Overview of Numerical Algorithms and Software for Extreme-Scale Science

Presented to
ATPESC 2019 Participants

Lois Curfman McInnes
L3 Lead for Mathematical Libraries, DOE Exascale Computing Project
Senior Computational Scientist, MCS Division, Argonne National Laboratory

Q Center, St. Charles, IL (USA)
Date 08/06/2019

U.S. DEPARTMENT OF ENERGY | Office of Science

ATPESC Numerical Software Track

EXASCALE COMPUTING PROJECT

Argonne NATIONAL LABORATORY · BERKELEY LAB · Sandia National Laboratories · Rensselaer · SMU

Track 5: Numerical Algorithms & Software for Extreme-Scale Science

Monday, August 5

Time	Title of presentation	Lecturer
8:30 am	Adaptive Linear Solvers and Eigensolvers	Jack Dongarra, U. Tennessee-Knoxville

Tuesday, August 6

Time	Title of presentation	Lecturer
8:30 am	Communication-Avoiding Algorithms for Linear Algebra	Jim Demmel, UC Berkeley
9:30 am – 9:30 pm	Numerical Algorithms and Software for Extreme-Scale Science: Integrated Lectures and Hands-on Lessons	Track 5 Team

See details next page

Thursday, August 8

Time	Title of presentation	Lecturer
8:30 am	The Convergence of Big Data and Large-scale Simulation	David Keyes, KAUST

Track 5: Numerical Algorithms & Software for Extreme-Scale Science

Tues, Aug 6, 2019

Time	Title of presentation	Lecturer
9:30 am	**Overview of Numerical Algorithms & Software for Extreme-Scale Science**	Lois Curfman McInnes, ANL
	… with hands-on lessons throughout the day for various topics (lead: Mark C. Miller, LLNL)	
10:30 am	Break	
11:00 am	**Structured/Adaptive Meshing and Discretization**	Ann Almgren and Don Willcox, LBNL
12:00 pm	**Time Integration and Nonlinear Solvers** (part 1)	Dan Reynolds, SMU
12:30 pm	Lunch	
1:30 pm	**Time Integration and Nonlinear Solvers** (part 2)	Dan Reynolds, SMU
2:00 pm	**Krylov Solvers and Preconditioning**	Jonathan Hu and Christian Glusa, SNL
3:00 pm	Break	
3:30 pm	**Panel: Extreme-Scale Algorithms & Software**	Track 5 Team
4:25 pm	**Numerical Optimization**	Alp Dener, ANL
5:10 pm	**Putting It All Together**	Ann Almgren, LBNL
5:30 pm	Dinner [+ Dinner talk]	Mark C. Miller, LLNL
6:30 pm	**Hands-on Deep Dives ... 1-on-1 Discussions … Prizes!**	Track 5 Team

+ Hands-on lessons

Additional contributors to lectures and hands-on lessons:

Satish Balay (ANL), Aaron Fisher (LLNL)

Additional contributors to gallery of highlights:

Various HPC package developers

See also Track 7:

Software Productivity (Aug 8)

Track 5: Numerical Algorithms and Software: Tutorial Goals

1. 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)

2. Provide an overview of software tools available to perform these tasks on HPC architectures … including where to go for more info

3. Practice using one or more of these software tools on basic demonstration problems

This presentation gives a high-level introduction to HPC numerical software

- How HPC numerical software addresses challenges in computational science and engineering (CSE)
- Toward extreme-scale scientific software ecosystems
- Using and contributing: Where to go for more info

Why is this important for you?

- Libraries enable users to focus on their primary interests
  - Reuse algorithms and data structures developed by experts
  - Customize and extend to exploit application-specific knowledge
  - Cope with complexity and changes over time
- More efficient, robust, reliable, scalable, sustainable scientific software
- Better science, broader impact of your work

This work is founded on decades of experience and concerted team efforts to advance numerical software …

https://exascaleproject.org

https://fastmath-scidac.org

- Exascale Computing Project
- FASTMath SciDAC Institute
- Developers of xSDK packages

Community efforts: Join us!

https://xsdk.info

https://bssw.io

… While improving software productivity & sustainability as key aspects of advancing overall scientific productivity

https://ideas-productivity.org

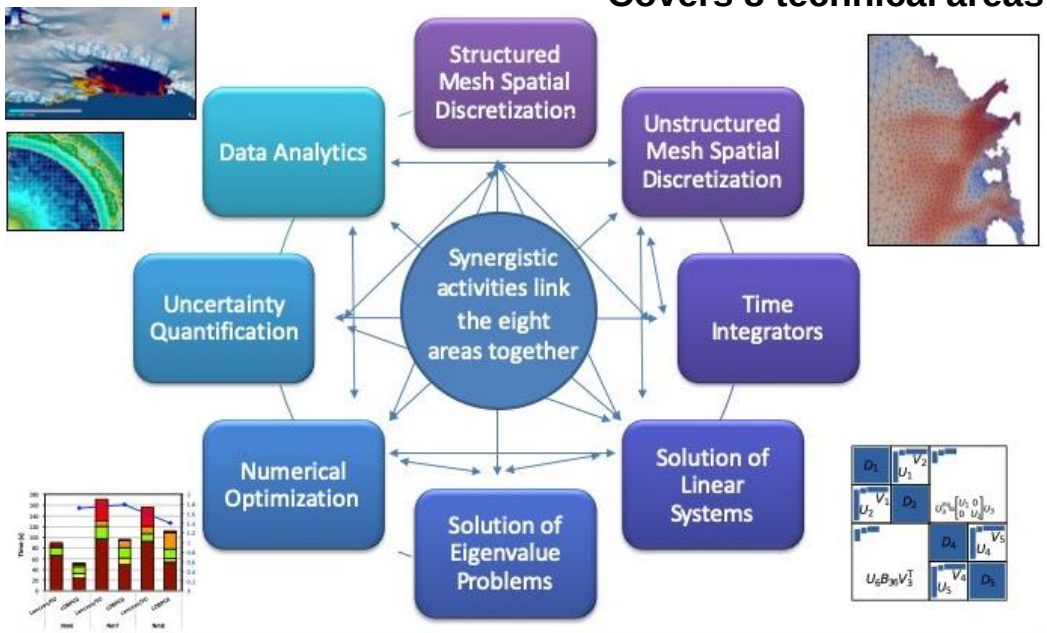
- IDEAS Software Productivity Project
- Better Scientific Software Community

See also Track 7: Software Productivity (Aug 8)

FASTMath: Frameworks, Algorithms & Scalable Technologies for Mathematics

https://fastmath-scidac.org

Covers 8 technical areas



FASTMath Goals:

- Develop advanced numerical techniques for DOE applications
- Deploy high-performance software on DOE supercomputers
- Demonstrate basic research technologies from applied mathematics
- Engage and support of the computational science community

50+ researchers from 5 DOE labs and 5 universities

Berkeley Lab, Argonne National Laboratory, Lawrence Livermore National Laboratory, Oak Ridge National Laboratory, Sandia National Laboratories

CU Boulder, USC, MIT, Rensselaer, SMU

100's of person years of experience building math software

SuperLU, AMReX, sundials, ZOLTAN, mfem, PETSc, hypre, UQTk, symPACK, PUMI, Trilinos, Albany, DAKOTA

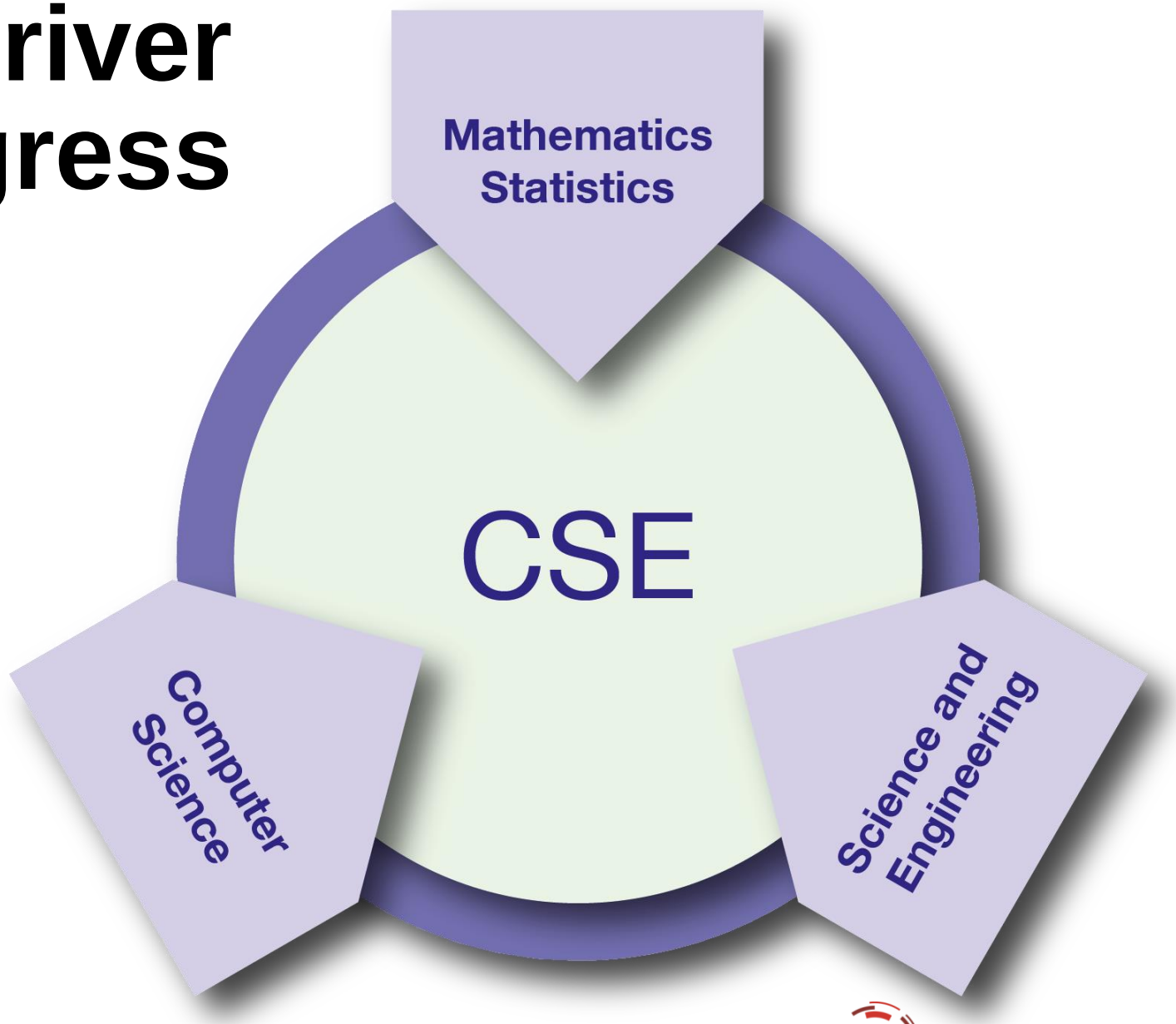
Esmond G. Ng, FASTMath Institute Director (EGNg@lbl.gov)

CSE: Essential driver of scientific progress

CSE = Computational Science & Engineering

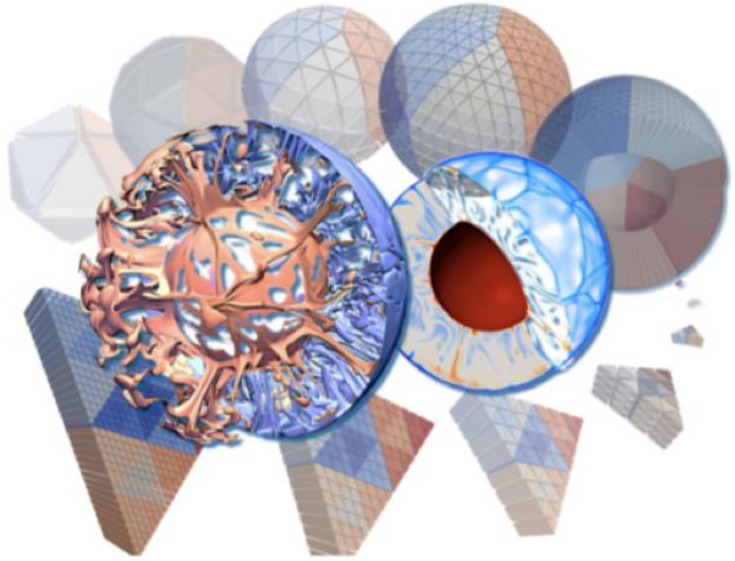
Development and use of computational methods for scientific discovery

- all branches of the sciences
- engineering and technology
- support of decision-making across a spectrum of societally important applications



Rapidly expanding role of CSE: New directions toward predictive science

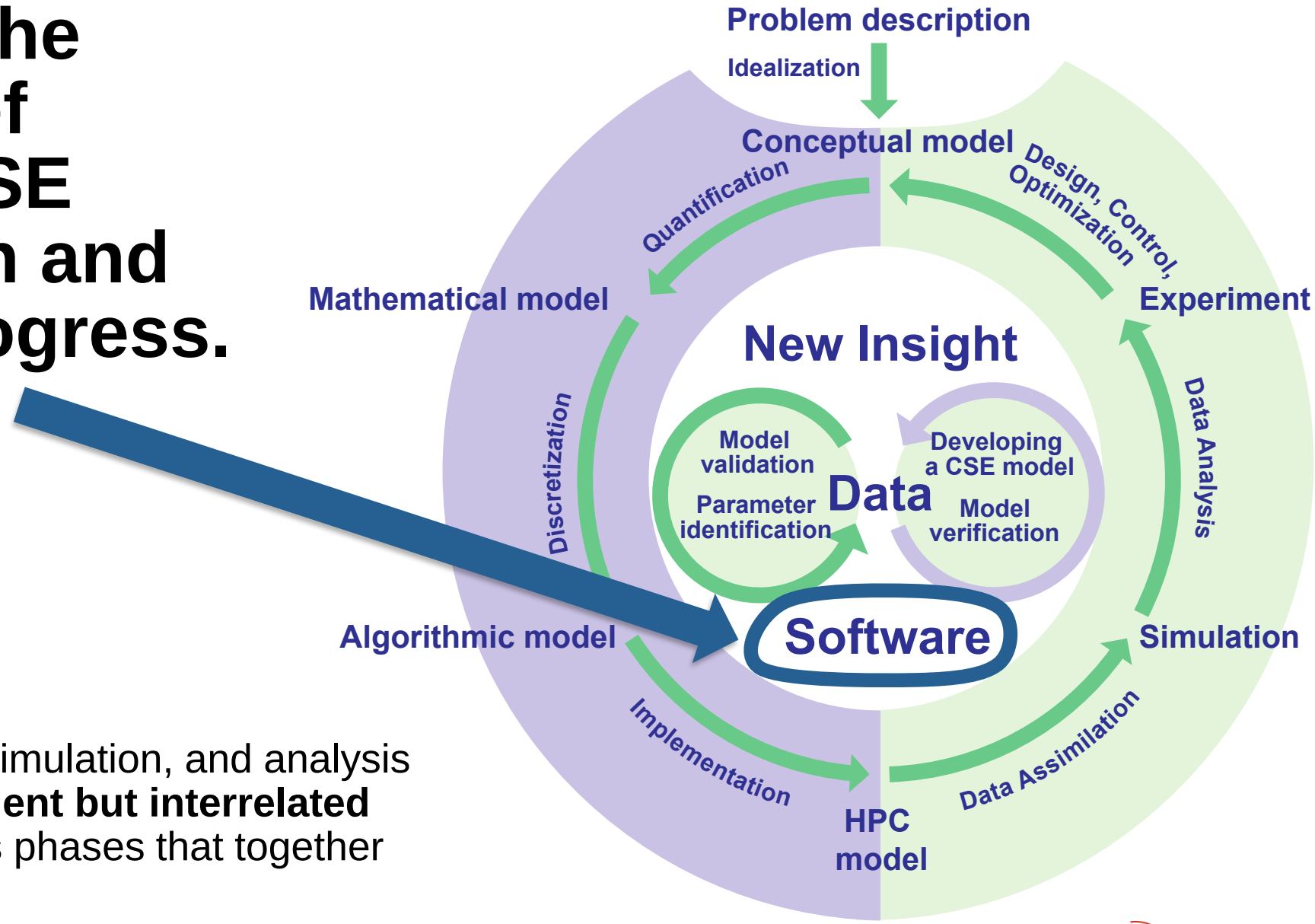
- Mathematical methods and algorithms
- CSE and HPC: Ubiquitous parallelism
- CSE and the data revolution
- CSE software
- CSE education & workforce development



Research and Education in Computational Science & Engineering

U. Rüde, K. Willcox, L.C. McInnes, H. De Sterck, G. Biros, H. Bungartz, J. Corones, E. Cramer, J. Crowley, O. Ghattas, M. Gunzburger, M. Hanke, R. Harrison, M. Heroux, J. Hesthaven, P. Jimack, C. Johnson, K. Jordan, D. Keyes, R. Krause, V. Kumar, S. Mayer, J. Meza, K.M. Mørken, J.T. Oden, L. Petzold, P. Raghavan, S. Shontz, A. Trefethen, P. Turner, V. Voevodin, B. Wohlmuth, C.S. Woodward, ***SIAM Review**, 60(3),* Aug 2018, https://doi.org/10.1137/16M1096840.

Software is the foundation of sustained CSE collaboration and scientific progress.

