
  - Zoltan: Suite of parallel partitioning tools, as well as data ordering and coloring algorithms
  - ParMA: Iterative partition improvement demonstrated to improve performance of multi-phase mesh-based applications
  - iZoltan: ITAPS-Compliant interfaces for mesh partitioning via Zoltan
- **Applications**: Adaptive finite element methods, particle-in-cell methods, crash and contact detection, linear solvers and preconditioners, circuit networks
- **For more information**: Karen Devine, kddevin@sandia.gov

FASTMath will focus on three areas for continued unstructured grid technology development

- **High Order Discretizations**
  - Generation, control, adaptation of curved mesh entities
  - A posteriori measures for optimal anisotropic mesh refinement
- **Dynamic Partitioning Strategies**
  - Development of hybrid connectivity/geometric based partitioners for improved scalability while maintaining quality
  - Fast, scalable partition improvement methods
  - Multi-criteria, incremental improvement methods
  - Partitioners that account for hierarchical architecture topology and data access costs
- **Unstructured grid calculations on LCFs**
  - Research blocking and data ordering effects for efficiency
  - Lightweight creation of data blocks for thread parallelism and accelerators, local reordering algorithms based on geometric and mesh-adjacency information
  - Algorithms with lightweight communications that avoid synchronization
  - Thread safe mesh adaptation and optimization
  - Parallel I/O based on synchronization and reduced blocking strategies

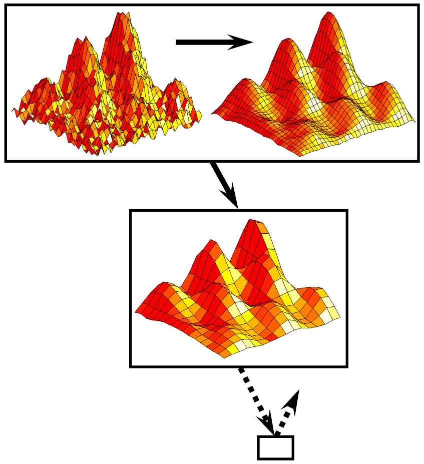
Particle Discretization Methods

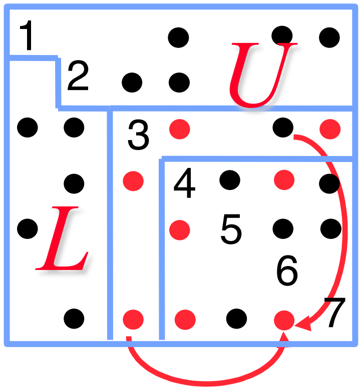
- **Goals:**
  - Efficient calculation of particle motions, velocities and forces
  - Mapping to a gridded representation in physical space for coupling for neighbor calculations
- **Technical Approach for FASTMath work:**
  - Provide a cross-cutting framework for particle calculations that addresses node-level parallelism and data locality
  - Provide capabilities to remap continuous solutions to a PDE represented by particles onto locally-refined grids
  - Provide particle to mesh coupling methods
    - **Particle-structured meshes**: Develop a collection of scalable tools that account for different distributions of particles and grid patches; use an intermediate mesh as a coverage of particles
    - **Particle-unstructured meshes**: Efficient point search and parallel representation



FASTMath Linear Solver Toolset

- **Scalable algebraic solvers** from millions to billions of degrees of freedom for one to million-way parallelism
- **Composable solvers built from**
  - Krylov methods
  - Geometric and algebraic multigrid
  - Domain decomposition
  - Schur complement preconditioners
  - LU and ILU factorizations
- **Associated software packages**: hypre, PETSc, ML and MueLu (Trilinos), SuperLU and PDSLin
- **For more information**: Rob Falgout, Barry Smith, Jonathan Hu, Sherri Li, Jim Demmel



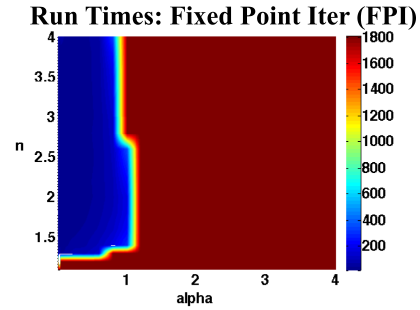


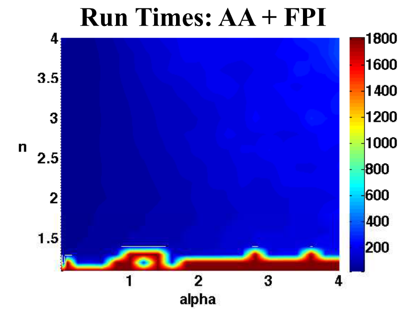
Significant effort will be devoted toward continued linear solvers development in FASTMath

