LLNL-PRES-501654

FASTMath: An overview of mathematical algorithms and software

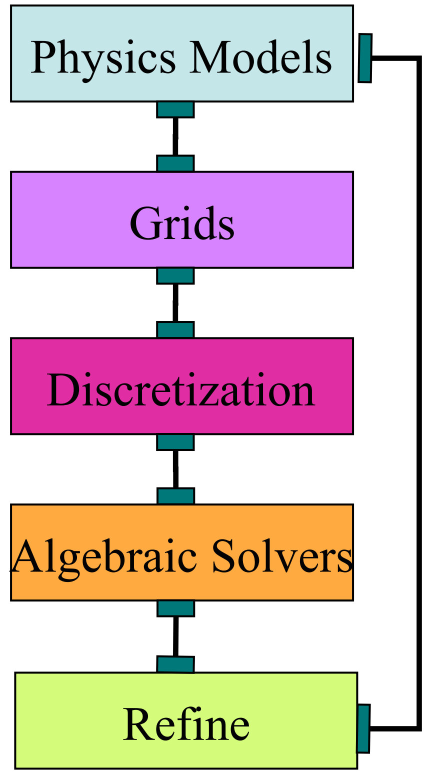
FASTMath Team

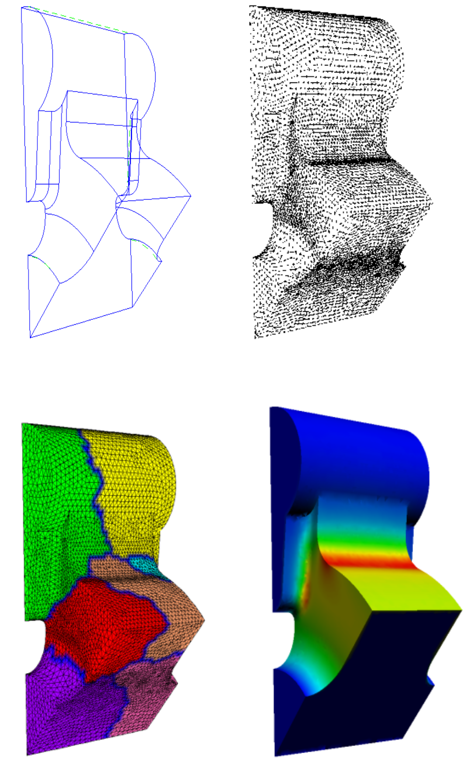
Lori Diachin, Institute Director

FASTMath SciDAC Institute

Mathematical algorithms and software are foundational to HPC simulations

- Develop a mathematical model of the phenomenon of interest
- Approximate the model using a discrete representation
- Solve the discrete representation
- Adapt and refine the mesh or model
- Couple different physics, scales, regions together





These steps require: CAD models, grid generation, high order discretizations, time integration techniques, linear and nonlinear solution of algebraic systems, eigensolvers, mesh refinement strategies, physics coupling methods, particle techniques, etc…

First consider a very simple example

- 1D rod with one end in a hot water bath, the other in a cold water bath
- Mathematical Model

$$\nabla^2 T = 0 \in \Omega$$

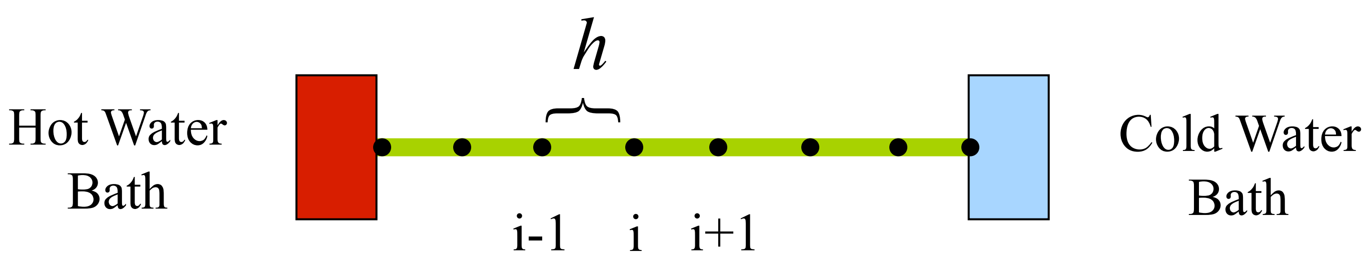
$$T(0) = 180^{o} \quad T(1) = 0^{o}$$

Discretizating the equations

- Approximate the derivatives in the continuous equations with a discrete representation that is easier to solve
- One approach: Finite Differences

$$\nabla^2 T \approx (T_{i+1} - 2T_i + T_{i-1})/h^2 = 0$$

$$T_0 = 180^o \qquad T_n = 0^o$$



Solve for the unknowns T_i

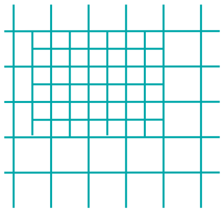
- Set up a matrix of the unknown coefficients
  - include the known boundary conditions
- Solve the linear system for T_i

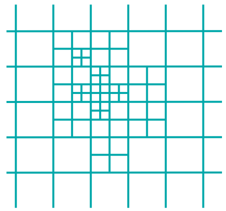
$$\begin{pmatrix} 2 & -1 & 0 & \ldots\ldots\ldots & 0 \\ -1 & 2 & -1 & 0\ldots\ldots & 0 \\ 0 & -1 & 2 & -1 & 0\ldots 0 \\ & & \ldots\ldots\ldots & & \\ 0\ldots\ldots\ldots 0 & & & -1 & 2 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ . \\ T_{n-1} \end{pmatrix} = \begin{pmatrix} 180\, h^2 \\ 0 \\ 0 \\ . \\ 0 \end{pmatrix}$$