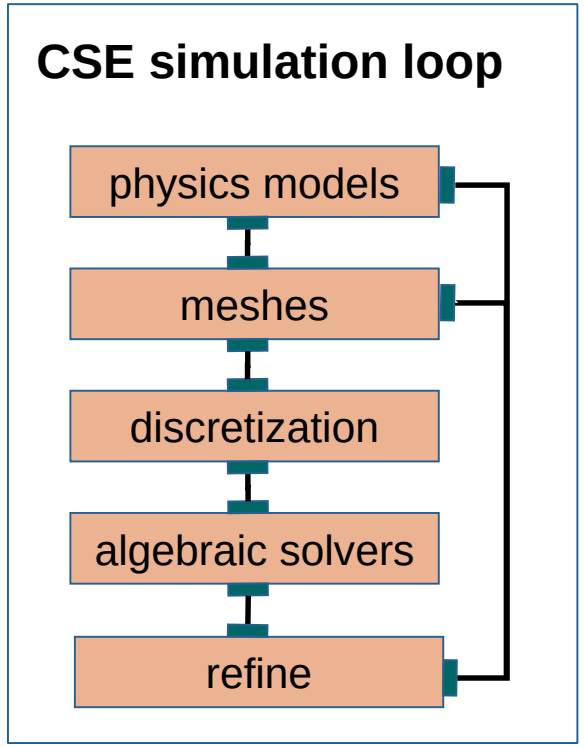


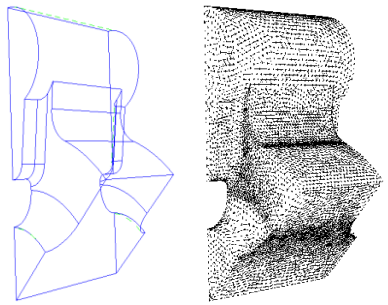
CSE cycle: Modeling, simulation, and analysis

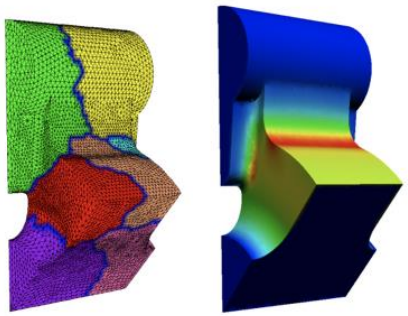
- **Software: independent but interrelated elements** for various phases that together enable CSE

CSE simulation starts with a forward simulation that captures the physical phenomenon of interest

- 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
- Incorporate different physics, scales

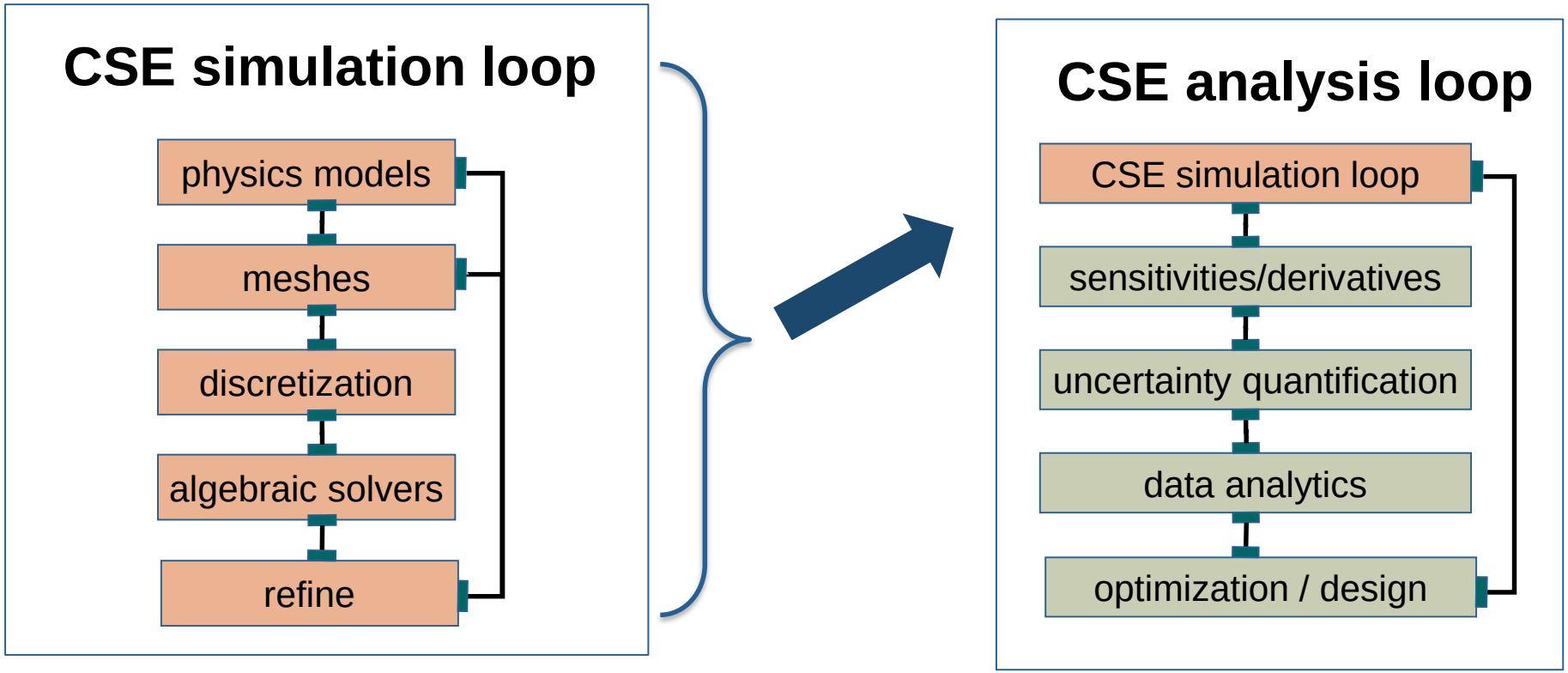






Requires: mesh generation, partitioning, load balancing, high-order discretization, time integration, linear & nonlinear solvers, eigensolvers, mesh refinement, multiscale/multiphysics coupling, etc.

CSE analysis builds on the CSE simulation loop … and relies on even more numerical algorithms and software



Beyond interpretive simulations …

working toward predictive science

Requires: adjoints, sensitivities, algorithmic differentiation, sampling, ensembles, data analytics, uncertainty quantification, optimization (derivative free & derivative based), inverse problems, 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

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

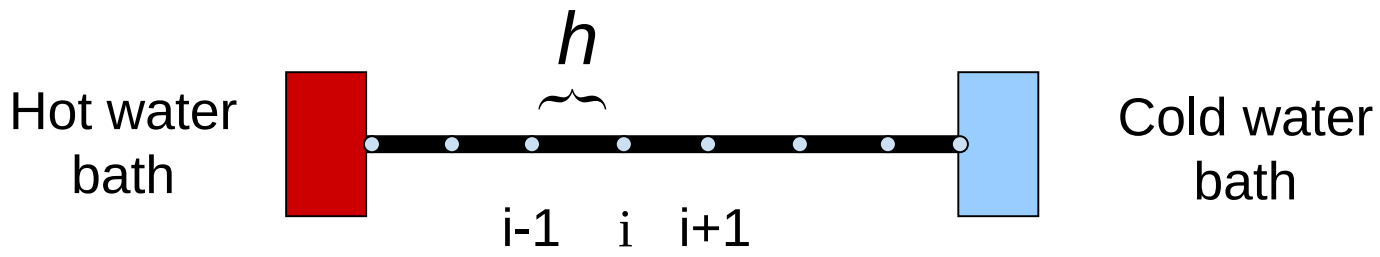
$$T(0) = 180^o \quad T(1) = 0^o$$

The first step is to discretize the equations

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

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

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



Then you can solve for the unknowns T_i

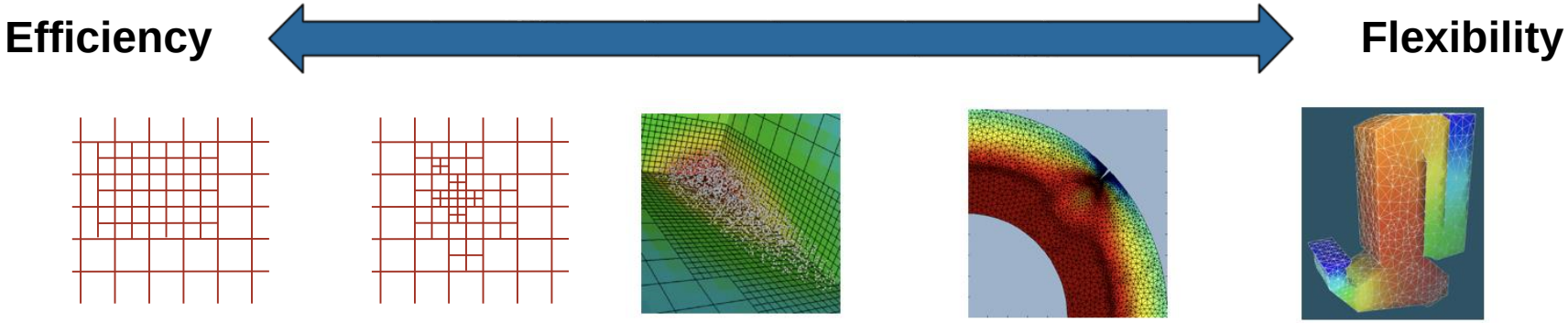
- Set up a matrix of the unknown coefficients
  - include the known boundary conditions
- Solve the linear system for T_i

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

- Visualize and analyze the results

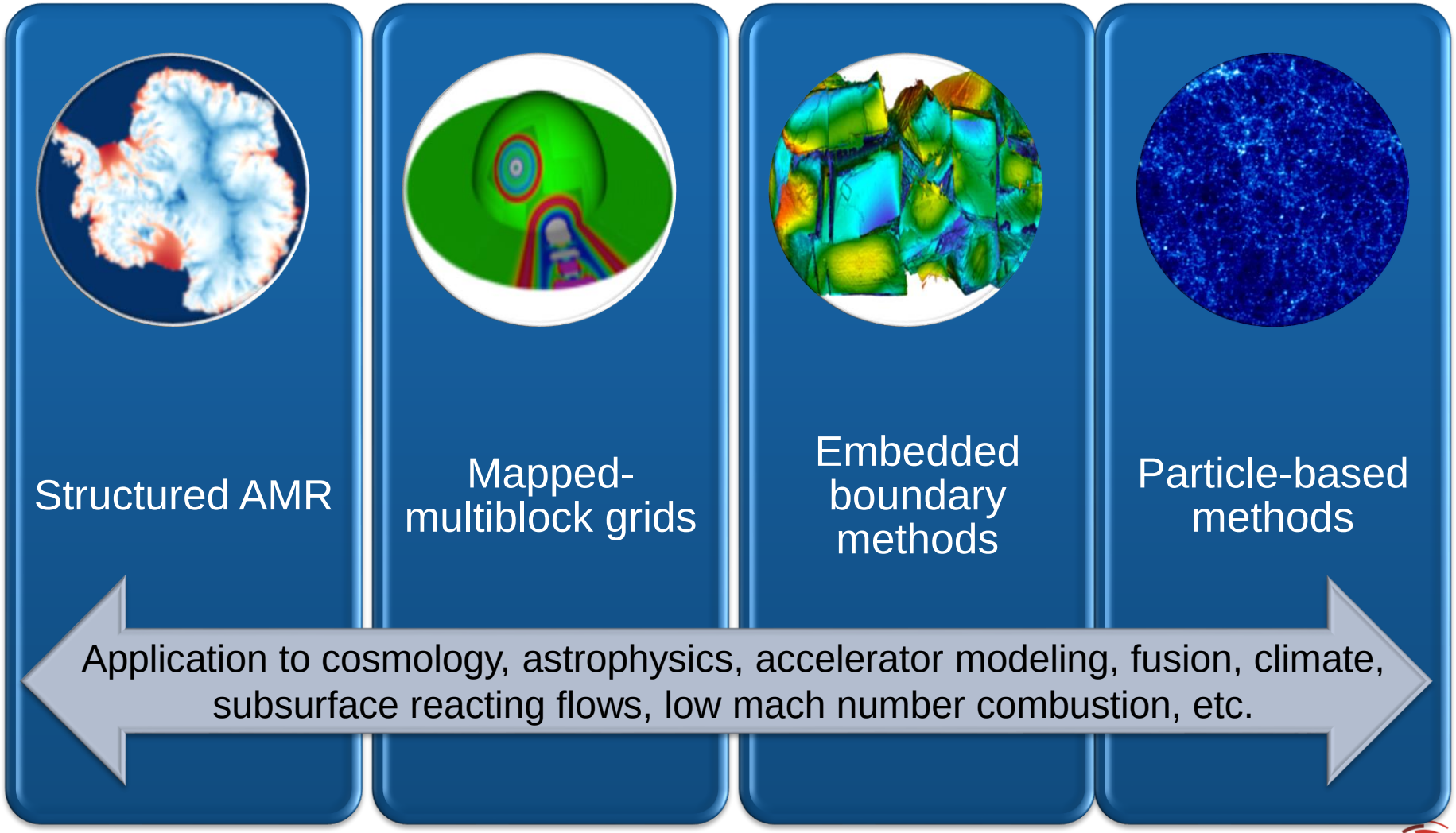
As problems get more complicated, so do the steps in the process

- Different discretization strategies exist for differing needs

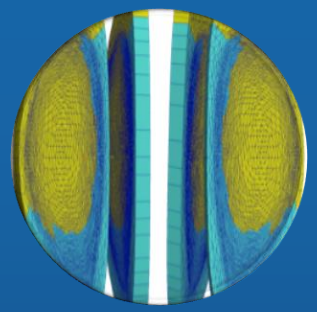


- Most problems are time dependent and nonlinear
  - Need higher algorithmic levels than linear solvers
- Increasingly combining multiple physical processes
  - Interactions require careful handling
- Goal-oriented problem solving requires optimization, uncertainty quantification

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

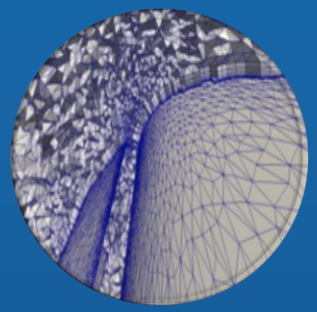


Unstructured grid capabilities focus on adaptivity, high-order, and the tools needed for extreme scaling

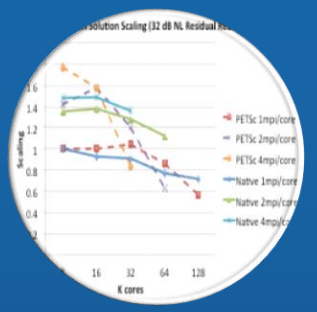


Parallel mesh infrastructures

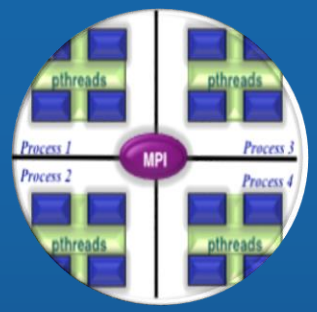
Dynamic load balancing



Mesh adaptation and quality control



Parallel performance on unstructured meshes

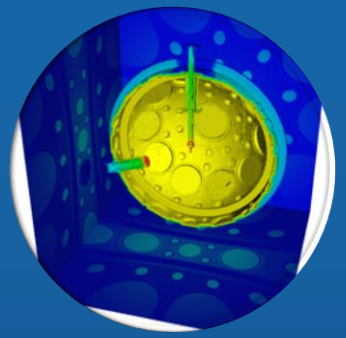


Architecture aware implementations

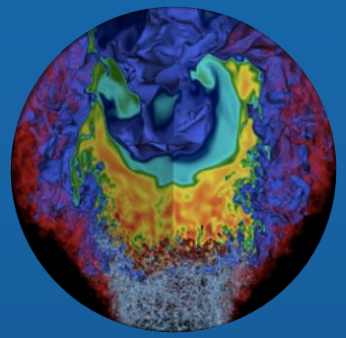
Application to fusion, climate, accelerator modeling, NNSA applications, nuclear energy, manufacturing processes, etc.

See presentations + hands-on lesssons @ ATPESC 2018

Time discretization methods provide efficient and robust techniques for stiff implicit, explicit and multi-rate systems

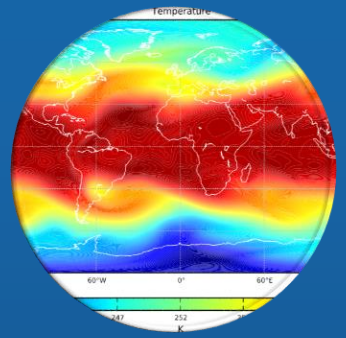


IMEX, multi-rate methods



Spectral deferred correction techniques

Parallel in time

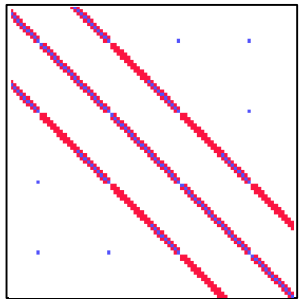


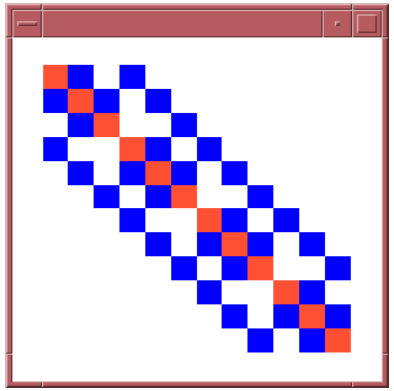
Constrained problems

Application to atmospheric dynamics, ice sheets, combustion, power grid, cosmology, subsurface, etc.

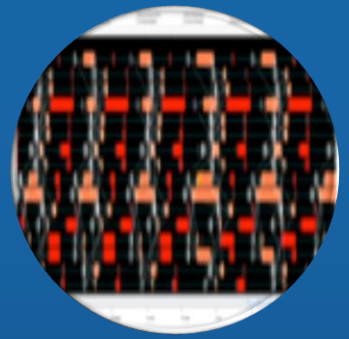
As problems grow in size, so do corresponding discrete systems

- 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 deliver this functionality as numerical libraries
  - hypre, PETSc, SuperLU, Trilinos, etc.

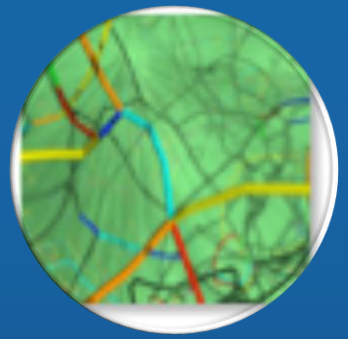




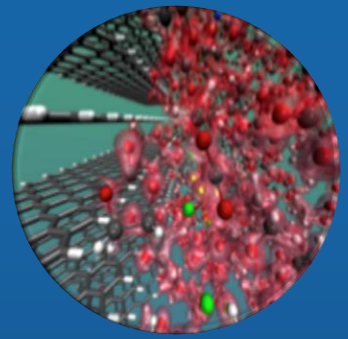
Research on algebraic systems provides key solution technologies to applications



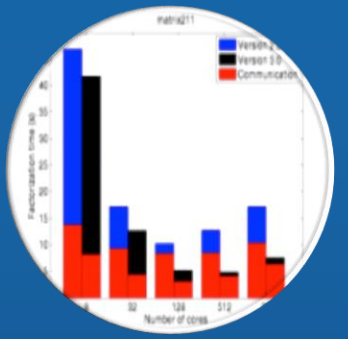
Linear system solution using direct and iterative solvers



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,

Disruptive changes in HPC architectures

- **Extreme levels of concurrency**
  - Increasingly deep memory hierarchies
  - Very high node and core counts
- **Additional complexities**
  - Hybrid architectures
  - GPUs, multithreading, manycore
  - Relatively poor memory latency and bandwidth
  - Challenges with fault resilience
  - Must conserve power – limit data movement
  - New (not yet stabilized) programming models
  - Etc.