- **Efficient many-core matrix/vector kernel classes**
  - Collaborative development of a suite of low-level, thread-aware kernels
  - Vector, matrix, smoothing, prolongation, restriction operations
- **Development of multigrid/multilevel methods**
  - Latency tolerant methods for algebraic multigrid; reduce communication through redundant computations at coarse levels and aggregated messaging
  - New algorithms to use idle processors at coarse levels to accelerate convergence
  - Energy minimizing multilevel algorithms for symmetric and non-symmetric systems
- **New generation of linear solvers that are memory and communication efficient**
  - Enhanced SuperLU_Dist and PDSLin for many-core architectures using threads for update of supernodal blocks
  - Scalable, sparse LDL^T factorization codes that use parallel, block fan-out algorithms that is multi-core aware
  - New sparse direct methods based on approximations of low rank structures using Hierarchically Semi-separable matrices
  - Communication avoiding algorithms for sparse matrix solvers using tournament pivoting, develop theoretical bounds for HSS method communication, exploration of HSS preconditioners that avoid communication

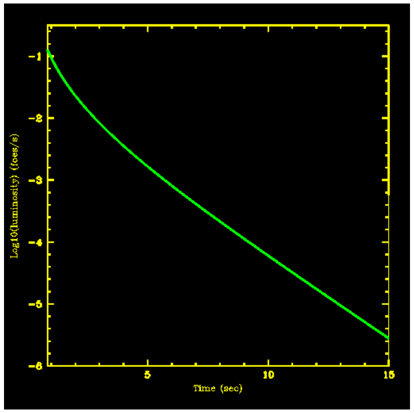
Newton-Krylov and Fixed Point Iterative Solvers for nonlinear solution of large-scale applications

- **Goal of the technology**: Efficient and robust solution of nonlinear algebraic systems arising from implicit approaches
- **Nonlinear solver capabilities**
  - Newton-Krylov and accelerated Fixed Point iterations
  - Differential Variational Inequality (DVI) solvers
  - Numerous options for tuning solvers for specific applications
  - Interfaces to FASTMath Linear Solvers
- **Associated software package**: KINSOL, PETSc, NOX
- **For more information**: Carol Woodward, Barry Smith, Andy Salinger





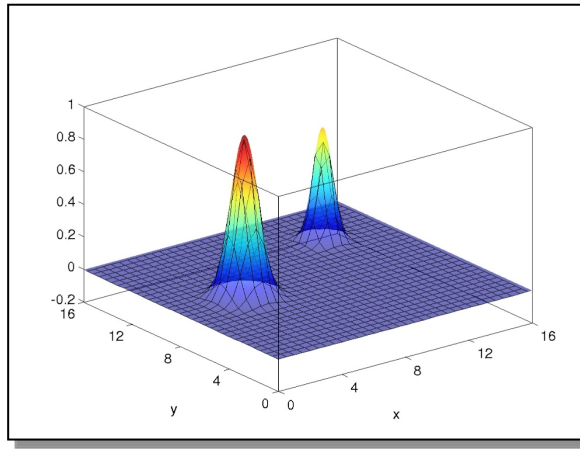
Accelerated fixed point methods (AA +FPI) improved robustness and speed of an unmodified fixed point method (FPI) for subsurface flow



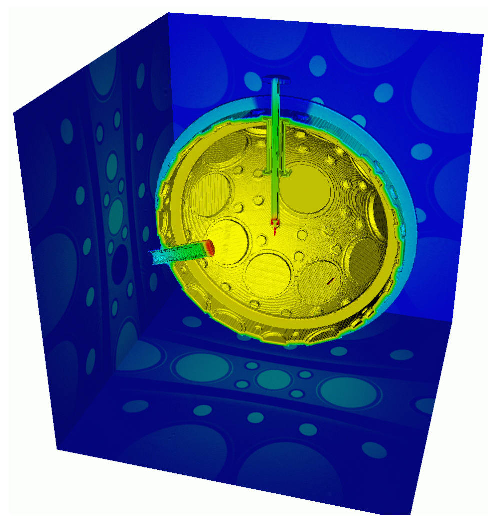
Newton-Krylov enabled first radiation-hydrodynamics model of entire proto-neutron star cooling period