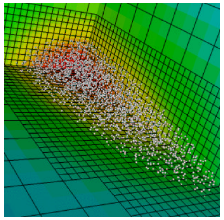
- Visualize and analyze the results

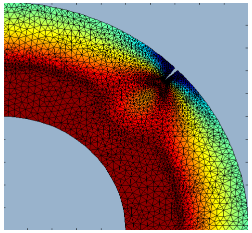
As problems get more complicated so do the steps in the process

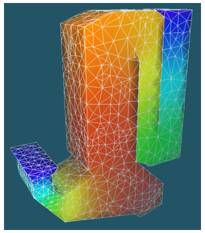
- Different discretization strategies exist for differing needs
  - Efficiency
  - Flexibility







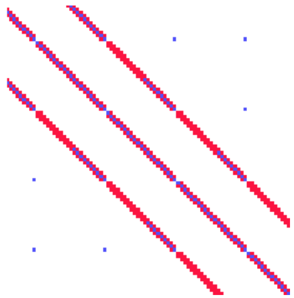


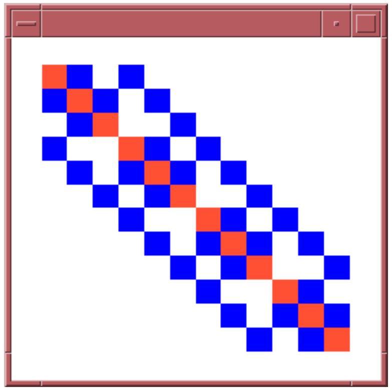


- Most problems are time dependent and nonlinear
  - Need higher algorithmic levels than linear solvers
- Increasingly combining multiple physical processes
  - Interactions require careful handling
- Goal-oriented problem solving required optimization, uncertainty quantification

As problems grow in size so do the corresponding discrete systems

- Targeting applications with billions grid points and unknowns
- Most linear systems resulting from these techniques are LARGE and sparse
- Often most expensive solution step
- Solvers:
  - Direct Methods (e.g. Gaussian Elimination)
  - Iterative Methods (e.g. Krylov Methods)
    - Preconditioning is typically critical
    - Mesh quality affects convergence rate
- Many software tools developed at DOE labs deliver this functionality as numerical libraries
  - PETSc, Hypre, SuperLU, etc.

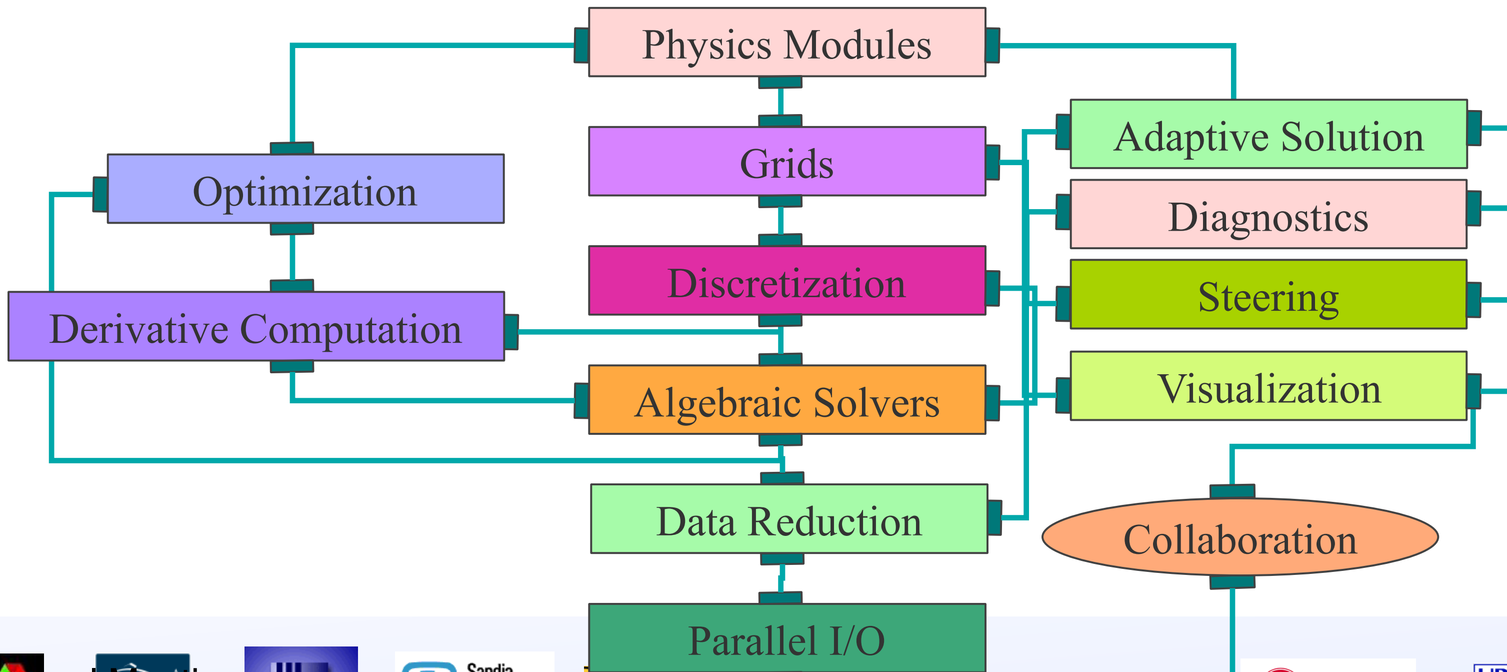




Modern scientific application development involves many different tools, libraries, and technologies