ANL: Intel/Cray KNL, 2016

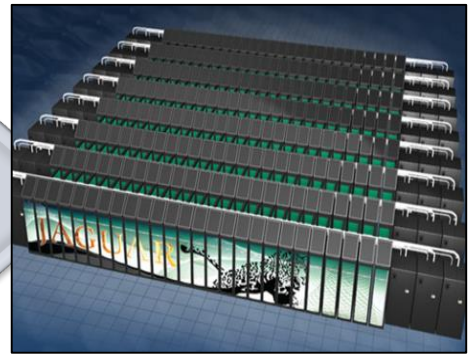
LBNL: Cray/Intel Xeon/KNL, 2016

ORNL: IBM/NVidia P9/Volta, 2018

Research to improve performance on HPC platforms focuses on inter- and intra-node issues

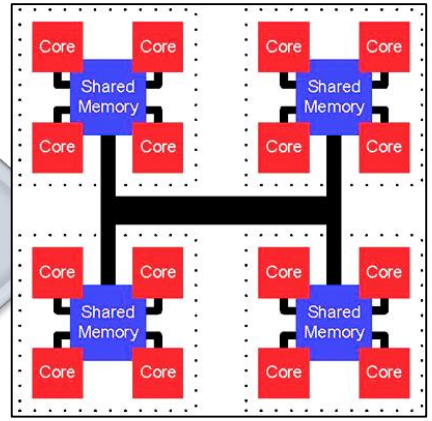
Inter-node: Massive Concurrency

- 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



Software libraries facilitate progress in computational science and engineering

- **Software library:** a high-quality, encapsulated, documented, tested, and multiuse software collection that provides functionality commonly needed by application developers
  - Organized for the purpose of being reused by independent (sub)programs
  - User needs to know only
    - Library interface (not internal details)
    - When and how to use library functionality appropriately

- **Key advantages** of software libraries
  - Contain complexity
  - Leverage library developer expertise
  - Reduce application coding effort
  - Encourage sharing of code, ease distribution of code

- **References:**
  - https://en.wikipedia.org/wiki/Library_(computing)
  - What are Interoperable Software Libraries? Introducing the xSDK

Broad range of HPC numerical software

Some packages with general-purpose, reusable algorithmic infrastructure in support of high-performance CSE:

- ★ **AMReX** – https://github.com/AMReX-codes/amrex
- ★ **Chombo** - https://commons.lbl.gov/display/chombo
- **Clawpack** - http://www.clawpack.org
- ★ **Deal.II** - https://www.dealii.org
- **FEniCS** - https://fenicsproject.org
- ★ **hypre** - http://www.llnl.gov/CASC/hypre
- **libMesh** - https://libmesh.github.io
- ★ **MAGMA** - http://icl.cs.utk.edu/magma
- ★ **MFEM** - http://mfem.org
- ★ **PETSc/TAO** – http://www.mcs.anl.gov/petsc
- ★ **PUMI** - http://github.com/SCOREC/core
- ★ **SUNDIALS** - http://computation.llnl.gov/casc/sundials
- ★ **SuperLU** - http://crd-legacy.lbl.gov/~xiaoye/SuperLU
- ★ **Trilinos** - https://trilinos.org
- **Uintah** - http://www.uintah.utah.edu
- **waLBerla** - http://www.walberla.net

See info about scope, performance, usage, and design, including:

- tutorials
- demos
- examples
- how to contribute

★ Discussed today: Gallery of highlights

… and many, many more ... Explore, use, contribute!

ECP applications need sustainable coordination among ECP math libraries

- **ECP application team interviews:**

  Astro, NWChemEx, WDMAPP, ExaFel, GAMESS, ExaSGD, Subsurface, EXAALT, WarpX, ExaAM, MFIX-Exa, ATDM (LANL, LLNL, SNL) apps, CoPA, AMREX, CEED, CODAR

- **Many ECP app teams rely on math libraries, often in combination**

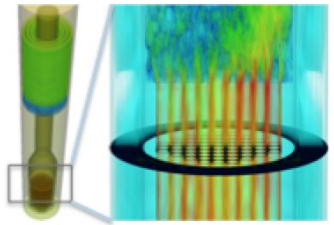
  Need **sustainable coordinated xSDK releases** and **increasing interoperability** among xSDK packages to achieve good performance and easily access advanced algorithms and data structures

hypre high performance preconditioners
PETSc/TAO
SuperLU
TRILINOS
MAGMA
MFEM
sundials
DTK
PLASMA
TASMANIAN
FleCSI
STRUMPACK
AMReX

Multiphysics: A primary motivator for exascale

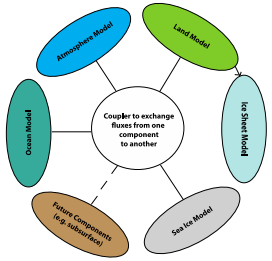
Multiphysics: greater than 1 component governed by its own principle(s) for evolution or equilibrium

– Also: broad class of coarsely partitioned problems possess similarities

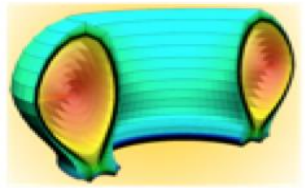


nuclear reactors
A. Siegel, ANL

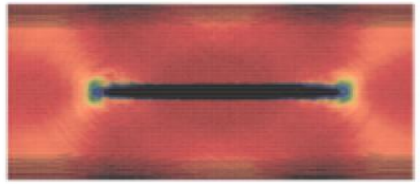
particle accelerators
K. Lee, SLAC



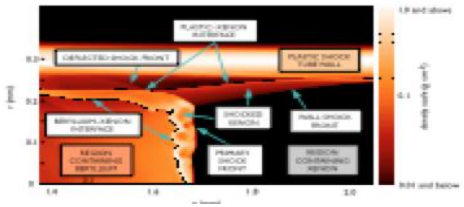
climate
K. Evans, ORNL



fusion
A. Hakim, PPPL



crack propagation
E. Kaxiras, Harvard



radiation hydrodynamics
E. Myra, Univ. of Michigan

IJHPCA, Feb 2013
Vol 27, Issue 1, pp. 4-83

THE INTERNATIONAL JOURNAL of HIGH PERFORMANCE COMPUTING APPLICATIONS

The International Journal of High Performance Computing Applications
27(1) 4–83

DOI: 10.1177/1094342012468181
hpc.sagepub.com
SAGE

Multiphysics simulations: Challenges and opportunities

David E Keyes[1,2], Lois C McInnes[3], Carol Woodward[4], William Gropp[5], Eric Myra[6], Michael Pernice[7], John Bell[8], Jed Brown[3], Alain Clo[1], Jeffrey Connors[4], Emil Constantinescu[3], Don Estep[9], Kate Evans[10], Charbel Farhat[11], Ammar Hakim[12], Glenn Hammond[13], Glen Hansen[14], Judith Hill[10], Tobin Isaac[15], Xiangmin Jiao[16], Kirk Jordan[17], Dinesh Kaushik[3], Efthimios Kaxiras[18], Alice Koniges[8], Kihwan Lee[19], Aaron Lott[4], Qiming Lu[20], John Magerlein[17], Reed Maxwell[21], Michael McCourt[22], Miriam Mehl[23], Roger Pawlowski[14], Amanda P Randles[18], Daniel Reynolds[24], Beatrice Rivière[25], Ulrich Rüde[26], Tim Scheibe[13], John Shadid[14], Brendan Sheehan[9], Mark Shephard[27], Andrew Siegel[3], Barry Smith[3], Xianzhu Tang[28], Cian Wilson[2] and Barbara Wohlmuth[23]

doi:10.1177/1094342012468181

Software libraries are not enough

Apps need to use software packages **in combination**

> **"The way you get programmer productivity is by eliminating lines of code you have to write."**
>
> – Steve Jobs, Apple World Wide Developers Conference, Closing Keynote, 1997

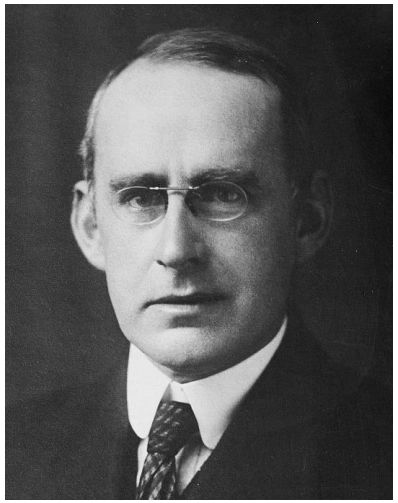
- **Need consistency** of compiler (+version, options), 3rd-party packages, etc.
- **Namespace and version conflicts** make simultaneous build/link of packages difficult
- **Multilayer interoperability** requires careful design and sustainable coordination

Need software ecosystem perspective

Ecosystem: A group of independent but interrelated elements comprising a unified whole

Ecosystems are challenging!

"We often think that when we have completed our study of one we know all about two, because 'two' is 'one and one.' We forget that we still have to make a study of 'and.' "



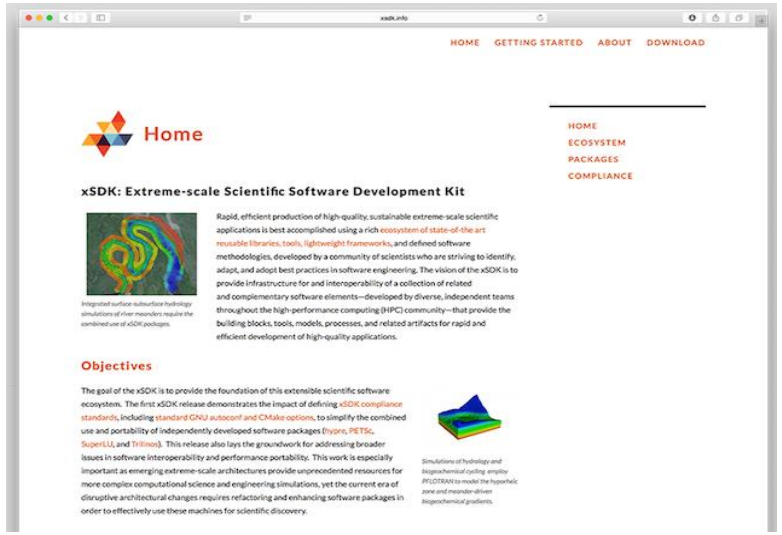
− Sir Arthur Stanley Eddington (1892−1944), British astrophysicist

Building the foundation of a highly effective extreme-scale scientific software ecosystem

Focus: Increasing the functionality, quality, and interoperability of important scientific libraries, domain components, and development tools

Impact:

- Improved code quality, usability, access, sustainability
- Inform potential users that an xSDK member package can be easily used with other xSDK packages
- Foundation for work on performance portability ,deeper levels of package interoperability

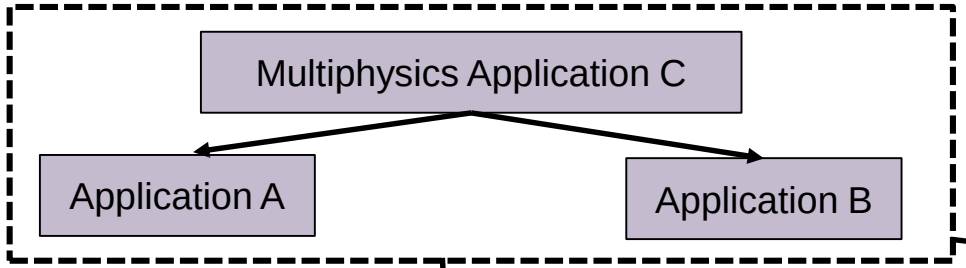


website: xSDK.info

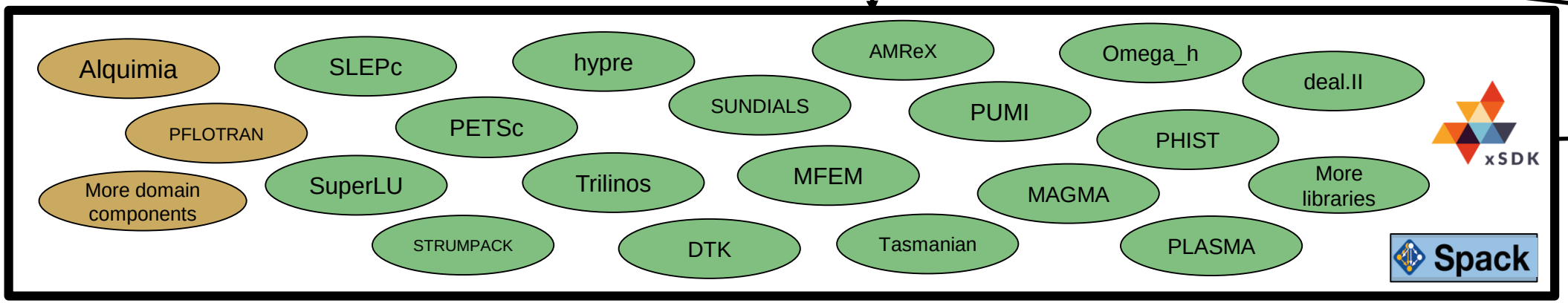
xSDK version 0.4.0: December 2018

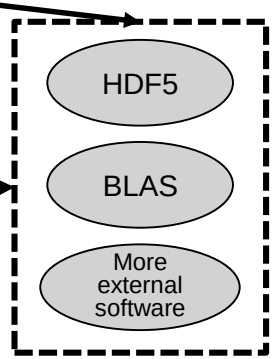
https://xsdk.info

Each xSDK member package uses or can be used with one or more xSDK packages, and the connecting interface is regularly tested for regressions.



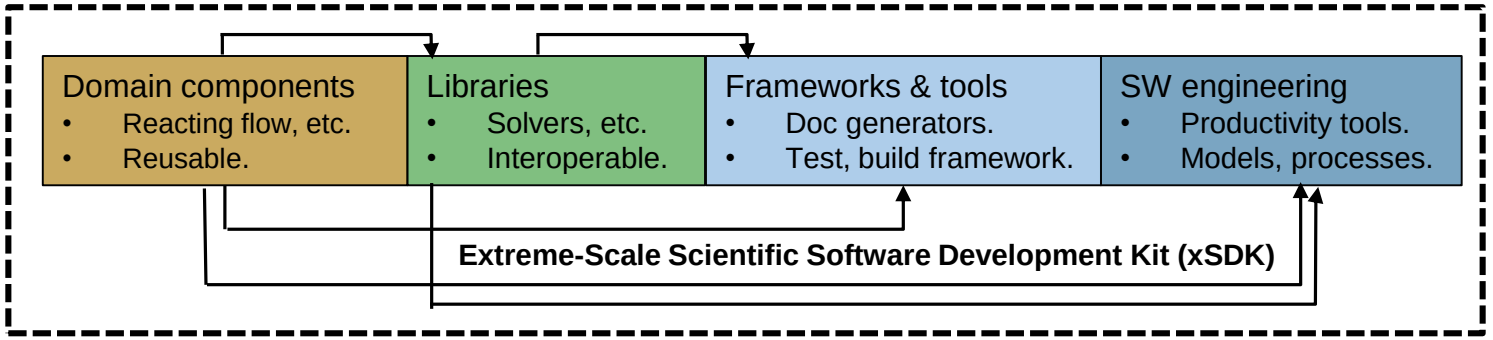
Tested on key machines at ALCF, NERSC, OLCF, also Linux, Mac OS X





December 2018

- 17 math libraries
- 2 domain components
- 16 mandatory xSDK community policies
- Spack xSDK installer



Impact: Improved code quality, usability, access, sustainability

Foundation for work on performance portability, deeper levels of package interoperability

xSDK collaborators

xSDK Release 0.4.0, Dec 2018

- **xSDK release lead:** Jim Willenbring, SNL
- **xSDK planning**
  - Lois Curfman McInnes (ANL)
  - Ulrike Meier Yang (LLNL)
- **Leads for xSDK testing**
  - Satish Balay, ANL: ALCF testing
  - Piotr Luszczek, UTK: OLCF testing
  - Aaron Fischer, LLNL: NERSC testing
  - Cody Balos, LLNL: general testing
  - Keita Teranishi, SNL: general testing
- **Spack liaison:** Todd Gamblin, LLNL

- **Package compatibility with xSDK community policies and software testing:**
  - **AMReX**: Ann Almgren, Michele Rosso (LBNL)
  - **DTK**: Stuart Slattery, Bruno Turcksin (ORNL)
  - **deal.II**: Wolfgang Bangerth (Colorado State University)
  - **hypre**: Ulrike Meier Yang, Sarah Osborn, Rob Falgout (LLNL)
  - **MAGMA** and **PLASMA**: Piotr Luszczek (UTK)
  - **MFEM**: Aaron Fischer, Tzanio Kolev (LLNL)
  - **Omega_h**: Dan Ibanez (SNL)
  - **PETSc/TAO**: Satish Balay, Alp Denner, Barry Smith (ANL)
  - **PUMI**: Cameron Smith (RPI)
  - **SUNDIALS**: Cody Balos, David Gardner, Carol Woodward (LLNL)
  - **SuperLU** and **STRUMPACK**: Sherry Li and Pieter Ghysels (LBNL)
  - **TASMANIAN**: Miroslav Stoyanov, Damien Lebrun Grandie (ORNL)
  - **Trilinos**: Keita Teranishi, Jim Willenbring, Sam Knight (SNL)
  - **PHIST**: Jonas Thies (DLR, German Aerospace Center)
  - **SLEPc**: José Roman (Universitat Politècnica de València)
  - **Alquimia**: Sergi Mollins (LBNL)
  - **PFLOTRAN**: Glenn Hammond (SNL)

and many more …

xSDK community policies

https://xsdk.info/policies

Version 0.5.0, July 2019

xSDK compatible package: Must satisfy mandatory xSDK policies:

M1. Support xSDK community GNU Autoconf or CMake options.
M2. Provide a comprehensive test suite.
M3. Employ user-provided MPI communicator.
M4. Give best effort at portability to key architectures.
M5. Provide a documented, reliable way to contact the development team.
M6. Respect system resources and settings made by other previously called packages.
M7. Come with an open source license.
M8. Provide a runtime API to return the current version number of the software.
M9. Use a limited and well-defined symbol, macro, library, and include file name space.
M10. Provide an accessible repository (not necessarily publicly available).
M11. Have no hardwired print or IO statements.
M12. Allow installing, building, and linking against an outside copy of external software.
M13. Install headers and libraries under <prefix>/include/ and <prefix>/lib/.
M14. Be buildable using 64 bit pointers. 32 bit is optional.
M15. All xSDK compatibility changes should be sustainable.
M16. The package must support production-quality installation compatible with the xSDK install tool and xSDK metapackage.