High order implicit, explicit, and IMEX time integrators

- **Goal of the technology**:
  Efficient and robust time integration of stiff implicit, explicit, and multirate systems
- **Time Integration capabilities**
  - Variable step and order BDF ODE and DAE integrators
  - Forward and adjoint sensitivities
  - Data structure neutral implementations
  - IMEX solvers for multirate systems
- **Associated software package**:
  CVODE(S), IDA(S), PETSc
- **For more information**:
  Carol Woodward, Barry Smith, Dan Reynolds



Reconstructed sources of test atmospheric contaminant release; CVODE(S) for forward integration and gradients via adjoints



Intensity on NOVA test chamber ; solution of Boltzmann transport equation using IDA

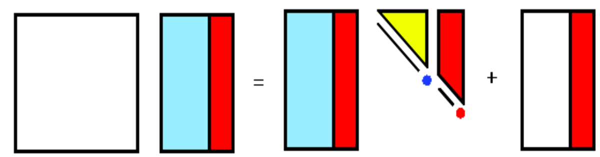
We will continue to develop nonlinear solvers and time integration technologies

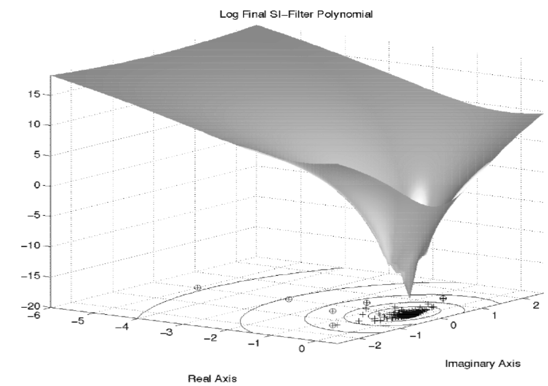
- **Nonlinear solvers**
  - Deploy Anderson accelerated fixed point methods that have been shown to be competitive with Newton-Krylov methods
  - Support for implicit and semi-implicit solution approaches including contributions to kernel library development, extension of Fortran90 interfaces, parallelism strategies to reduce communication costs
- **DVIs**
  - Provide a flexible, abstract framework that allows scalable linear algebra to be consistently incorporated with the switching states
  - Box-constrained DVI problems using scalable active set methods that leverage multigrid component of FASTMath
  - Polyhedrally-constrained DVI problems and cone-constrained problems using active set methods based on gradient-projection approaches
- **Time integrators for multi-physics and multi-scale problems**
  - Combined Implicit/Explicit (IMEX) methods that allow greater accuracy and stability over traditional splittings
  - Develop general integration infrastructure in SUNDIALS for IMEX, symplectic, and multiscale methods

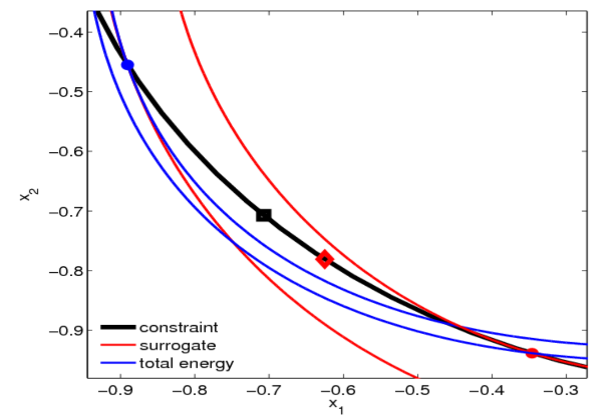
FASTMath Eigenvalue Toolset

- **Scalable eigenvalue algorithms and solvers** from millions to billions of degrees of freedom
- **Techniques include**
  - Implicitly Restarted Lanczos and Arnoldi iterations for linear eigenvalue problems
  - Nonlinear Arnoldi, Jacobi-Davidson, rational Krylov, second order Arnoldi methods for one-parameter nonlinear eigenvalue problems
  - Optimization approaches for solving eigenvalue problems with nonlnearity in the eigenvectors
- **Associated software packages**:
  - PARPACK, KSSOLV