Observation: Exascale computing will enable high-fidelity calculations based on multiple coupled physical processes and multiple physical scales

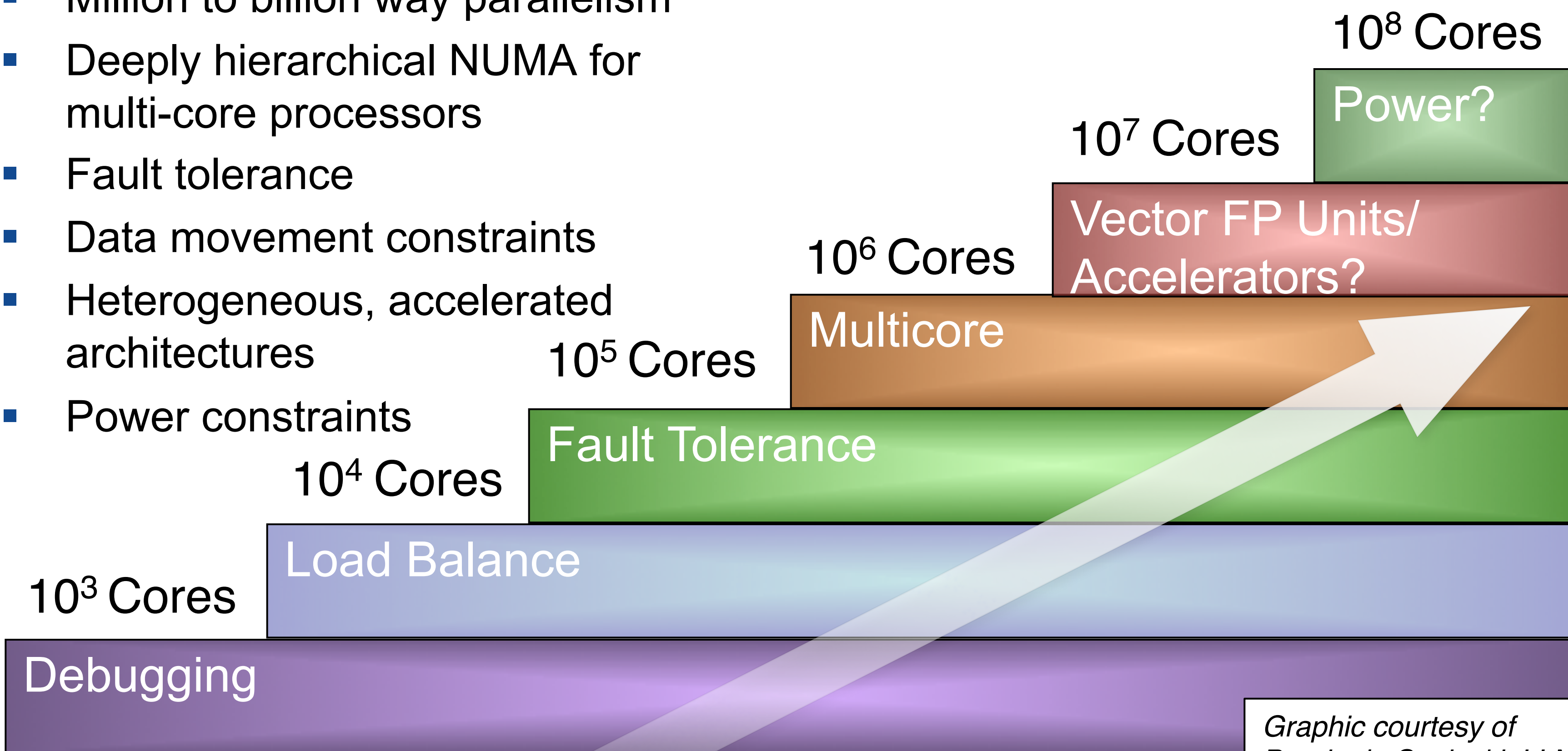
- Adaptive algorithms
- Composite or hybrid solution strategies
- High-order discretization strategies
- Sophisticated numerical tools



Modeling and simulation is significantly complicated by the change in computing architectures

Scientific computing software must address ever increasing challenges:

- Million to billion way parallelism
- Deeply hierarchical NUMA for multi-core processors
- Fault tolerance
- Data movement constraints
- Heterogeneous, accelerated architectures
- Power constraints



Graphic courtesy of Bronis de Supinski, LLNL

These complexities result in common challenges facing application scientists

- Reliability
  - Accurate, stable discretizations
  - Robust solution algorithms
  - Error minimization
- Software Complexity
  - Interoperating numerical software
  - New algorithms (e.g., interactive/ dynamic techniques, algorithm composition)
  - New programming models
- Performance
  - Load balancing (perhaps dynamic)
  - Portability across architectures
  - Massive scale