Also **recommended policies**, which currently are encouraged but not required:

R1. Have a public repository.
R2. Possible to run test suite under valgrind in order to test for memory corruption issues.
R3. Adopt and document consistent system for error conditions/exceptions.
R4. Free all system resources it has acquired as soon as they are no longer needed.
R5. Provide a mechanism to export ordered list of library dependencies.
R6. Provide versions of dependencies.

xSDK member package: Must be an xSDK-compatible package, *and* it uses or can be used by another package in the xSDK, and the connecting interface is regularly tested for regressions.

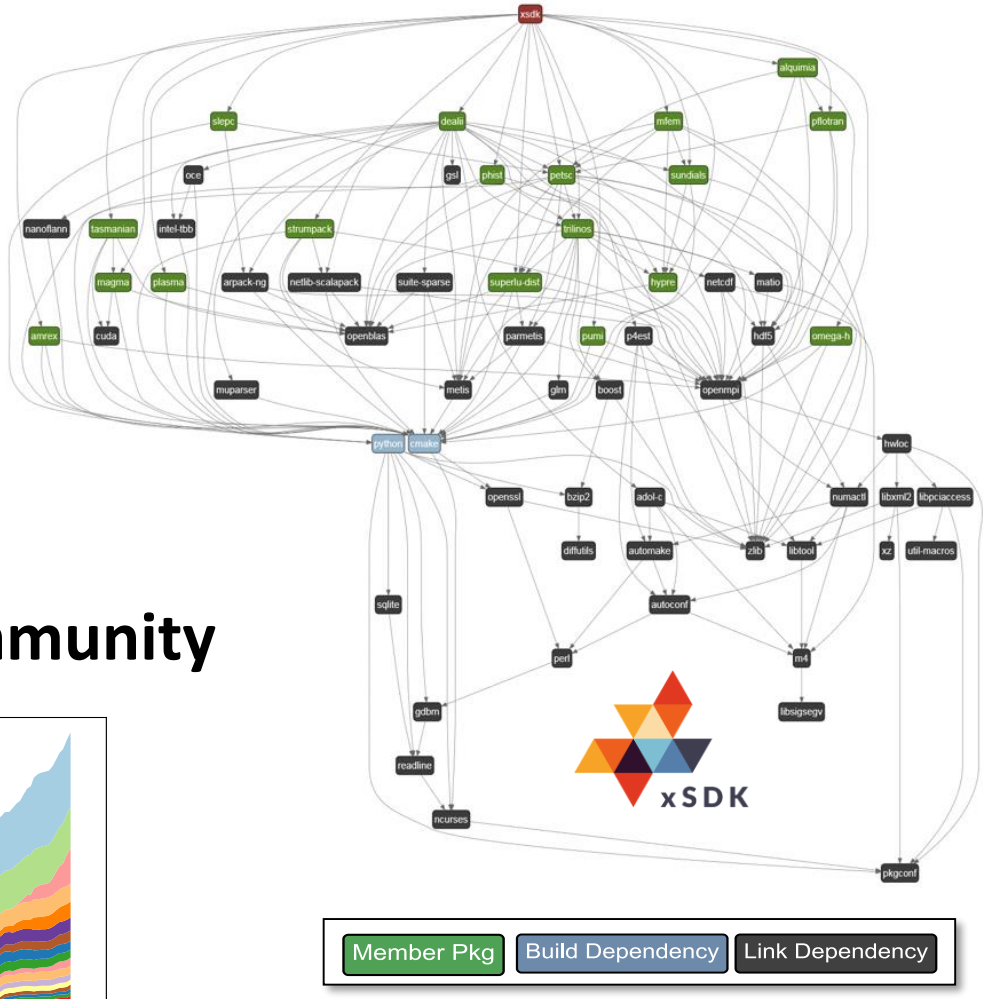
We welcome feedback. What policies make sense for your software?

EXASCALE COMPUTING PROJECT

The xSDK is using Spack to deploy its software

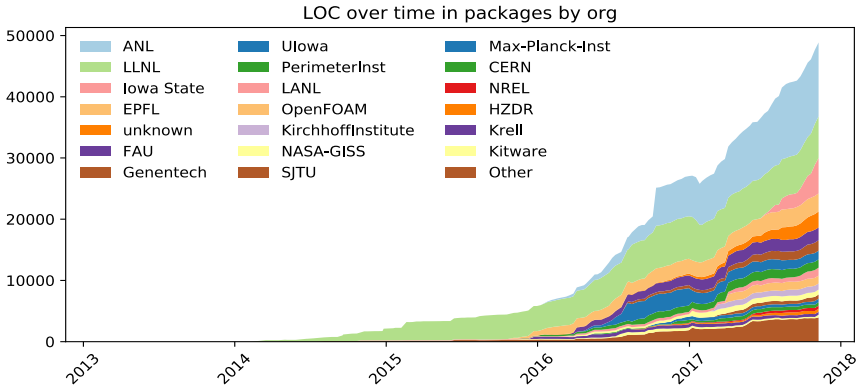
- The xSDK packages depend on a number of open source libraries
- Spack is a flexible package manager for HPC
- Spack allows the xSDK to be deployed with a single command
  - User can optionally choose compilers, build options, etc.
  - Will soon support combinatorial test dashboards for xSDK packages

Spack



Spack has grown into a thriving open source community

- Over 300 contributors
- Over 2,800 software packages
- Key component of ECP strategy for software deployment

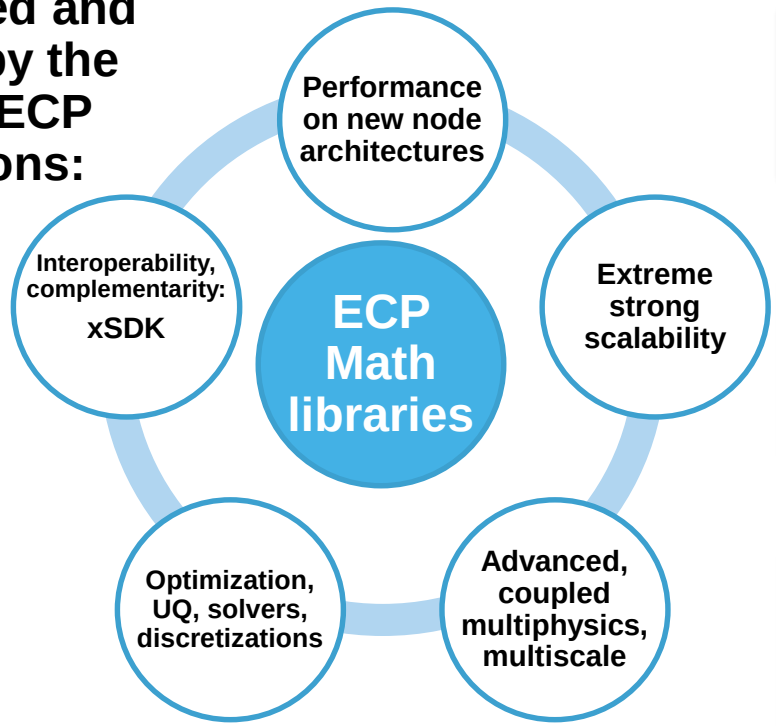


https://spack.io github.com/spack

EXASCALE COMPUTING PROJECT

xSDK: Primary delivery mechanism for ECP math libraries' continual advancements toward predictive science

As motivated and validated by the needs of ECP applications:



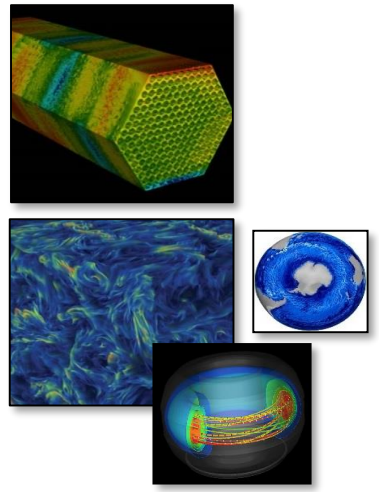
Next-generation algorithms

Advances in data structures for new node architectures

Improving library quality, sustainability, interoperability

Toward predictive scientific simulations

Increasing performance, portability, productivity



Timeline:

xSDK release 1 — xSDK release 2 — . . . . . — xSDK release n

Gallery of highlights

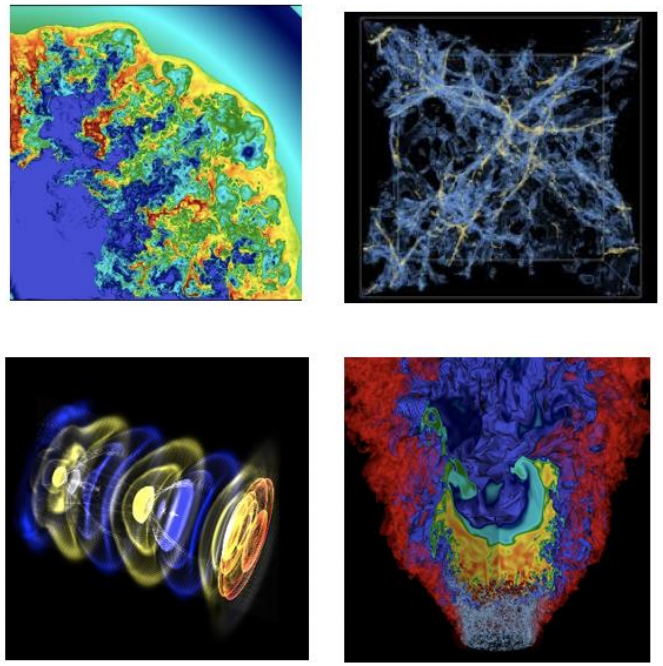
- Overview of HPC numerical software packages
- 1 slide per package, emphasizing key capabilities, highlights, and where to go for more info
  - Listed first
    - Packages featured in ATPESC 2019 lectures and hands-on lessons
    - Packages whose developers are available for evening discussions
  - Listed next
    - Additional highlighted packages (not a comprehensive list)

AMReX

Block-structured adaptive mesh refinement framework. Support for hierarchical mesh and particle data with embedded boundary capability.

- **Capabilities**
  - Support for solution of PDEs on hierarchical adaptive mesh with particles and embedded boundary representation of complex geometry
  - Support for multiple modes of time integration
  - Support for explicit and implicit single-level and multilevel mesh operations, multilevel synchronization, particle, particle-mesh and particle-particle operations
  - Hierarchical parallelism –
    - hybrid MPI + OpenMP with logical tiling on multicore architectures
    - hybrid MPI + GPU support for hybrid CPU/GPU systems (CUDA and beyond)
  - Native multilevel geometric multigrid solvers for cell-centered and nodal data
  - Highly efficient parallel I/O for checkpoint/restart and for visualization – native format supported by Visit, Paraview, yt
  - Tutorial examples available in repository
- **Open source software**
  - Used for divers apps, including accelerator modeling, adaptive manufacturing, astrophysics, combustion, cosmology, multiphase flow, phase field modeling, …
  - Freely available on github with extensive documentation

Examples of AMReX applications



FASTMATH xSDK ECP

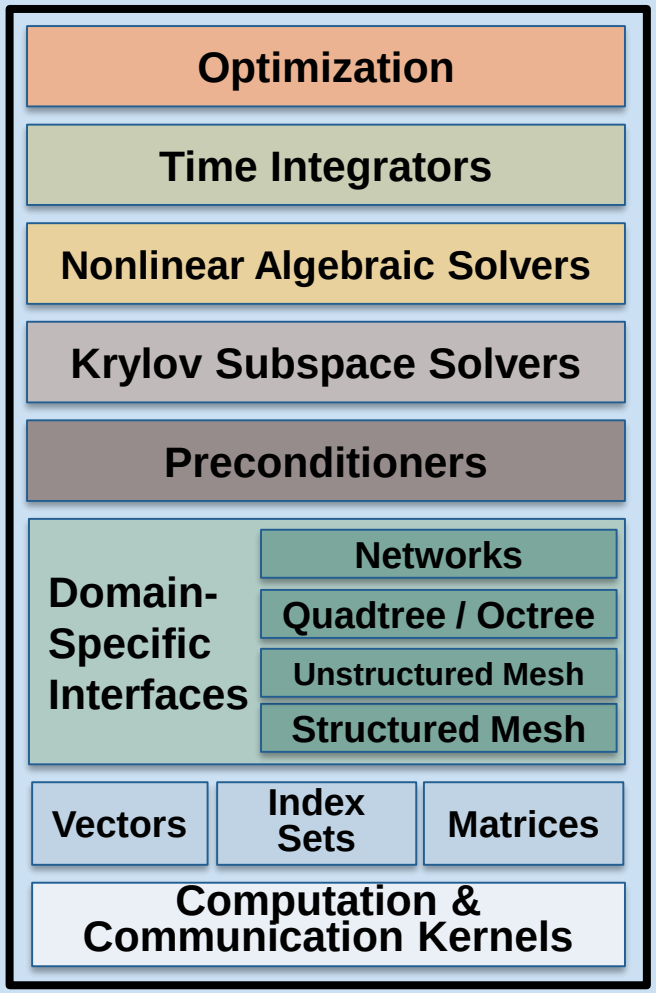
https://www.github.com/AMReX-Codes/amrex

ECP EXASCALE COMPUTING PROJECT

PETSc/TAO

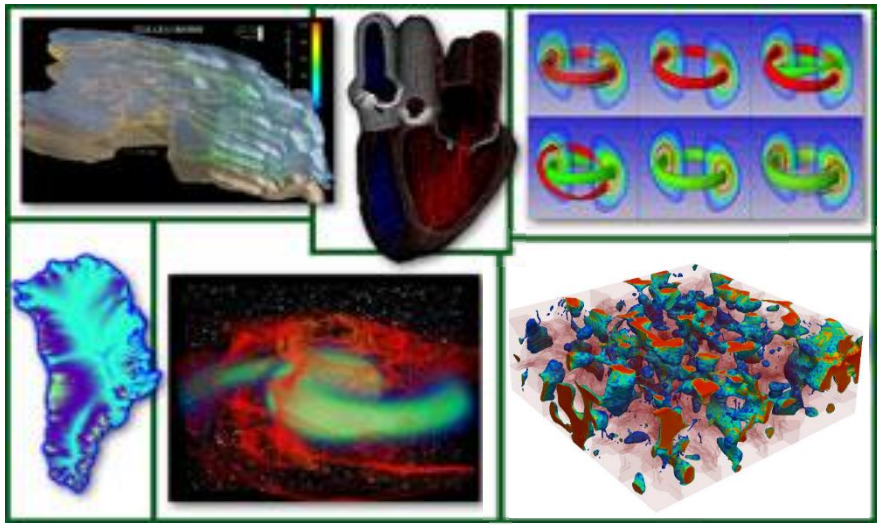
Portable, Extensible Toolkit for Scientific Computation / Toolkit for Advanced Optimization

Scalable algebraic solvers for PDEs. Encapsulate parallelism in high-level objects. Active & suppported user community. Full API from Fortran, C/C++, Python.



- **Easy customization and composability of solvers at runtime**
  - Enables optimality via flexible combinations of physics, algorithmics, architectures
  - Try new algorithms by composing new/existing algorithms (multilevel, domain decomposition, splitting, etc.)
- **Portability & performance**
  - Largest DOE machines, also clusters, laptops
  - Thousands of users worldwide

Argonne NATIONAL LABORATORY



PETSc provides the backbone of diverse scientific applications.

clockwise from upper left: hydrology, cardiology, fusion, multiphase steel, relativistic matter, ice sheet modeling

R&D 100

xSDK

FASTMATH

https://www.mcs.anl.gov/petsc

ECP EXASCALE COMPUTING PROJECT

SUNDIALS

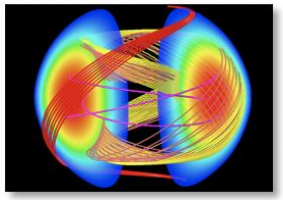
Suite of Nonlinear and Differential /Algebraic Equation Solvers

Adaptive time integrators for ODEs and DAEs and efficient nonlinear solvers
Used in a variety of applications. Freely available. Encapsulated solvers & parallelism.

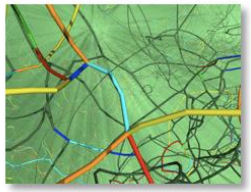
- **ODE integrators:**
  - CVODE: adaptive order and step BDF (stiff) & Adams (non-stiff) methods
  - ARKode: adaptive step implicit, explicit, IMEX, and multirate Runge-Kutta methods
- **DAE integrators**: IDA – adaptive order and step BDF integrators
- **Sensitivity Analysis:** CVODES and IDAS provide forward and adjoint sensitivity analysis capabilities for ODEs and DAEs respectively
- **Nonlinear Solvers:** KINSOL – Newton-Krylov, Picard, and accelerated fixed point
- **Modular Design:** Users can supply own data structures and solvers or use SUNDIALS provided modules
  - Written in C with interfaces to Fortran
  - Vectors modules: serial, MPI, OpenMP, CUDA, RAJA, hypre, PETSc, & Trilinos
- **Open Source:** Freely available (BSD License) from LLNL site, GitHub, and Spack. Can be used from MFEM, PETSc, and deal.II

Lawrence Livermore National Laboratory | FASTMATH | ECP | xSDK

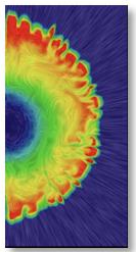
SUNDIALS is used by thousands worldwide in applications from research and industry



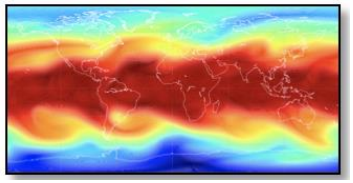
Magnetic Reconnection



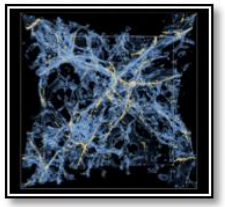
Dislocation Dynamics



Core Collapse Super-nova

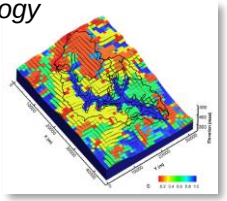


Atmospheric Dynamics



Cosmology

SUNDIALS is supported by extensive documentation, a user email list, and an active user community



Subsurface Flow

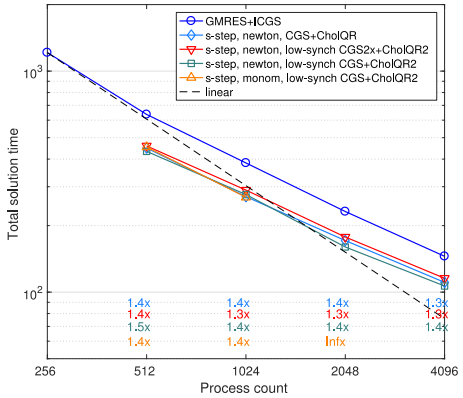
http://www.llnl.gov/casc/sundials

ECP EXASCALE COMPUTING PROJECT

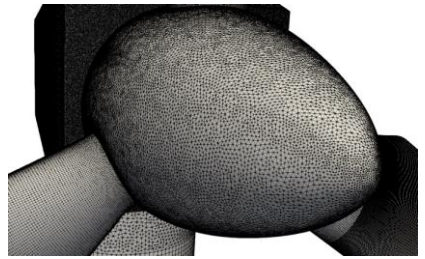
Trilinos/Belos

Iterative Krylov-based solvers. C++ permits one implementation that supports multiple scalar types and thread-parallel programming models.