IRAM





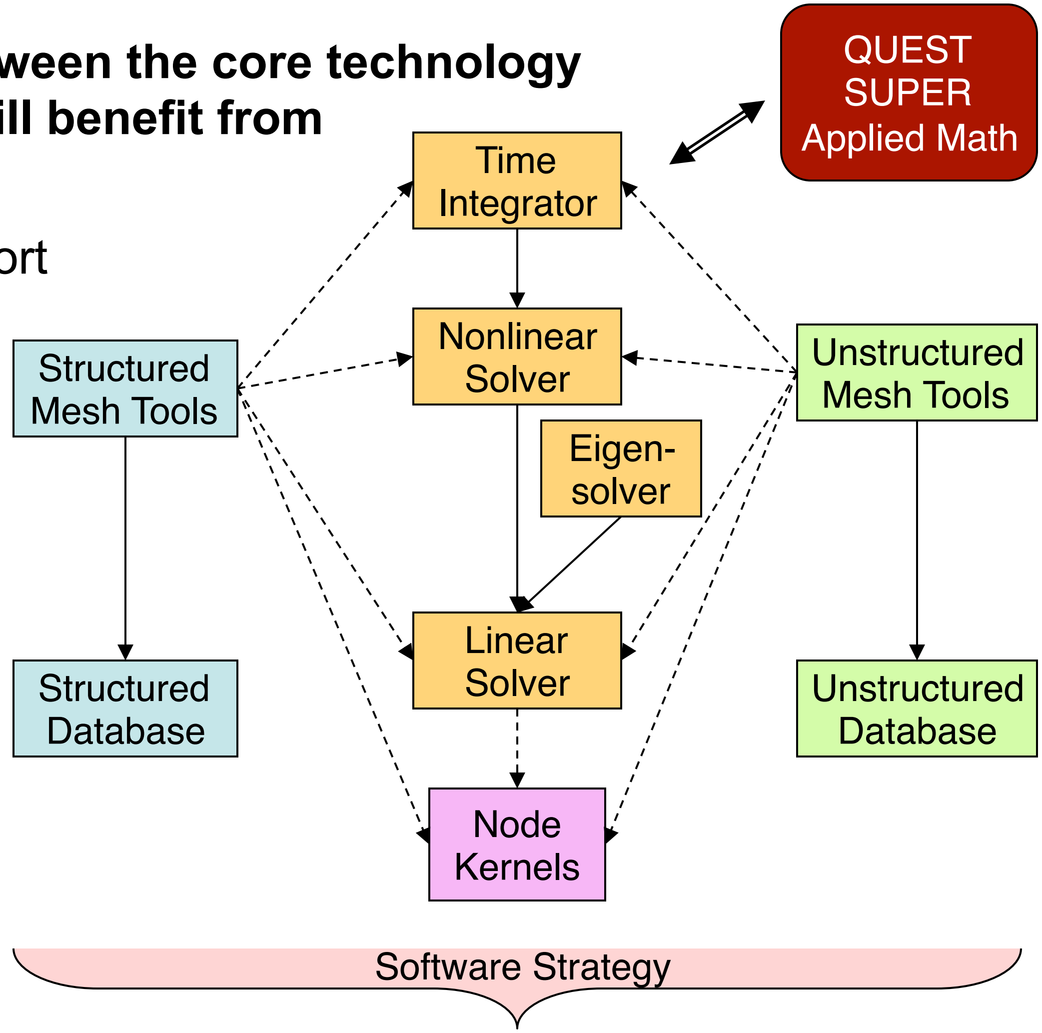


FASTMath includes a focused effort for continued eigensolver development

- Hybrid OpenMP/MPI programming models for linear eigensolvers that compute a relatively large number of eigenpairs
- Solution of eigenvalue problems with nonlinear eigenvectors
  - Leveraging previous convergence analysis of existing iterative schemes
  - Extend work in selected inversion-based fast techniques
  - Special constrained minimization algorithms to develop fast iterative schemes

FASTMath provides a unique opportunity to develop integrated capabilities

- **As we provide integration between the core technology areas, science applications will benefit from**
  - Expanded capabilities
  - Decreased development effort
- **Development efforts for expanded capability integration**:
  - Adaptivity through the software stack
  - Common linear system fill interface
  - Architecture-aware compute node kernels
  - Unified software strategy



Large efficiency gains can be achieved by incorporating adaptivity through the software stack