Application life cycle costs are increasing

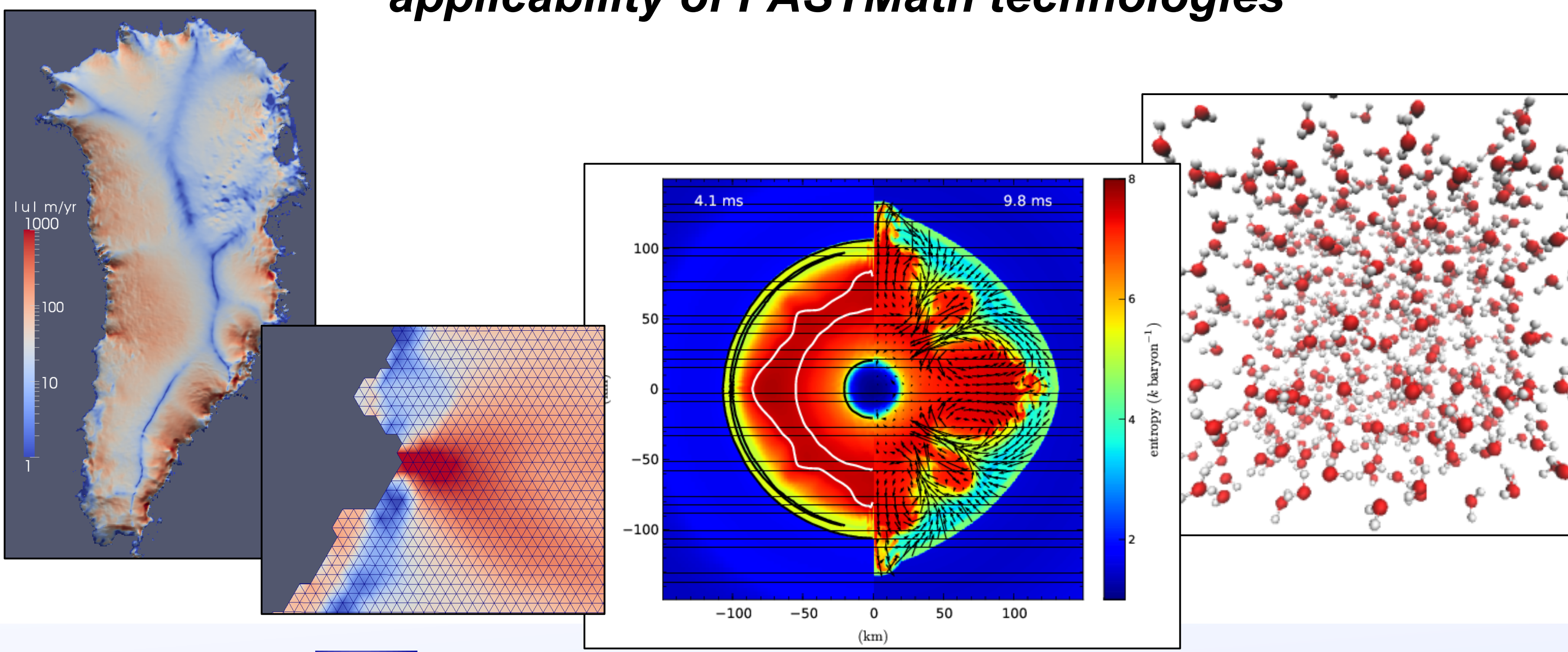
- Require the combined use of software developed by different groups
- Difficult to leverage expert knowledge and advances in subfields
- Difficult to obtain portable performance

Too much energy focused on too many details

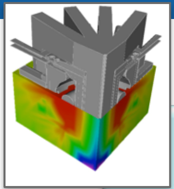
- Little time to think about modeling, physics, mathematics
- Fear of bad performance without custom code
- Even when code reuse is possible, it is far too difficult

The FASTMath SciDAC project focuses on the development and use of mathematics software libraries

The FASTMath SciDAC Institute develops and deploys scalable mathematical algorithms and software tools for reliable simulation of complex physical phenomena and collaborates with DOE domain scientists to ensure the usefulness and applicability of FASTMath technologies

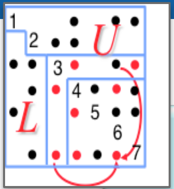


FASTMath encompasses three broad topical areas



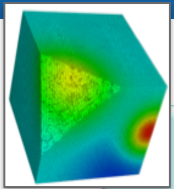
Tools for Problem Discretization

- Structured grid technologies
- Unstructured grid technologies
- Adaptive mesh refinement
- Complex geometry
- High-order discretizations
- Particle methods
- Time integration



Solution of Algebraic Systems

- Iterative solution of linear systems
- Direct solution of linear systems
- Nonlinear systems
- Eigensystems
- Differential variational inequalities

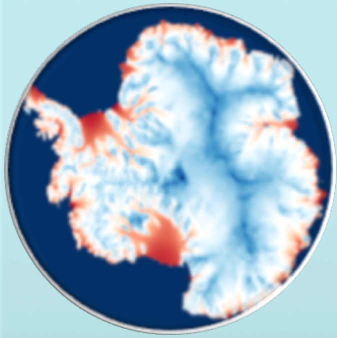


High Level Integrated Capabilities

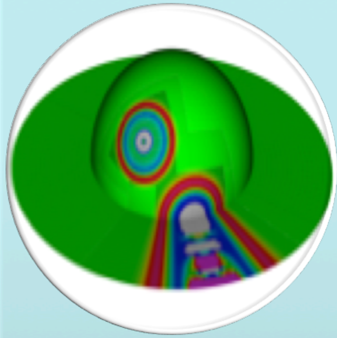
- Adaptivity through the software stack
- Management of field data
- Coupling difference physics domains
- Mesh/particle coupling methods

Argonne National Laboratory · Berkeley Lab · Sandia National Laboratories · Berkeley University of California · Rensselaer · SMU · The University of British Columbia

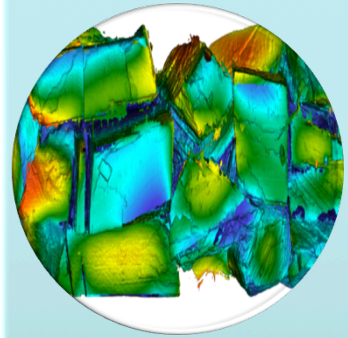
Structured grid capabilities focus on high order, mapped grids, embedded boundaries, AMR and particles