- **Ability to solve single or sequence of linear systems**
  - Simultaneously solved systems w/ multiple-RHS: $AX = B$
  - Sequentially solved systems w/ multiple-RHS: $AX_i = B_i$, $i=1,\ldots,t$
  - Sequences of multiple-RHS systems: $A_iX_i = B_i$, $i=1,\ldots,t$
- **Standard methods**
  - Conjugate Gradients (CG), GMRES
  - TFQMR, BiCGStab, MINRES, fixed-point
- **Advanced methods**
  - Block GMRES, block CG/BICG
  - Hybrid GMRES, CGRODR (block recycling GMRES)
  - TSQR (tall skinny QR), LSQR
- **Ongoing research**
  - Communication avoiding methods
  - Pipelined and s-step methods



Speed-ups of various s-step Krylov methods within low Mach CFD wind-energy code Nalu-Wind.



Close-up of Vestas V27 hub section and nacelle.

Sandia National Laboratories

ASC

ECP EXASCALE COMPUTING PROJECT

SAND2019-8938 PE

Results courtesy of M. Hoemmen and I. Yamazaki (SNL). Image from Thomas et al, "High-fidelity simulation of wind-turbine incompressible flows", to appear in SISC, 2019.

https://trilinos.github.io/belos.html

ECP EXASCALE COMPUTING PROJECT

Trilinos/MueLu

Structured and unstructured aggregation-based algebraic multigrid (AMG) preconditioners

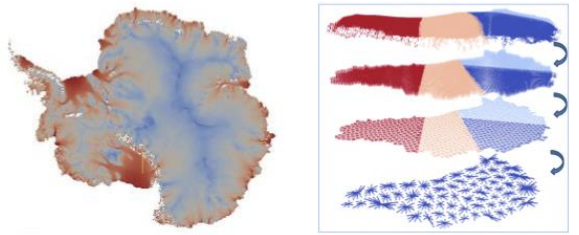
- **Robust, scalable, portable AMG preconditioning critical for many large-scale simulations**
  - Multifluid plasma simulations
  - Shock physics
  - Magneto-hydrodynamics (MHD)
  - Low Mach computational fluid dynamics (CFD)

- **Capabilities**
  - Aggregation-based coarsening
  - Smoothers: Jacobi, GS, /1 GS, polynomial, ILU, sparse direct
  - Load-balancing for good parallel performance

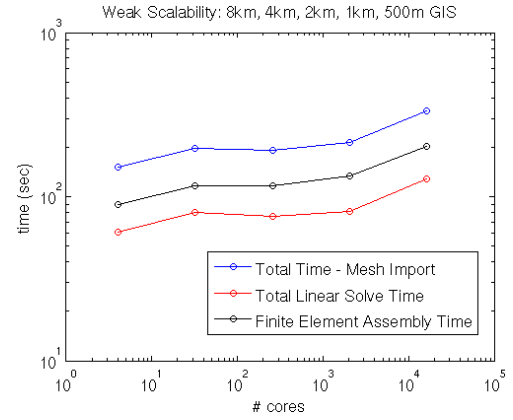
- **Research Areas**
  - Performance on next-generation architectures
  - AMG for multiphysics
  - Multigrid for coupled structured/unstructured meshes
  - Algorithm selection via machine learning

SAND2019-8938 PE

Image courtesy of Irina Tezaur (SNL). Semi-coarsening and scaling results courtesy Ray Tuminaro (SNL).



AMG operator-dependent semi-coarsening is key enabling technology in ASCR/BER ProSPect project's ice sheet simulations.



Sandia National Laboratories · SciDAC · ECP Exascale Computing Project · FASTMATH

https://trilinos.github.io/muelu.html

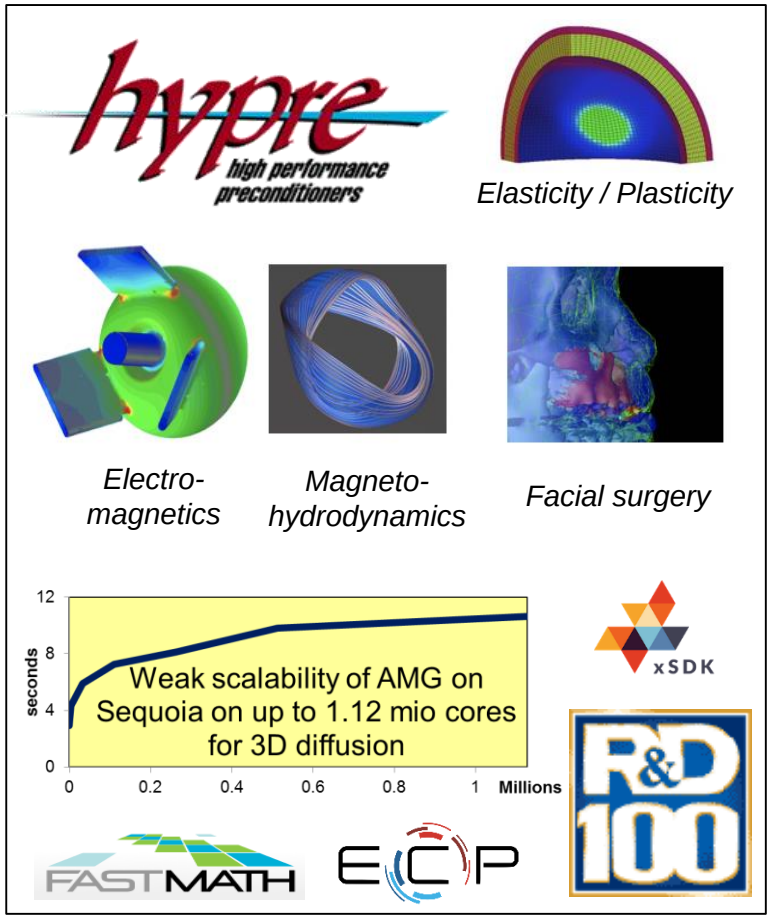
ECP Exascale Computing Project

hypre

Lawrence Livermore National Laboratory

Highly scalable multilevel solvers and preconditioners. Unique user-friendly interfaces. Flexible software design. Used in a variety of applications. Freely available.

- **Conceptual interfaces**
  - Structured, semi-structured, finite elements, linear algebraic interfaces
  - Provide natural "views" of the linear system
  - Provide for more efficient (scalable) linear solvers through more effective data storage schemes and more efficient computational kernels
- **Scalable preconditioners and solvers**
  - Structured and unstructured algebraic multigrid solvers
  - Maxwell solvers, H-div solvers
  - Multigrid solvers for nonsymmetric systems: pAIR, MGR
  - Matrix-free Krylov solvers
- **Open source software**
  - Used worldwide in a vast range of applications
  - Can be used through PETSc and Trilinos
  - Available on github: https://www.github.com/hypre-space/hypre



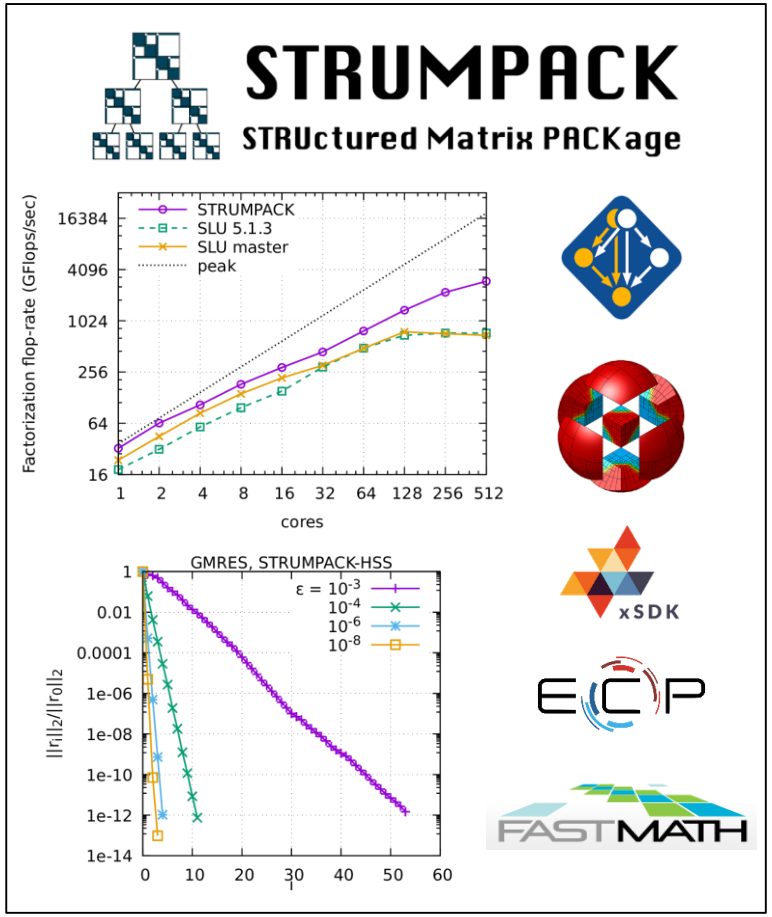
http://www.llnl.gov/CASC/hypre

ECP EXASCALE COMPUTING PROJECT

STRUMPACK

STRUctured Matrix PACKage. Hierarchical solvers for dense rank-structured matrices; fast sparse solver and robust and scalable preconditioners.

- **Dense Matrix Solvers, Hierarchical Approximations**
  - Hierarchical partitioning, low-rank approximations
  - Formats: Hierarchically Semi-Separable (HSS), Hierarchically Off-Diagonal Block Low-Rank (HODLR), Block Low-Rank (BLR)
  - Applicable to integral equations discretized with boundary element methods, structured matrices such as Cauchy or Toeplitz, kernel matrices, covariance matrices, …
  - Algorithms with much lower asymptotic complexity than corresponding ScaLAPACK routines
- **Sparse Direct Solver**
  - Multifrontal algorithm, Fill-reducing orderings: Par-METIS, PT-Scotch, RCM, spectral
  - Good scalability, fully distributed, parallel symbolic phase
- **Sparse Preconditioners**
  - Sparse direct solver with dense hierarchical (low-rank) approximations
  - Scalable and robust, aimed at PDE discretizations, indefinite systems, …
  - Iterative solvers: GMRES, BiCGStab, iterative refinement
- **Software**
  - BSD License, MPI+OpenMP, scalable to 10K+ cores
  - Interfaces from PETSc, MFEM (Trilinos coming), available in Spack
  - Under very active development

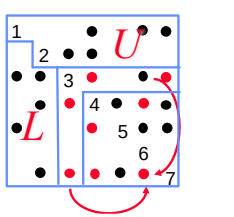


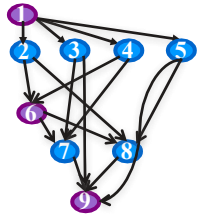
github.com/pghysels/STRUMPACK

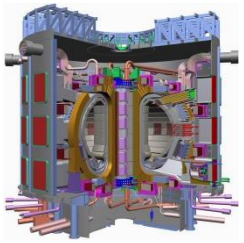
SuperLU

Supernodal Sparse LU Direct Solver. Unique user-friendly interfaces. Flexible software design. Used in a variety of applications. Freely available.

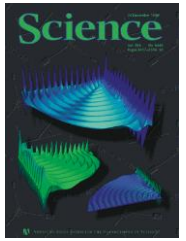
- **Capabilities**
  - Serial (thread-safe), shared-memory (SuperLU_MT, OpenMP or Pthreads), distributed-memory (SuperLU_DIST, hybrid MPI+ OpenM + CUDA).
    - Implemented in C, with Fortran interface
  - Sparse LU decomposition, triangular solution with multiple right-hand sides
  - Incomplete LU (ILU) preconditioner in serial SuperLU
  - Sparsity-preserving ordering:
    - Minimum degree ordering applied to A^TA or A^T+A
    - Nested dissection ordering applied to A^TA or A^T+A [(Par)METIS, (PT)-Scotch]
  - User-controllable pivoting: partial pivoting, threshold pivoting, static pivoting
  - Condition number estimation, iterative refinement.
  - Componentwise error bounds
- **Performance**
  - Factorization strong scales to 24,000 cores (IPDPS'18)
  - Triangular solve strong scales to 4000 cores (CSC'18)
- **Open source software**
  - Used worldwide in a vast range of applications, can be used through PETSc and Trilinos, available on github







ITER tokamak



quantum mechanics

Widely used in commercial software, including AMD (circuit simulation), Boeing (aircraft design), Chevron, ExxonMobile (geology), Cray's LibSci, FEMLAB, HP's MathLib, IMSL, NAG, SciPy, OptimaNumerics, Walt Disney Animation.

FASTMATH xSDK

http://crd-legacy.lbl.gov/~xiaoye/SuperLU

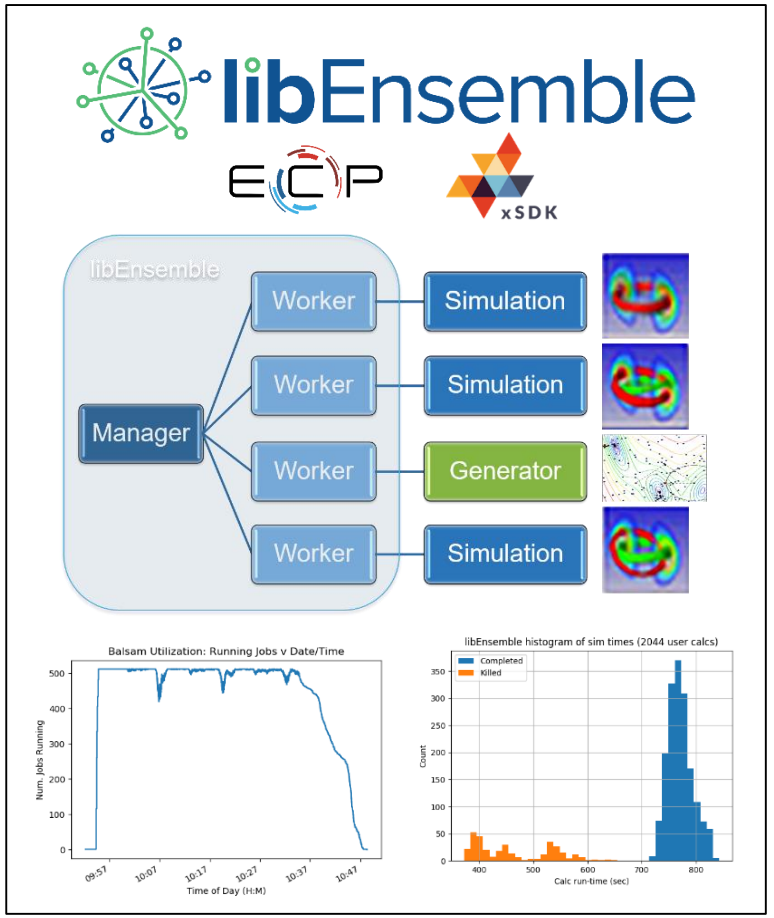
EXASCALE COMPUTING PROJECT

libEnsemble

Argonne NATIONAL LABORATORY

A Python library to run dynamic ensembles. Use massively parallel resources to accelerate the solution of design, decision, and inference problems.

- **libEnsemble aims for:**
  - Extreme scaling
  - Resilience / fault tolerance
  - Monitoring / killing jobs (and recovering resources)
  - Portability and flexibility
- **Manager and Workers**
  - Manager and Workers can run on MPI, Multi-processing or TCP
  - Job controller interface with auto-detection of system and resources
  - Supported on Summit (ORNL), Theta (ALCF), Cori (NERSC)
- **Dynamic ensembles**
  - Workers can be running simulations or generating new simulations
  - Integrate with PETSc solvers (or any other)



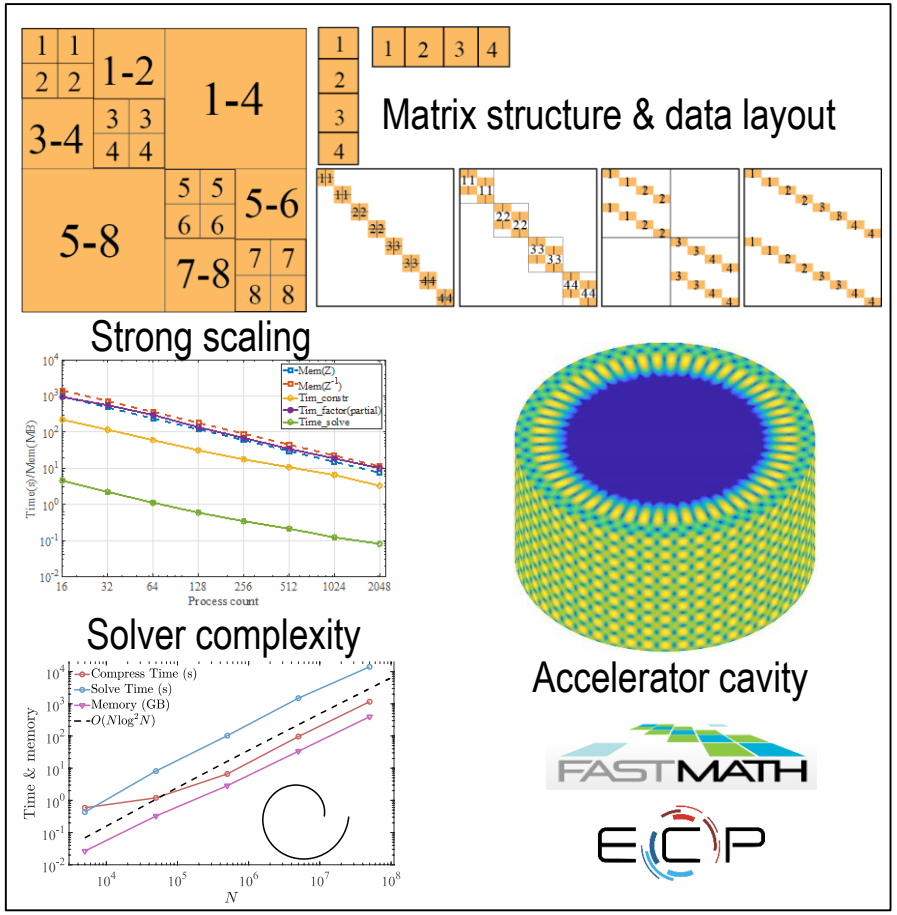
https://libensemble.readthedocs.io

ECP EXASCALE COMPUTING PROJECT

ButterflyPACK

Fast direct solvers. Low-rank and butterfly compression. Distributed-memory parallel. Particularly for highly-oscillatory wave equations.