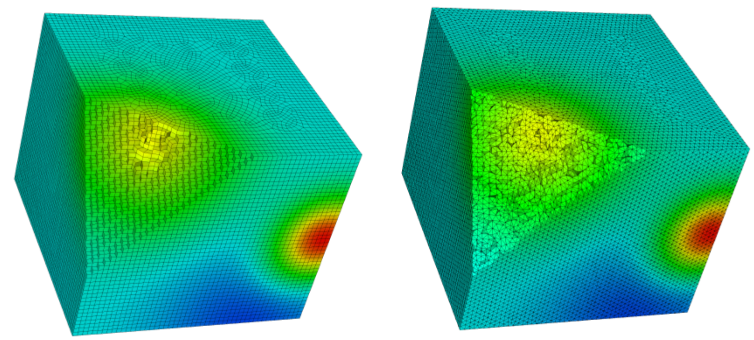
- **Structured grid interactions with linear solvers**
  - Customize data structures in solver libraries to work efficiently with structured grids
    - Develop matrix-free operator class and specialized vector class in PETSc to eliminate current need to 'flatten' the data rectangular patches
    - Retain data storage used by BoxLib and Combo
    - Matrix-free representation of linear operators through direct calls to operators
  - Develop efficient block-structured AMR multigrid methods that take advantage of constant coefficients and varying anisotropy
- **Unstructured grid solvers**
  - Ensure all information transfer with application data is performed without the use of intermediary files
  - Tightly integrate partition model used by mesh adaptation with global system
  - Extend solvers' matrix and vector data structures to improve their dynamism
    - New sparse matrix and vector classes that efficient insertion and removal of entries after mesh modification
    - Fast methods for dynamically load balancing new systems

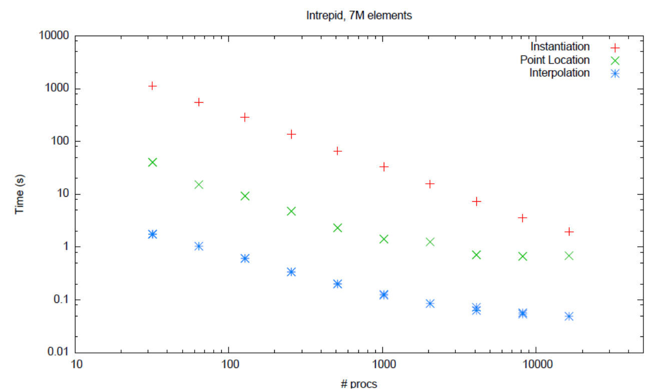
Transfer and manipulation of application field data is required throughout the solution process

- Interactions between CAD and mesh models including both structured and unstructured grids
- Equation solvers
  - Support for numbering matrix equations segregated by physics, interleaved, grouped into small dense blocks based on physics
  - Opens the door for Schur-complement preconditioners or optimal ordering for cache usage or preconditioner performance
  - Facilitate development of a matrix-loading software layer that allows stronger integration of equation discretization and solver libraries
- Multi-scale / Multi-physics simulation support
  - Changing field representations (projections, norms)
  - Satisfy constraints (conservation of a scalar)
  - Change form (particles to continuum)
  - Evolution of fields within adaptive processes

Some parallel mesh-to-mesh solution transfer tools exist in FASTMath

- **Goal/Objectives of the technology**: map solutions between disparate meshes on distributed memory parallel machines
- **Approach/Technical Details**
  - Source/target meshes on disparate sets of processors or the same ones
  - Decomposes problem into initialization, point location, interpolation, conservation/normalization
  - Extensible to multiple element types (FD, FE, SE), orders
  - Scalable to thousands of processors
- **Associated software package**: MOAB, Trilinos/Intrepid
- **For more information**: Tim Tautges, tautges@mcs.anl.gov





Future work focuses on both loosely and strongly coupling strategies

- **Loosely coupled solution transfer**
  - Initial, parallel solution transfer in MOAB
  - Support for conservation and accuracy constraints on the transferred field
    - Minimizing L2 norm of transferred solution
    - Guaranteeing divergence-free properties
    - Support for element-wise integral constraints
  - Fast local transfer for mesh modification
- **Strongly coupled solution transfer**
  - Development of abstractions for interfacing to solvers that simplify construction of coupling equations
  - Express point location information in terms of matrices and operator stencils (requiring mesh solver interoperability)