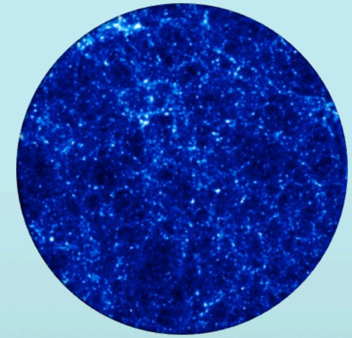
Structured AMR



Mapped-multiblock grids



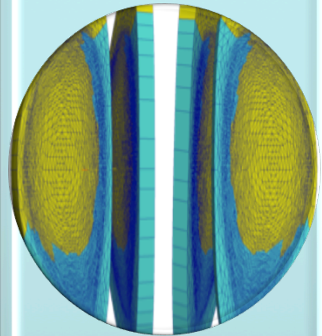
Embedded boundary methods



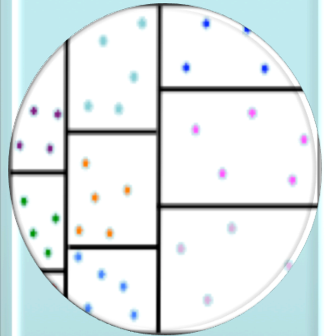
Particle-based methods

Application to cosmology, astrophysics, accelerator modeling, fusion, climate, subsurface reacting flows, low mach number combustion, etc.

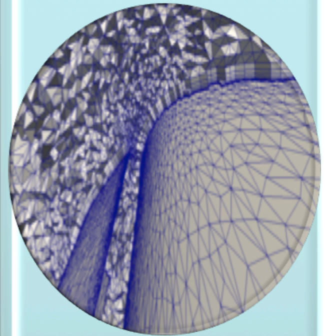
Our unstructured grid capabilities focus on adaptivity, high order, and the tools needed for extreme scaling



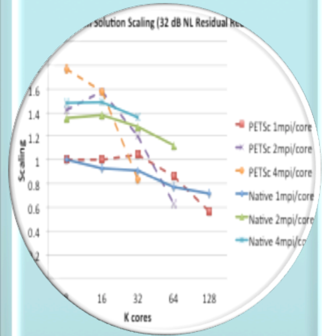
Parallel mesh infrastruct-ures



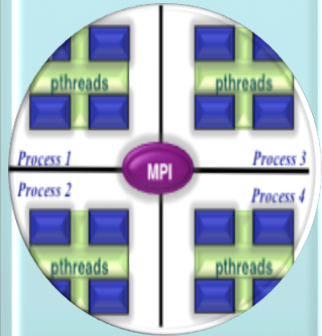
Dynamic load balancing



Mesh adaptation and quality control



Parallel performance on unstructured meshes

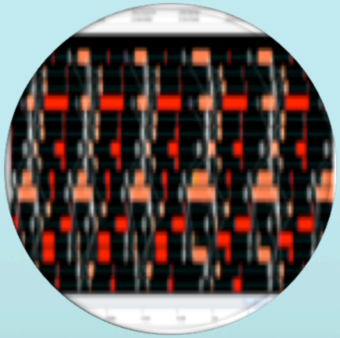


Architecture aware implementa-tions

Application to fusion, climate, accelerator modeling, NNSA applications, nuclear energy, manufacturing processes, etc.

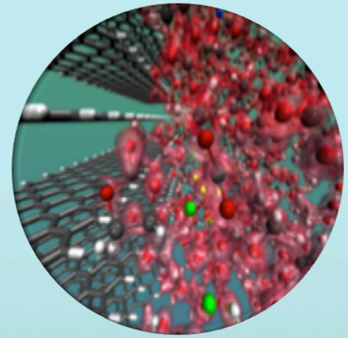
Argonne NATIONAL LABORATORY · BERKELEY LAB · Sandia National Laboratories · Berkeley University of California · Rensselaer · SMU · UBC THE UNIVERSITY OF BRITISH COLUMBIA

Our work on algebraic systems provides key solution technologies to applications

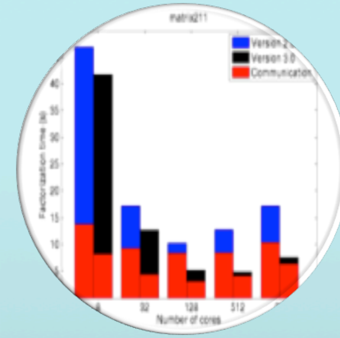


Linear system solution using direct and iterative solvers

Nonlinear system solution using acceleration techniques and globalized Newton methods



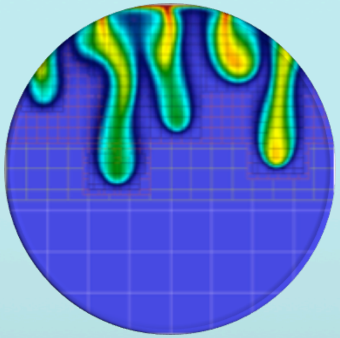
Eigensolvers using iterative techniques and optimization

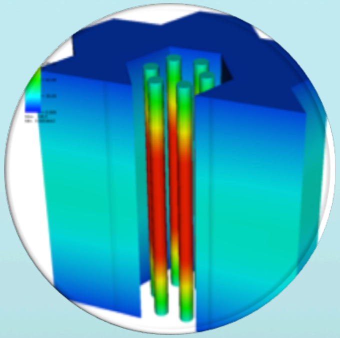


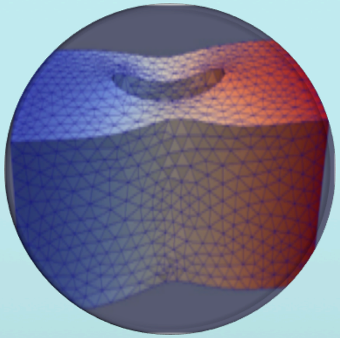
Architecture aware implementations

Application to fusion, nuclear structure calculation, quantum chemistry, accelerator modeling, climate, dislocation dynamics etc,

Integrating technologies is a key value added by the FASTMath Institute







Mesh/solver interactions

Mesh-to-mesh coupling methods

Unstructured mesh technologies into simulation workflows

Software unification strategies

Application to climate, plasma surface interactions, structural mechanics, nuclear energy, cosmology, fluid flow, etc.