- **Capabilities**
  - Fast algebraic operations for rank-structured dense and sparse matrices, including matrix compression, multiplication, factorization and solution
  - Support distributed-memory H-matrix, HODLR formats with low-rank and butterflies
  - Particularly targeted at high-frequency electromagnetic, acoustic and elastic applications
- **Conceptual interfaces**
  - User input: a function to compute arbitrary matrix entries or to multiply the matrix with arbitrary vectors
  - Both Fortran2008 and C++ interface available
  - Highly interoperable with STRUMPACK
- **Open source software**
  - Software dependence: BLAS, LAPACK, SCALAPACK, ARPACK
  - Newly released on github with tutorial examples available: https://github.com/liuyangzhuan/ButterflyPACK/tree/master/EXAMPLE

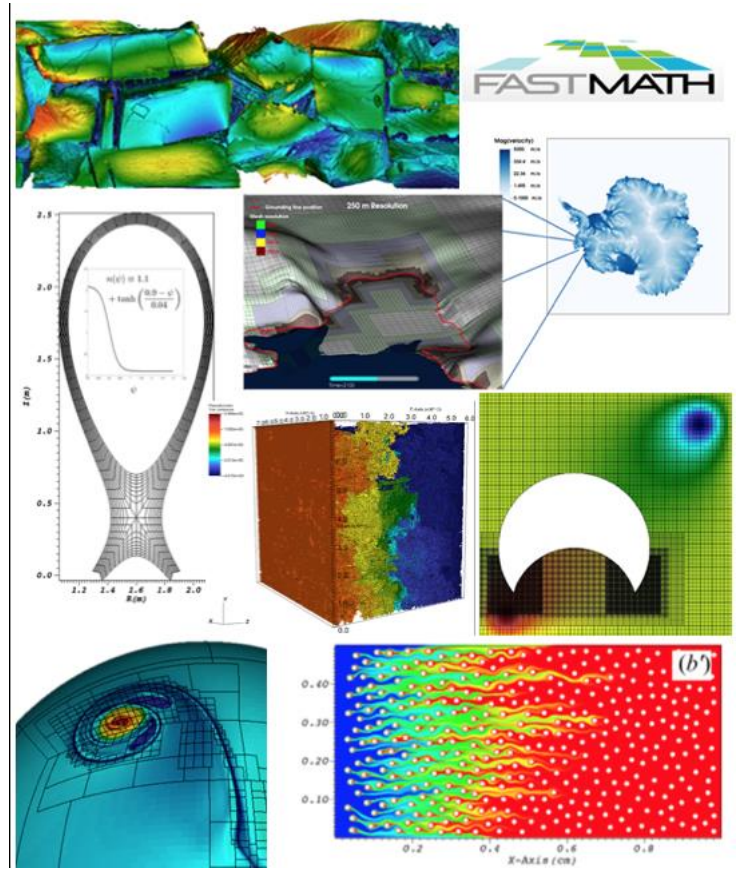


https://github.com/liuyangzhuan/ButterflyPACK

Chombo

Scalable adaptive mesh refinement framework. Enables implementing scalable AMR applications with support for complex geometries.

- **Adaptive Mesh Refinement (AMR)**
  - Block structured AMR dynamically focuses computational effort where needed to improve solution accuracy
  - Designed as a developers' toolbox for implementing scalable AMR applications
  - Implemented in C++/Fortran
  - Solvers for hyperbolic, parabolic, and elliptic systems of PDEs
- **Complex Geometries**
  - Embedded-boundary (EB) methods use a cut-cell approach to embed complex geometries in a regular Cartesian mesh
  - EB mesh generation is extremely efficient
  - Structured EB meshes make high performance easier to attain
- **Higher-order finite-volume**
  - Higher (4th)-order schemes reduce memory footprint & improve arithmetic intensity
  - Good fit for emerging architectures
  - Both EB and mapped-multiblock approaches to complex geometry



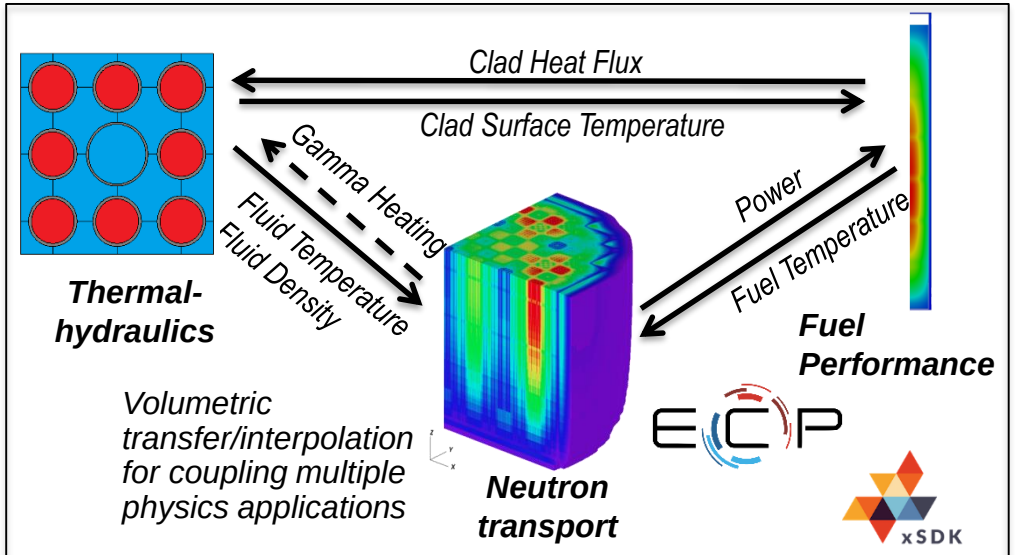
http://Chombo.lbl.gov

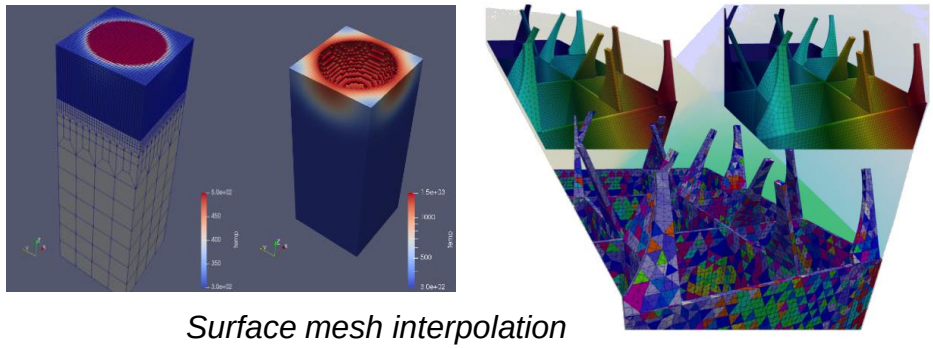
DataTransferKit

OAK RIDGE National Laboratory

Open source library for parallel solution transfer.
Support for grid-based and mesh-free applications.

- **Overview**
  - Transfers application solutions between grids with differing layouts on parallel accelerated architectures
  - Coupled applications frequently have different grids with different parallel distributions; DTK is able to transfer solution values between these grids efficiently and accurately
  - Used for a variety of applications including conjugate heat transfer, fluid structure interaction, computational mechanics, and reactor analysis
- **Capabilities**
  - Support for DOE leadership class machines through MPI+Kokkos programming model
  - Algorithms demonstrated scalable to billions of degrees of freedom
  - General geometric search algorithms
    - Comparable serial performance to Boost r-Tree and NanoFlann
    - Also thread scalable on many core CPU and GPUs and distributed via MPI
  - Grid interpolation operators and mesh-free transfer operators



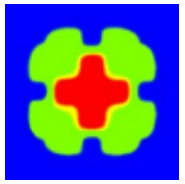


Surface mesh interpolation

https://github.com/ORNL-CEES/DataTransferKit

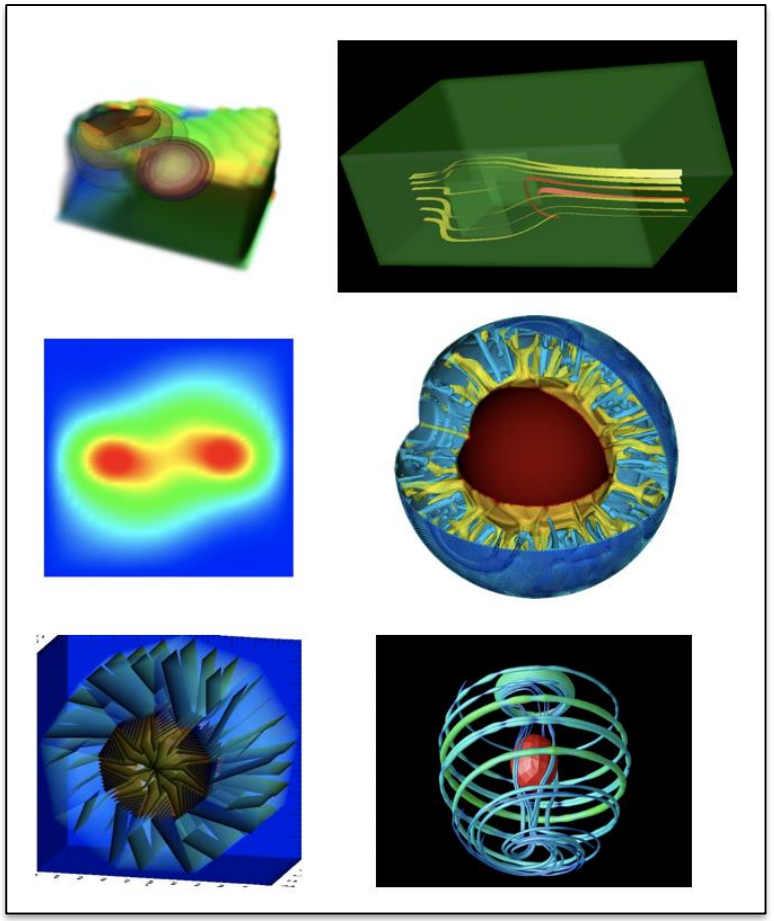
ECP EXASCALE COMPUTING PROJECT

deal.II



deal.II — an open source finite element library. Modern interface to the complex data structures and algorithms required for solving partial differential equations computationally using state-of-the-art programming techniques.

- **Meshes and elements:**
  - Supports h- and p-adaptive meshes in 1d, 2d, and 3d
  - Easy ways to adapt meshes: Standard refinement indicators already built in
  - Many standard finite element types (continuous, discontinuous, mixed, Raviart-Thomas, Nedelec, ABF, BDM,...)
  - Full support for coupled, multi-component, multi-physics problems
- **Linear algebra:**
  - Has its own sub-library for dense and sparse linear algebra
  - Interfaces to PETSc, Trilinos, UMFPACK, ScaLAPACK, ARPACK
- **Pre- and postprocessing:**
  - Can read most mesh formats
  - Can write almost any visualization file format
- **Parallelization:**
  - Uses threads and tasks on shared-memory machines
  - Uses up to 100,000s of MPI processes for distributed-memory machines
  - Can use CUDA
- **Open-source software:**
  - Used for a wide range of applications, including heart muscle fibers, microfluidics, oil reservoir flow, fuel cells, aerodynamics, quantum mechanics, neutron transport, numerical methods research, fracture mechanics, damage models, sedimentation, biomechanics, root growth of plants, solidification of alloys, glacier mechanics, and many others.
  - Freely available on GitHub



https://www.dealii.org

MAGMA

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Linear algebra solvers and spectral decompositions for hardware accelerators.
Portable dense direct and sparse iterative solvers for GPUs and coprocessors.

- **Dense Linear Algebra Solvers**
  - Linear systems of equations
  - Linear least squares
  - Singular value decomposition
- **Matrix spectrum methods**
  - Symmetric and non-symmetric eigenvalues
  - Generalized eigenvalue problems
  - Singular Value Decomposition
- **Sparse Solvers & Tensor Computations**

MAGMA SPARSE

ROUTINES	BiCG, BiCGSTAB, Block-Asynchronous Jacobi, CG, CGS, GMRES, IDR, Iterative refinement, LOBPCG, LSQR, QMR, TFQMR
PRECONDITIONERS	ILU / IC, Jacobi, ParILU, ParILUT, Block Jacobi, ISAI
KERNELS	SpMV, SpMM
DATA FORMATS	CSR, ELL, SELL-P, CSR5, HYB

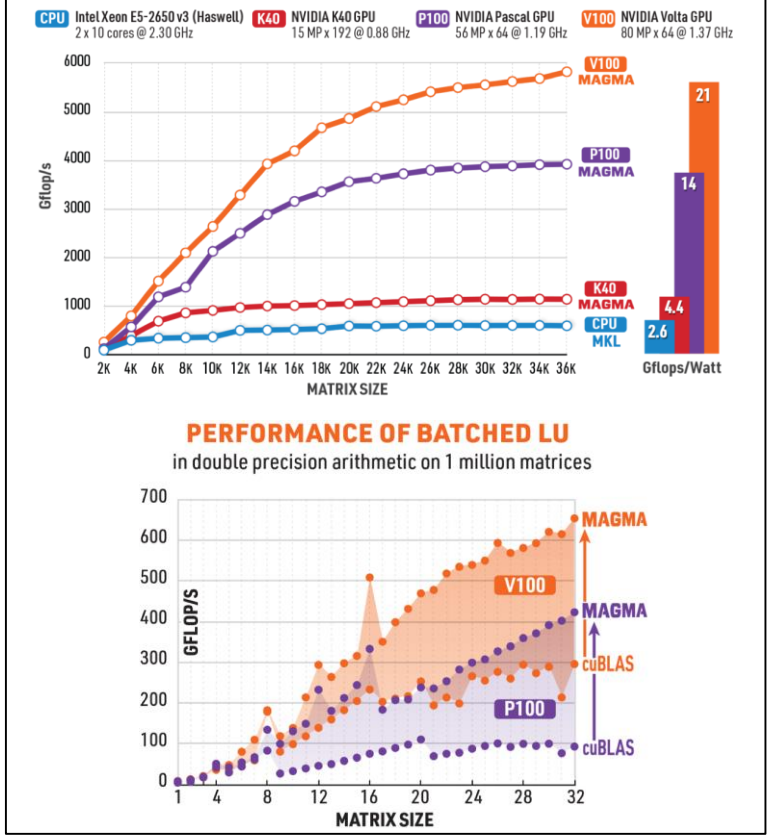
FEATURES AND SUPPORT

- **MAGMA 2.3** FOR CUDA
- **clMAGMA 1.4** FOR OpenCL
- **MAGMA MIC 1.4** FOR Intel Xeon Phi

CUDA	OpenCL	Intel Xeon Phi	
●	●	●	Linear system solvers
●	●	●	Eigenvalue problem solvers
●	●		Auxiliary BLAS
●			Batched LA
●		●	Sparse LA
●	●	●	CPU/GPU Interface
●	●	●	Multiple precision support
●			Non-GPU-resident factorizations
●	●	●	Multicore and multi-GPU support
● NEW			MAGMA Analytics/DNN
●	●	●	LAPACK testing
●	●	●	Linux
●	●		Windows
●	●		Mac OS

PERFORMANCE & ENERGY EFFICIENCY

MAGMA LU factorization in double precision arithmetic



PERFORMANCE OF BATCHED LU

in double precision arithmetic on 1 million matrices

http://icl.utk.edu/magma

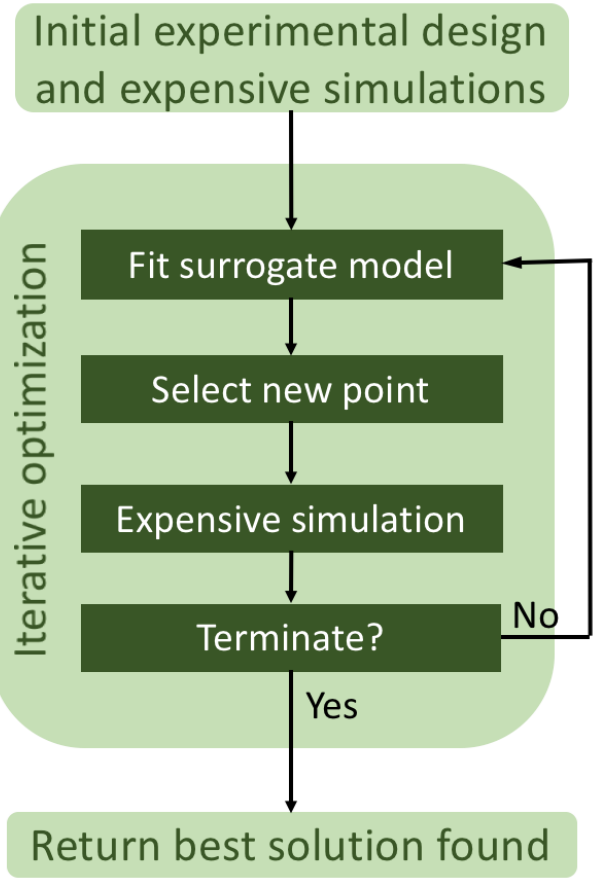
EXASCALE COMPUTING PROJECT

MATSuMoTo

MATLAB Surrogate Model Toolbox

Efficient optimization of computationally-expensive black-box problems. For integer, mixed-integer, and continuous variables. Your choice of surrogate model, sampling method, and initial design strategy. Easy to use. Freely available.