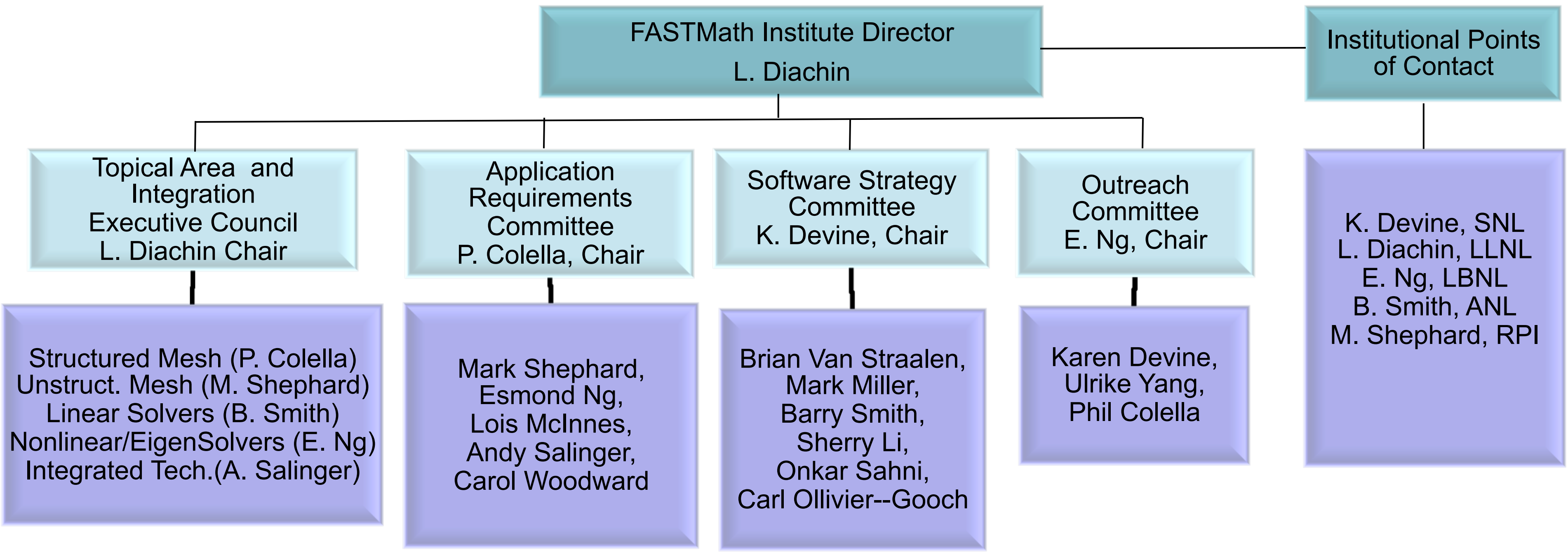
All FASTMath technologies will focus on performance engineering for multi-/many-core architectures

- **MPI Paralelism**
  - Ensure all FASTMath tools operate efficiently at 10^5 to 10^6 cores
  - Architecture-aware and multi-objective partitioners for load balancing
  - Communication avoiding/hiding and latency tolerant algorithms
  - Synchronization reducing algorithms by focusing on neighborhoods, use of one-sided messages, remote memory access
- **Node-level parallelism**
  - Use of threading techniques
  - Multi-core kernels and data ordering
  - Exploit compilers, code transformation tools, programming models and run-time systems as they become available
- **Data locality**
  - Hierarchical partitioning methods and local data ordering methods
  - Shared efficient data layouts in software packages to prevent re-organization in integrated services
  - Code transformation systems, domain specific language extensions to gain performance while maintaining reusability
- **Coordinated parallelism between different levels (MPI, node, instruction)**

Unifying our software processes will improve usability and interoperability of FASTMath products

- **Software quality**: Define a minimum set of practices that each FASTMath-compliant software product has (e.g., versioning, documentation)
- **East of adoption**:
  - Maintain a web-based portal documenting and providing links to software
  - FASTMath-help mailing list
  - Software testing and portability
  - Provide example installations on DOE LCFs
  - Shared utilities (I/O, error handling) where feasible
- **Integration of different products**
  - Identification of early adopters to help with usability and deployment
  - Application motivated mini-applications for testing
  - High-level build systems for integrated software
  - Regulation of backward compatibility issues
  - Regular release cycle

FASTMath's management is designed to be responsive to application needs and SciDAC program priorities



For more information, please contact any of the following

- FASTMath Institute Director:
  - Lori Diachin,
    diachin2@llnl.gov, 925-422-7130

- FASTMath Executive Council
  - Phil Colella, Structured Mesh Tools
    pcolella@lbl.gov, 510-486-5412
  - Esmond Ng, Nonlinear/Eigensolvers
    egng@lbl.gov, 510-495-2851
  - Andy Salinger, Integrated Technologies
    agsalin@sandia.gov, 505-845-3523
  - Mark Shephard, Unstructured Mesh Tools
    shephard@scorec.rpi.edu, 518-276-8044
  - Barry Smith, Linear Solvers,
    bsmith@mcs.anl.gov, 630-252-9174