Argonne National Laboratory · Berkeley Lab · Sandia National Laboratories · Berkeley University of California · Rensselaer · SMU · The University of British Columbia

FASTMath encompasses our algorithm development in widely used software

FASTMath Software

Structured Mesh Tools
- BoxLib (Ann Almgren)
- Chombo (Phil Colella)

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

Unstructured Mesh Tools
- PUIMI (Seegyoung Seol)
- MeshAdapt (Mark Shephard)
- MOAB (Vijay Mahadevan)
- Mesquite (Lori Diachin)
- PHASTA (Ken Jansen)
- APF (Cameron Smith)

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

Partitioning Tools
- Zoltan (Karen Devine)
- ParMA (Mark Shephard)

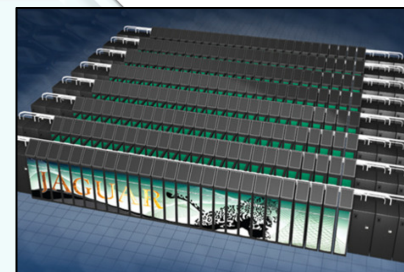
Eigensolvers
- PARPACK (Chao Yang)

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

Our research to improve performance on HPC platforms focuses on both inter- and intra-node issues

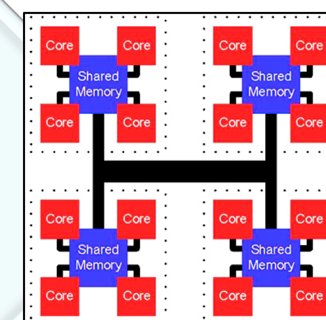
Inter-node: Massive Concurrency

- Reduce communication
- Increase concurrency
- Reduce synchronization
- Address memory footprint
- Enable large communication/computation overlap



Intra-node: Deep NUMA

- MPI + threads for many packages
- Compare task and data parallelism
- Thread communicator to allow passing of thread information among libraries
- Low-level kernels for vector operations that support hybrid programming models



We are developing new algorithms that address key bottlenecks on modern day computers

Reduce communication

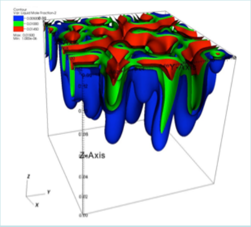
- AMG: develop non-Galerkin approaches, use redundancy or agglomeration on coarse grids, develop additive AMG variants (hypre) (2X improvement)
- Hierarchical partitioning optimizes communication at each level (Zoltan) (27% improvement in matrix-vector multiply)
- Relaxation and bottom solve in AMR multigrid (Chombo) (2.5X improvement in solver, 40% overall)
- HSS methods

Increase concurrency

- New spectrum slicing eigensolver in PARPACK (Computes 10s of thousands of eigenvalues in small amounts of time)
- New pole expansion and selected inversion schemes (PEXSI) (now scales to over 100K cores)
- Utilize BG/Q architecture for extreme scaling demonstrations (PHASTA) (3.1M processes on 768K cores unstructured mesh calculation)

Reduce synchronization points

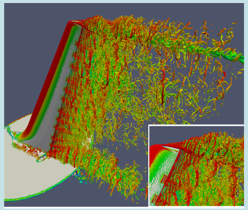
- Implemented pipelined versions of CG and conjugate residual methods; 4X improvement in speed (PETSc) (30% speed up on 32K cores)



Used in PFLOTRAN applications

Address memory footprint issues

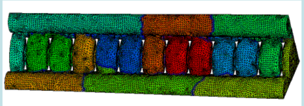
- Predictive load balancing schemes for AMR (Zoltan) (Allows AMR runs to complete by maintaining memory footprint)
- Hybrid programming models



Used in PHASTA extreme scale applications

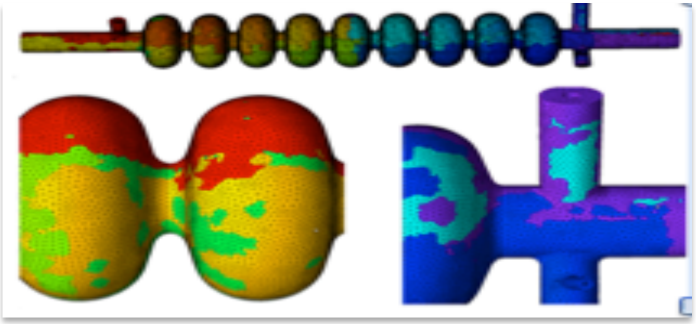
Increase communication and computation overlap

- Improved and stabilized look-ahead algorithms (SuperLU) (3X run time improvement)