- **Capabilities**
  - Efficient solution of parameter optimization problems that involve time-consuming black-box HPC simulations during the objective function evaluation
  - Surrogate models approximate the expensive function and aid in iterative selection of sample points
  - Adaptive sampling for continuous, integer, and mixed-integer problems *without* relaxation of integer constraints
- **Available User options**
  - **Surrogate model choices:** radial basis functions, polynomial regression, multivariate adaptive regression splines, surrogate model ensembles
  - **Iterative sample point selection:** local perturbations, global candidate points, minimization over the surrogate model
  - **Initial experimental design:** Latin hypercube, symmetric Latin hypercube, design space corners



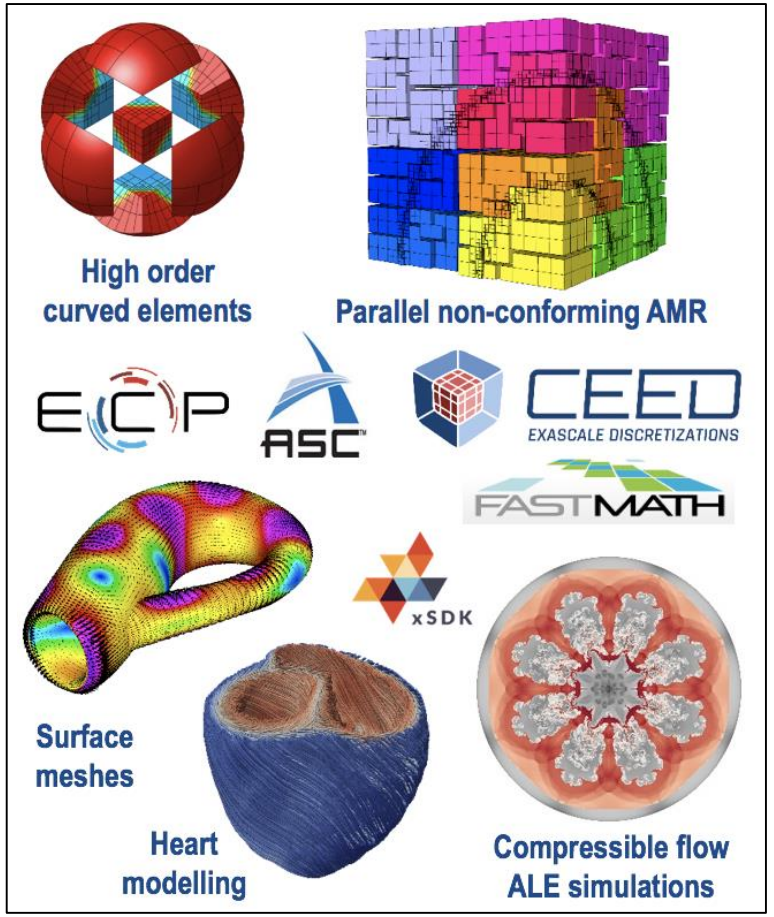
https://optimization.lbl.gov/downloads

MFEM

Lawrence Livermore National Laboratory

Free, lightweight, scalable C++ library for finite element methods. Supports arbitrary high order discretizations and meshes for wide variety of applications.

- **Flexible discretizations on unstructured grids**
  - Triangular, quadrilateral, tetrahedral and hexahedral meshes.
  - Local conforming and non-conforming refinement.
  - Bilinear/linear forms for variety of methods: Galerkin, DG, DPG, …
- **High-order and scalable**
  - Arbitrary-order H1, H(curl), H(div)- and L2 elements. Arbitrary order curvilinear meshes.
  - MPI scalable to millions of cores and includes initial GPU implementation. Enables application development on wide variety of platforms: from laptops to exascale machines.
- **Built-in solvers and visualization**
  - Integrated with: HYPRE, SUNDIALS, PETSc, SUPERLU, …
  - Accurate and flexible visualization with VisIt and GLVis
- **Open source software**
  - LGPL-2.1 with thousands of downloads/year worldwide.
  - Available on GitHub, also via OpenHPC, Spack. Part of ECP's CEED co-design center.



http://mfem.org

ECP EXASCALE COMPUTING PROJECT

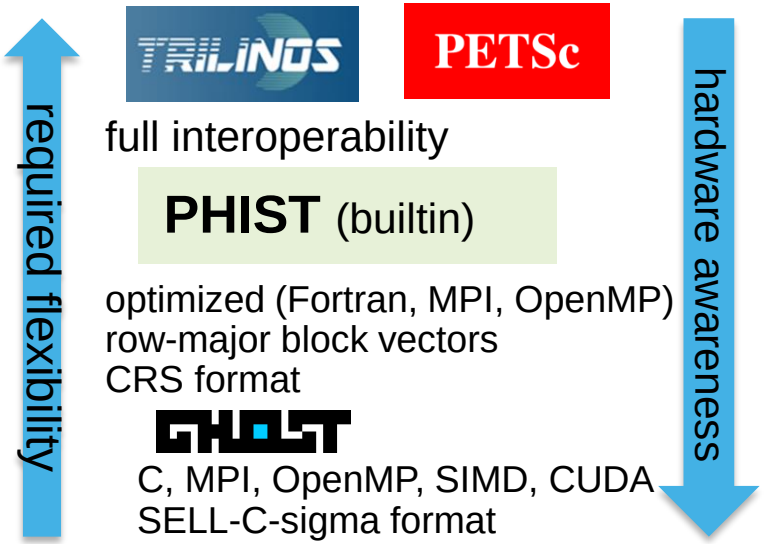
PHIST

Pipelined, Hybrid-parallel Iterative Solver Toolkit

Hybrid-parallel Iterative Sparse Eigenvalue and linear solvers
Integration with different linear algebra backends and preconditioners

- **Sparse Eigenvalue Solver: Block Jacobi-Davidson QR**
  - Hermitian or non-Hermitian matrices
  - Generalized problems $\boldsymbol{A}x = \lambda \boldsymbol{B}x$ (for Hermitian pos. def. matrix $\mathbf{B}$)
  - Blocked iterative linear solvers like GMRES, BiCGStab and CGMN
  - Can be accelerated by preconditioning
  - Matrix-free interface
  - Supported data types: D, Z, S, C
- **Algorithmic Building Blocks**
  - block orthogonalization
  - Eigenvalue counting (kernel polynomial method/KPM)
  - Fused basic operations for better performance
- **Various interfaces**
  - C, C++, Fortran 2003, Python

Can choose from several backends at compile time



Funded by the DFG project ESSEX
SPPEXA

https://bitbucket.org/essex/phist

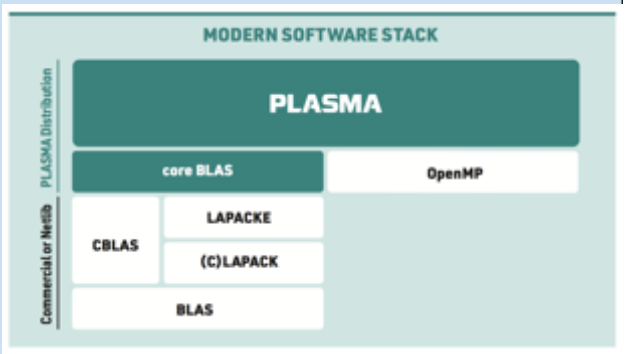
ECP EXASCALE COMPUTING PROJECT

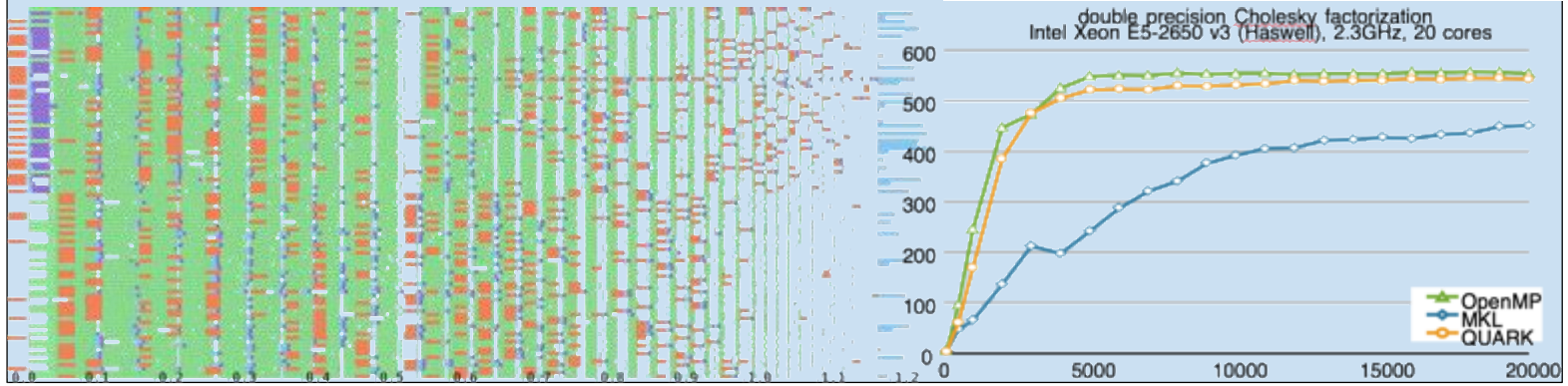
PLASMA

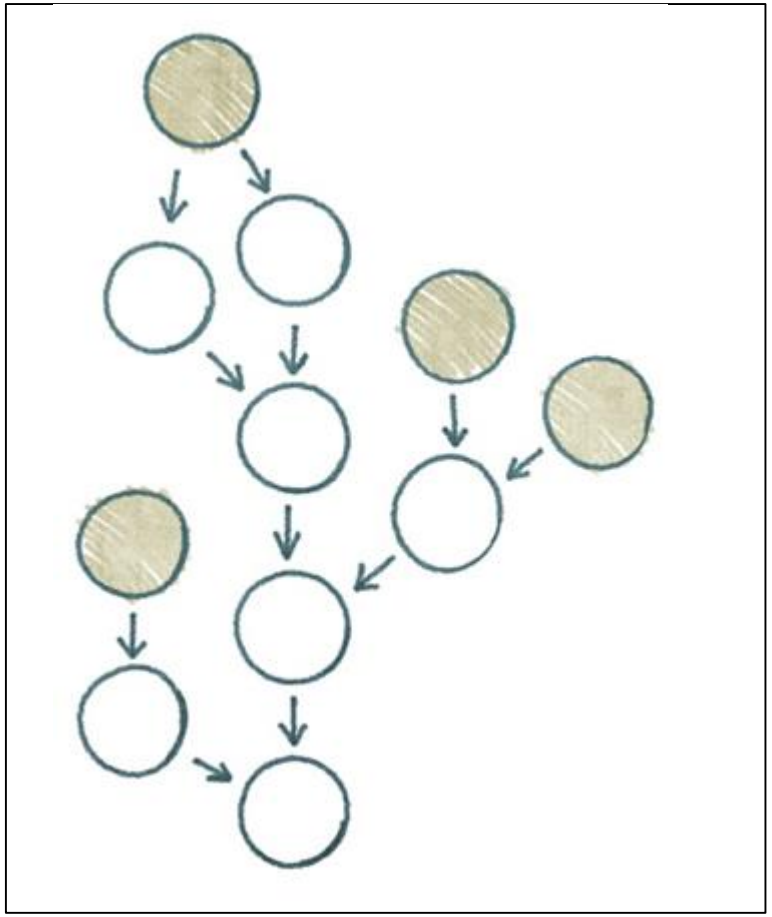
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Linear algebra solvers and spectral decompositions for multicore processors.
Portable and scalable dense solvers for large core counts.

- **Dense Linear Algebra Solvers**
  - Linear systems of equations
  - Linear least squares
  - Positive/Hermitian definitive solvers
- **Matrix spectrum methods**
  - Symmetric and non-symmetric eigenvalues
  - Generalized eigenvalue problems
  - Singular Value Decomposition
- **Data conversion and thread control**







http://icl.utk.edu/plasma

ECP EXASCALE COMPUTING PROJECT

PUMi: Parallel Unstructured Mesh Infrastructure

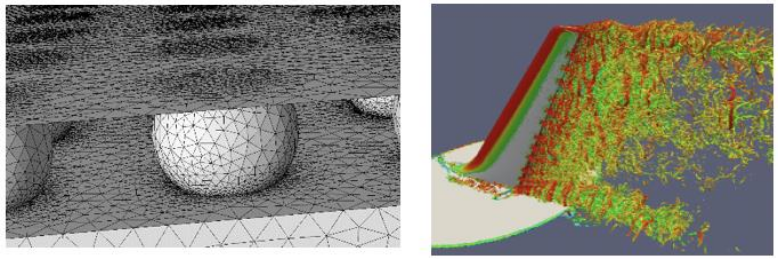
Parallel management and adaptation of unstructured meshes.
Interoperable components to support the development of unstructured mesh simulation workflows

- **Core functionality**
  - Distributed, conformant mesh with entity migration, remote read only copies, fields and their operations
  - Link to the geometry and attributes
  - Mesh adaptation (straight and curved), mesh motion
  - Multi-criteria partition improvement
  - Distributed mesh support for Particle In Cell methods
- **Designed for integration into existing codes**
  - Conformant with XSDK
  - Permissive license enables integration with open and closed-source codes
- **In-memory integrations developed**
  - MFEM: High order FE framework
  - PHASTA: FE for turbulent flows
  - FUN3D: FV CFD
  - Proteus: Multiphase FE
  - ACE3P: High order FE for EM
  - M3D-C1: FE based MHD
  - Nektar++: High order FE for flow
  - Albany/Trilinos: Multi-physics FE

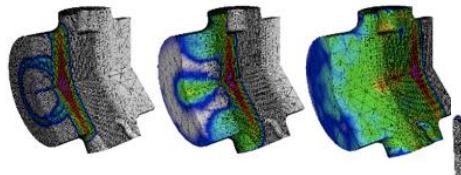
PUMi

xSDK SCOREC ECP

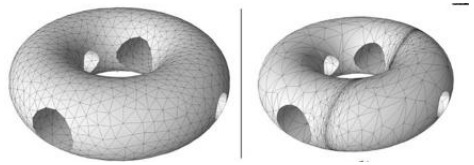
Rensselaer FASTMATH



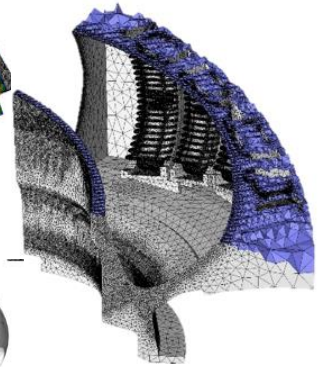
Applications with billions of elements: flip-chip (L), flow control (R)



Mesh adaptation for evolving features



Anisotropic adaptation for curved meshes



RF antenna and plasma surface in vessel

Source Code: github.com/SCOREC/core
Paper: www.scorec.rpi.edu/REPORTS/2014-9.pdf

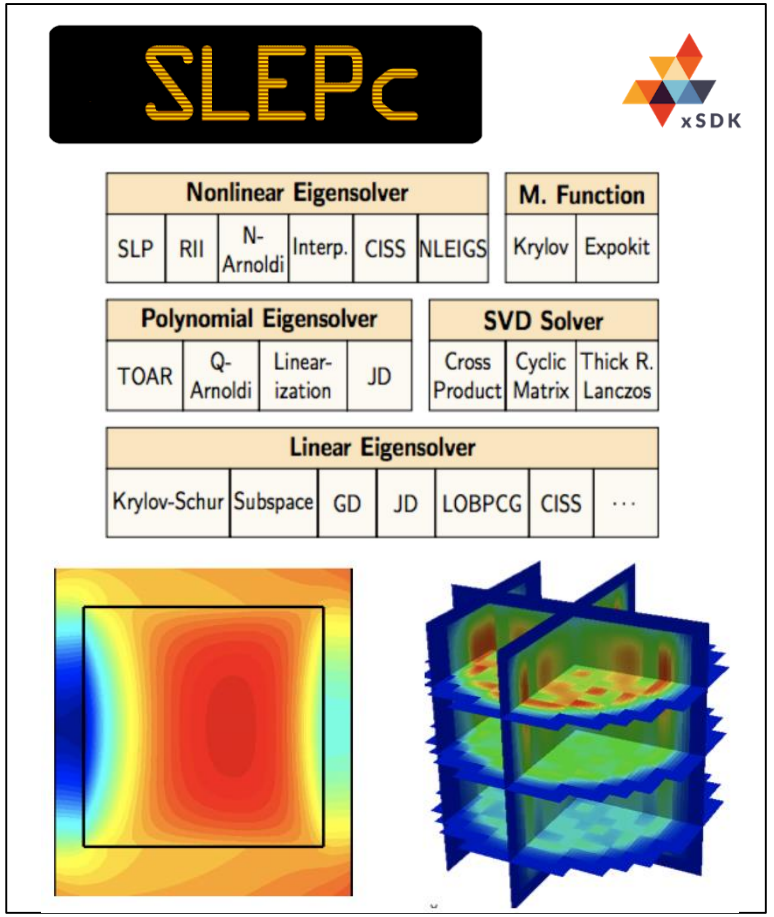
ECP EXASCALE COMPUTING PROJECT

SLEPc

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Scalable Library for Eigenvalue Problem Computations. Parallel solvers for linear and nonlinear eigenproblems. Also functionality for matrix functions.