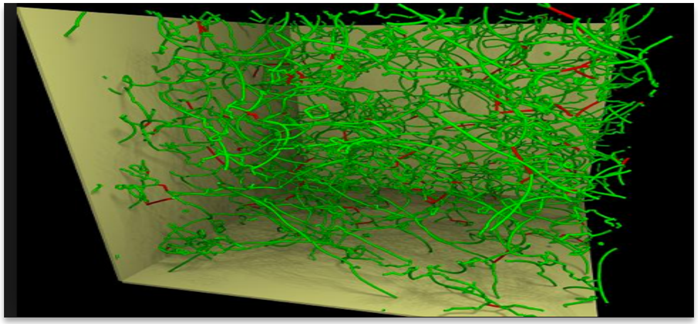


Used in Omega3P accelerator simulations

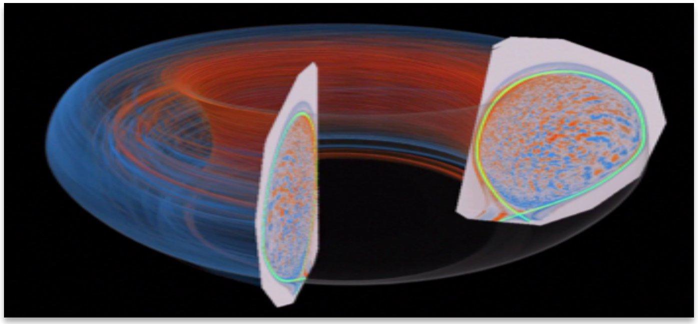
We have helped the application teams significantly reduce time to solution in their simulations



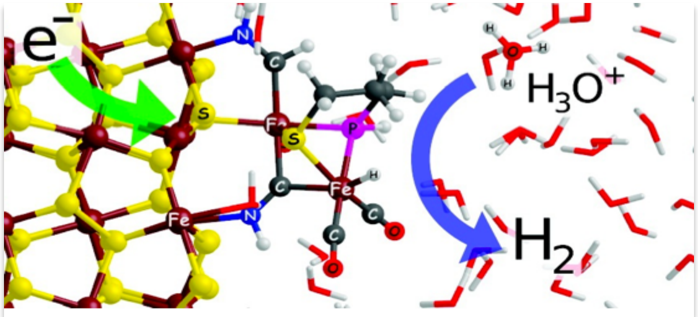
Sparse direct solves improve time to solution 20X for accelerators allowing 8 cavity simulation (Spentzouris)



Acceleration-based nonlinear solvers speed up dislocation dynamics 35-50%; multistage Runge-Kutta methods reduce time steps by 94% (Arsenlis)

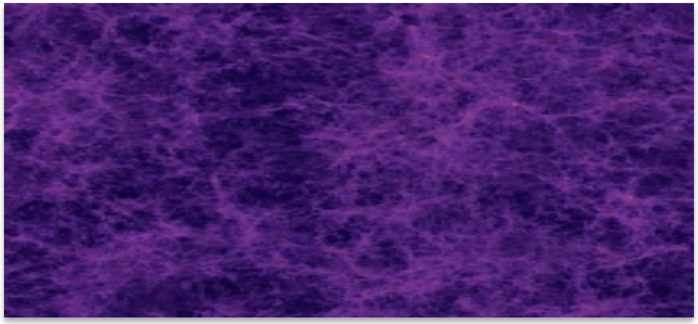


Sped up flux surface creation to improve 2D mesh generation in fusion application from 11.5 hours to 1 minute (Chang)

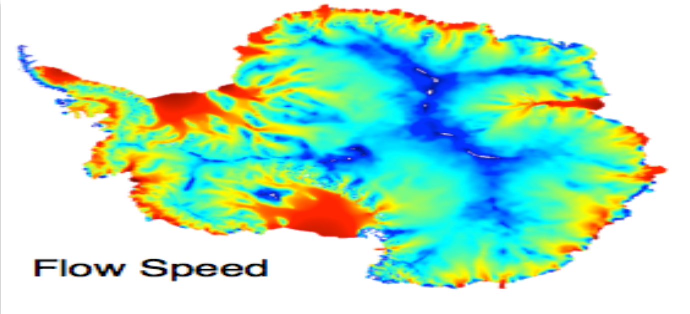


Sophisticated eigensolvers significantly improve materials calculations in many domains including ions in solution (Car), excited state phenomenon (Chelikowsky, Head-Gordon)

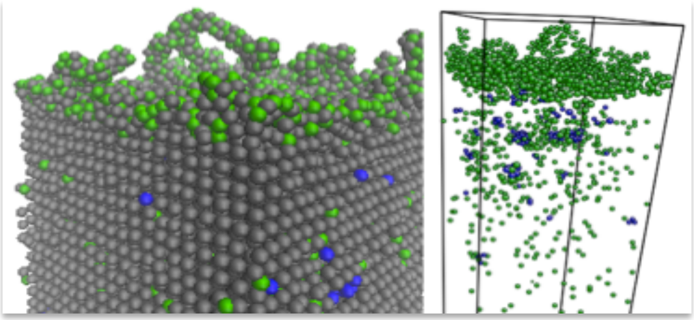
We have helped the application teams achieve unprecedented resolution and increased reliability



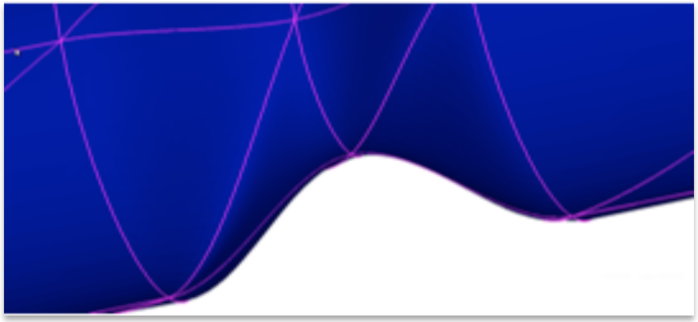
Astrophysics Lyman-α forest simulation at 4096^3 in an 80Mpc/h box; produced statistics at 1% accuracy for first time (Habib)



Predictions of grounding line match experiment for first time in ice sheet modeling due to AMR (Price)



Implicit ODE integrators combined with AMG linear solvers enables solution of 4D reaction-diffusion eqns for plasma surface interactions (Wirth)



High-order unstructured meshes for particle accelerators overcome mesh generation/ adaptation bottlenecks (Spentzouris)

We are addressing key package management issues targeting interoperability software

Issues Addressed:

- Inconsistent installation processes
- Inconsistent or missing configuration information
- Copying sources as a means of managing dependencies
- Spoofing sources (e.g. MPI) as a means of simplifying package code
- Inconsistent or missing versioning
- Managed installations

- Publication: "Package Management Practices Essential for Interoperability: Lessons learned and Strategies Developed for FASTMath", First workshop on Sustainable Software for Science: Practice and Experiences, 2013

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

Lawrence Berkeley National Laboratory

Mark Adams
Ann Almgren
Phil Colella
Anshu Dubey
Dan Graves
Sherry Li
Lin Lin
Terry Ligocki
Mike Lijewski
Peter McCorquodale
Esmond Ng
Brian Van Straalen
Chao Yang
Subcontract: Jim Demmel (UC Berkeley)

Lawrence Livermore National Laboratory

Barna Bihari
Lori Diachin
Milo Dorr
Rob Falgout
Mark Miller
Jacob Schroder
Carol Woodward
Ulrike Yang
Subcontract: Carl Ollivier-Gooch (Univ of British Columbia)
Subcontract: Dan Reynolds (Southern Methodist)

Rensselear Polytechnic Inst.

E. Seegyoung Seol
Onkar Sahni
Mark Shephard
Cameron Smith
Subcontract: Ken Jansen (UC Boulder)

Argonne National Laboratory

Jed Brown
Lois Curfman McInnes
Todd Munson
Vijay Mahadevan
Barry Smith
Subcontract: Jim Jiao (SUNY Stony Brook)
Subcontract: Paul Wilson (Univ of Wisconsin)

Sandia National Laboratories

Karen Devine
Glen Hansen
Jonathan Hu
Vitus Leung
Siva Rajamanickam
Michel Wolf
Andrew Salinger

FASTMath Tutorial Goals

- Provide a basic understanding of a variety of applied mathematics algorithms for scalable linear, nonlinear, and ODE solvers as well as discretization technologies (e.g., adaptive mesh refinement for structured and unstructured grids)
- Provide an overview of FASTMath software tools available to perform these tasks on HPC architectures
- Practice using one or more of these software tools on basic demonstration problems

FASTMath Tutorial Schedule: Friday, August 8

11:00	An Overview of Mathematical Algorithms and Software	Lori Diachin, LLNL
11:40	Algebraic Solvers in FASTMath: An Introduction	Barry Smith, ANL
12:00 PM	Lunch and Hands-on Exercises	
1:00	PETSc: Portable, Extensible Toolkit for Scientific Computing	Barry Smith, ANL
2:00	HYPRE: High Performance Preconditioners	Rob Falgout, LLNL
2:30	Break	
3:00	SuperLU: Parallel Direct Solvers	Xiaoye (Sherry) Li, LBNL
3:30	Sundials: Suite of Nonlinear and Differential/Algebraic Equation Solvers	Carol Woodward, LLNL
4:00	Intro to Unstructured Mesh Technologies (Part 1)	Vijay Mahadevan, ANL, Mark Shephard and Cameron Smith, RPI, and Glen Hansen, SNL
4:30	Panel: Challenges in Extreme Scale Solvers	FASTMath Team
5:30	Dinner Talk: "Perspectives on Teaming from the DOE National Labs"	Lori Diachin, LLNL
6:30	FASTMath Hands-on Exercises	Mark Miller, LLNL and the FastMath Team
9:30	Wrap-up	

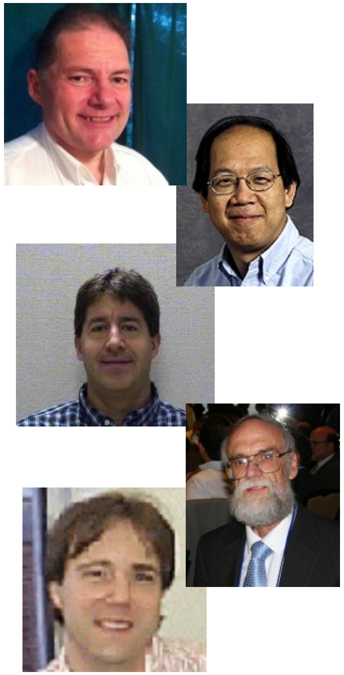
FASTMath Tutorial Schedule: Saturday, August 9

7:30 AM	Continental Breakfast	
8:30	Unstructured MeshTechnologies (Part 2)	Vijay Mahadevan, ANL, Mark Shephard and Cameron Smith, RPI, and Glen Hansen, SNL
9:30	Block Structured AMR Libraries and Their Interoperability with Other Math Libraries	Mark Adams and Anshu Dubey (LBNL)
10:30	Break	
11:00	FASTMath Hands-on Exercises for meshing, AMR	Mark Miller, LLNL and FASTMath Team
12:00 PM	Lunch and Hands-on Exercises	
1:00	Wrap-up	

For more information, please contact any of the following or visit our web site

- 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



http://www.fastmath-scidac.org

Auspices and Disclaimer

Support for this work was provided through Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.