- **Linear eigenvalue problems and SVD**
  - Standard and generalized eigenproblem, $Ax=\lambda x$, $Ax=\lambda Bx$; singular values $Au=\sigma v$
  - Easy selection of target eigenvalues, shift-and-invert available for interior ones
  - Many solvers: Krylov, Davidson, LOBPCG, contour integral, …
- **Nonlinear eigenvalue problems**
  - Polynomial eigenproblem $P(\lambda)x=0$, for quadratic or higher-degree polynomials
  - Solvers: Krylov with compact basis representation; Jacobi-Davidson
  - General nonlinear eigenproblem $T(\lambda)x=0$, for any nonlinear function incl. rational
- **Matrix functions**
  - Parallel Krylov solver to evaluate $y=f(A)v$
  - Support for matrix exponential, square root, etc. and combinations thereof
- **Extension of PETSc**
  - Runtime customization, portability and performance, C/C++/Fortran/python
  - Can use any PETSc linear solvers and preconditioners



http://slepc.upv.es

ECP EXASCALE COMPUTING PROJECT

Trilinos

Sandia National Laboratories

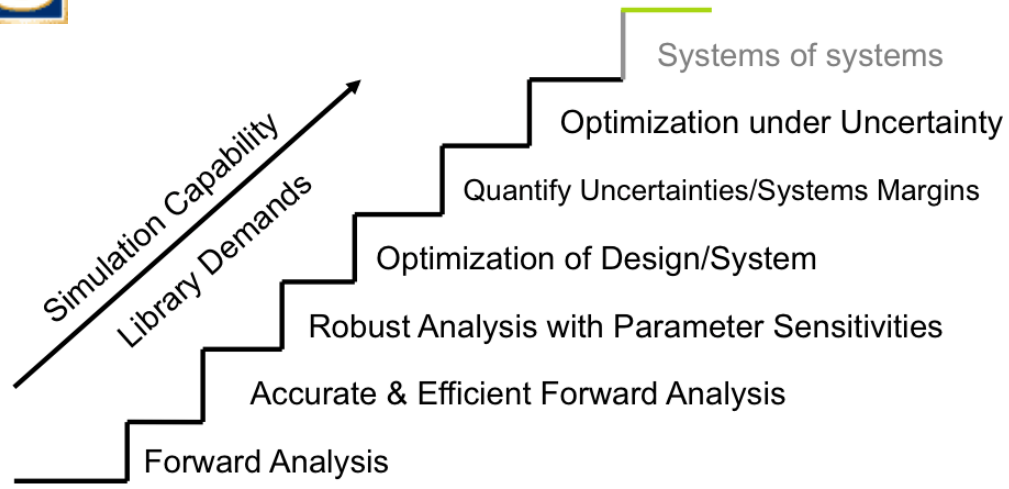
Optimal kernels to optimal solutions. Over 60 packages. Laptops to leadership systems. Next-gen systems, multiscale/multiphysics, large-scale graph analysis.

- **Optimal kernels to optimal solutions**
  - Geometry, meshing
  - Discretization, load balancing
  - Scalable linear, nonlinear, eigen, transient, optimization, UQ solvers
  - Scalable I/O, GPU, manycore
  - Performance Portability Across Multiple Platforms provided by **Kokkos**
- **60+ packages**
  - Other distributions: Cray LIBSCI, Github repo
  - Thousands of users, worldwide distribution
  - Laptops to leadership systems



R&D 100

Transforming Computational Analysis To Support High Consequence Decisions



Each stage requires *greater performance* and *error control* of prior stages:
Always will need: more accurate and scalable methods.
more sophisticated tools.

https://trilinos.org

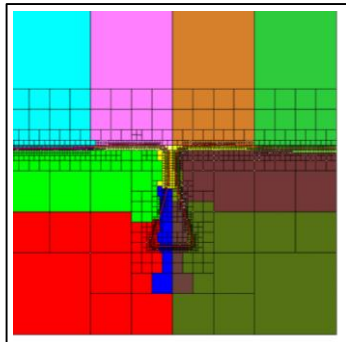
xSDK FASTMath TRILINOS

ECP EXASCALE COMPUTING PROJECT

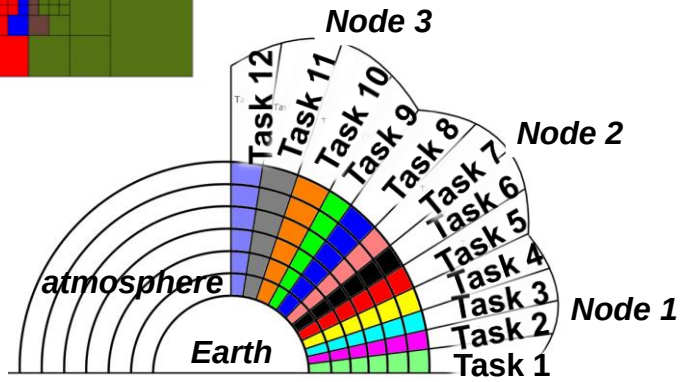
Zoltan/Zoltan2

Parallel partitioning, load balancing, task placement, graph coloring, matrix ordering, unstructured communication utilities, distributed directories

- **Partitioning & load-balancing support many applications**
  - Fast geometric methods maintain spatial locality of data (e.g., for adaptive finite element methods, particle methods, crash/contact simulations)
  - Graph and hypergraph methods explicitly account for communication costs (e.g., for electrical circuits, finite element meshes, social networks)
  - Single interface to popular partitioning TPLs: XtraPuLP (SNL, RPI); ParMA (RPI); PT-Scotch (U Bordeaux); ParMETIS (U Minnesota)
- **Architecture-aware MPI task placement reduces application communication time**
  - Places interdependent MPI tasks on "nearby" nodes in computing architecture
  - Reduces communication time and network congestion
- **Graph algorithms for coloring, ordering, connectivity**
- **Use as a stand-alone library or as a Trilinos component**



Zoltan's fast, geometric partitioner redistributes work to maintain load balance in a surface deposition simulation with adaptive meshing



Zoltan2's task placement reduced communication time in E3SM HOMME (atmospheric modeling) by up to 31% on 16K cores of IBM BG/Q

Sandia National Laboratories

FASTMATH

ASC

https://www.cs.sandia.gov/Zoltan

ECP EXASCALE COMPUTING PROJECT

HandsOnLessons

- Hand-coded heat equation intro
- Structured meshing & discretization
- Time integration & nonlinear solvers
- Krylov solvers & preconditioners
- Numerical optimization – boundary control

ATPESC 2019 Hands On Lessons

Github pages site:

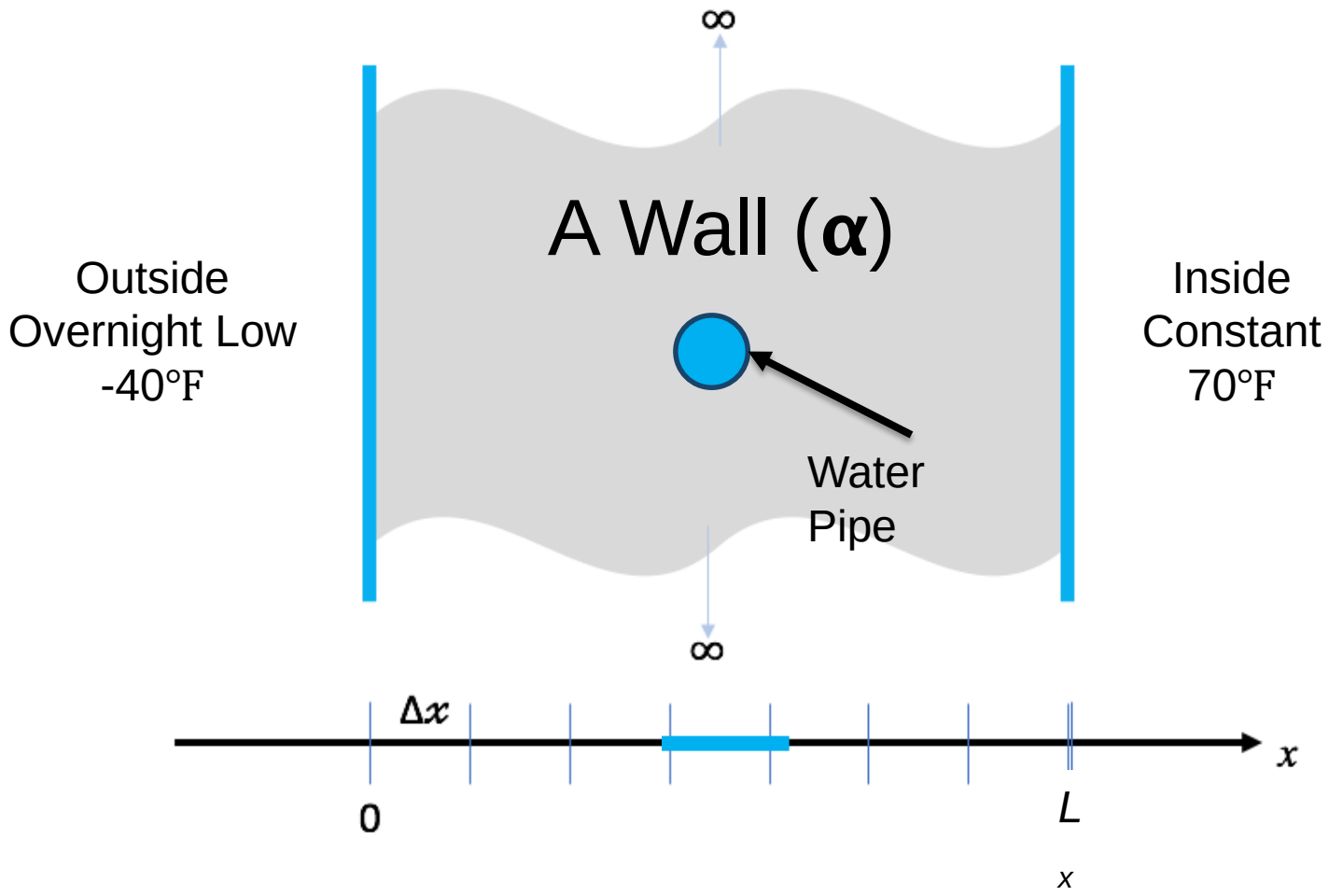
https://xsdk-project.github.io/MathPackagesTraining/lessons/

Hello World
(for numerical packages)

Mark C Miller, LLNL

IDEAS-ECP/ATPESC SQE Support and Training Coordinator

A Science Problem of Interest:
Will My Water Pipes Freeze?



$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

The One-Dimensional Heat Equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

- $u(x,t)$ is temperature in Kelvin
- x is distance in meters
- t is time in seconds
- **α** is thermal diffusivity of the material (m^2/s)

Given boundary and initial conditions

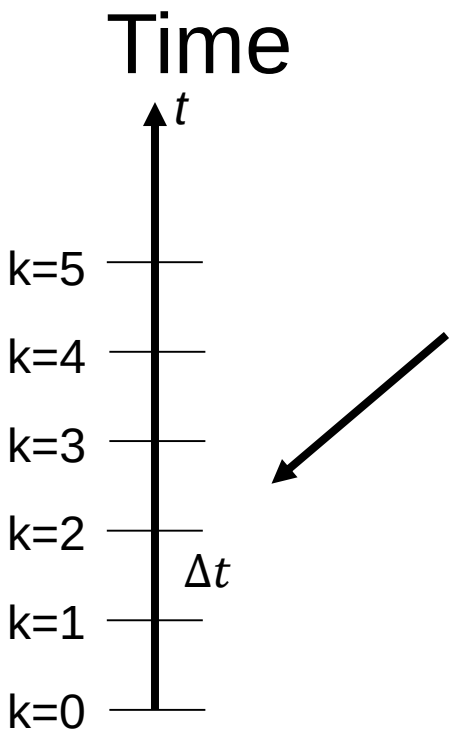
- Left end-point: $u(0,t) = U_0$
- Right end-point: $u(L_x,t)=U_L$
- Initial temperature profile: $u(x,0)=U(x)$

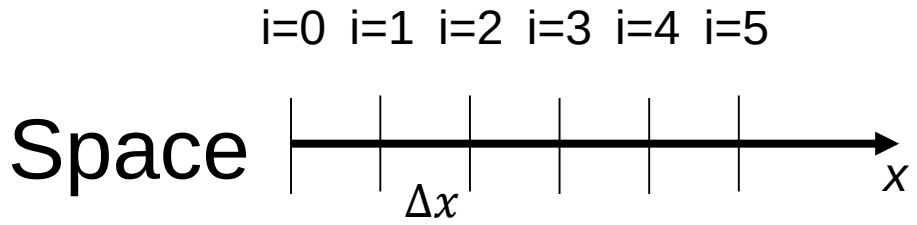
We seek a numerical software solution for $u(x,t)$ (all space & time)

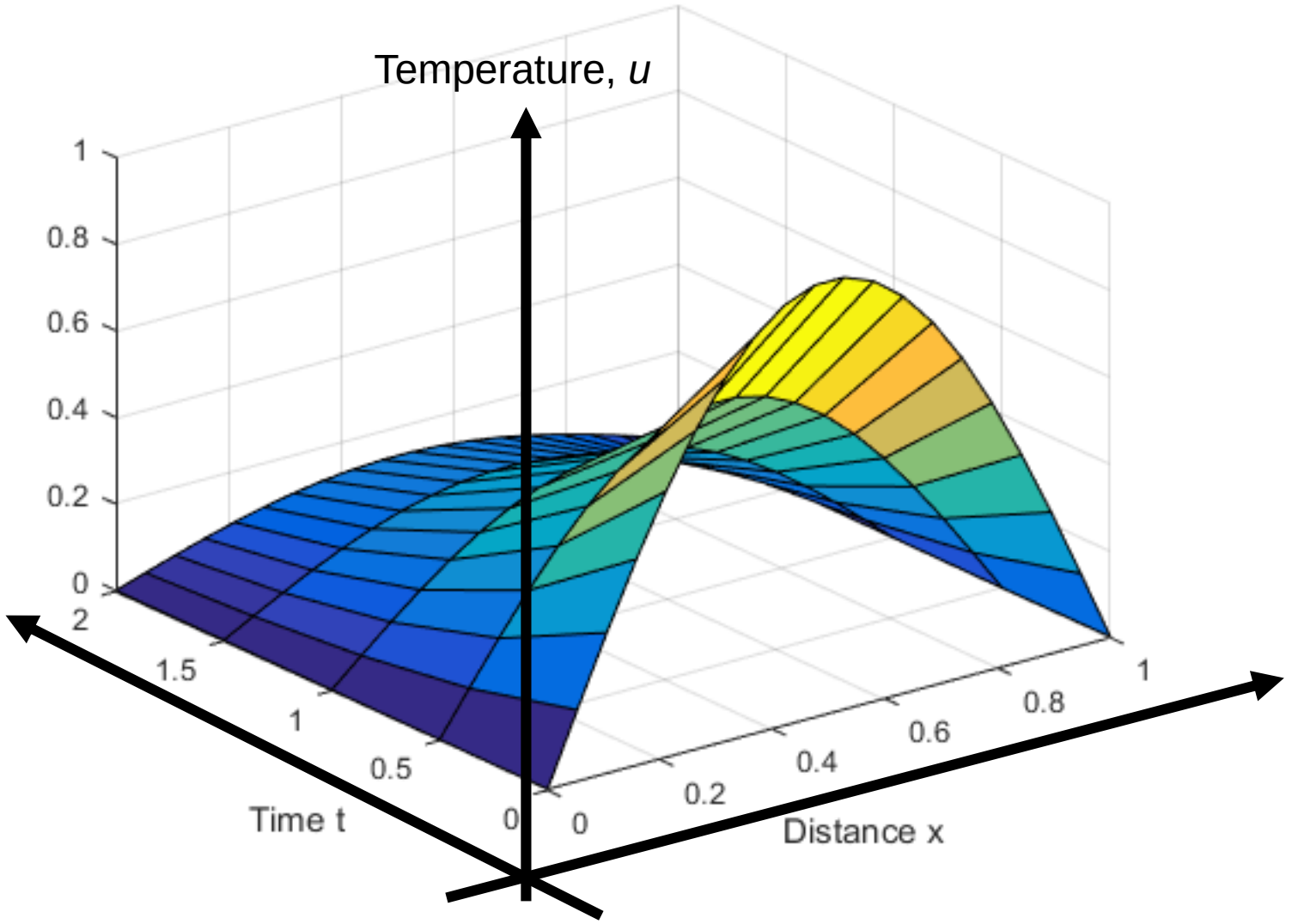
Discretize: Continuous → Discrete

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \alpha \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x}$$







A numerical, iterative solution algorithm

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \alpha \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x}$$

$$u_i^{k+1} = ru_{i+1}^k + (1-2r)u_i^k + ru_{i-1}^k \qquad r = \alpha \frac{\Delta t}{(\Delta x)^2}$$

- *k* is indexing time, *t*, and *i* is indexing distance, *x*
- Known as “FTCS” algorithm
- Is an *explicit* method
- Known to be **unstable** for $r > \frac{1}{2}$

Exercise #1 (3 mins)

Open ftcs.C w/editor and write the body of this function

$$u_i^{k+1} = ru_{i+1}^k + (1-2r)u_i^k + ru_{i-1}^k$$

$$r = \alpha \frac{\Delta t}{(\Delta x)^2}$$

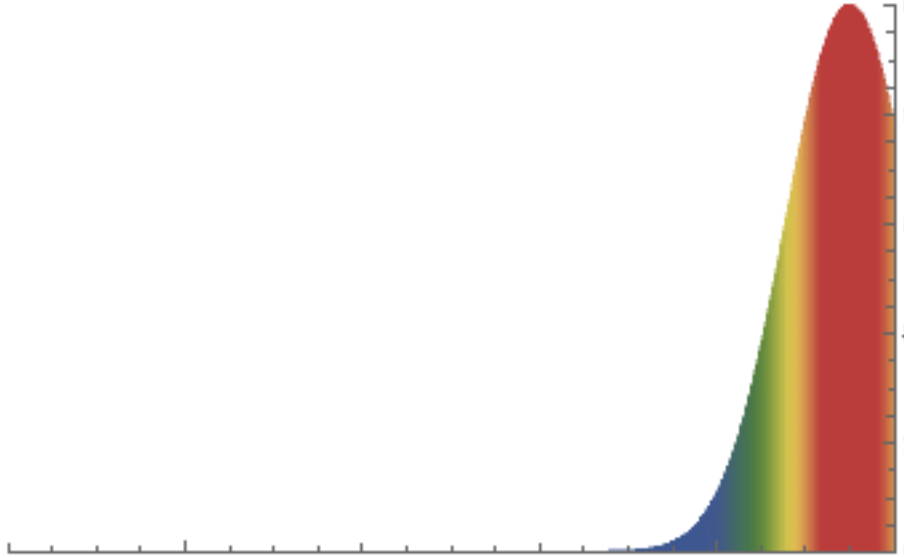
```
bool                                // true if valid, false if not
update_solution_ftcs(
    int n,                          // number of values
    Double *uk1,                    // new values: u(i) i=0…n-1 @ t=k+1
    Double const *uk0,              // last values: u(i) i=0…n-1 @ t=k
    Double alpha,                   // thermal diffusivity
    Double dx, Double dt,           // spacing in space, x, and time, t.
    Double bc0, Double bc1)         // boundary conditions @ x=0 & x=L
{




}
```

Exercise #2 (1 min)
Build and test the application

```
% make
c++ -c   heat.C -o heat.o
c++ -c   utils.C -o utils.o
c++ -c   args.C -o args.o
c++ -c   exact.C -o exact.o
c++ -c   ftcs.C -o ftcs.o
c++ -c   upwind15.C -o upwind15.o
c++ -c   crankn.C -o crankn.o
c++ -o heat heat.o utils.o args.o exact.o ftcs.o upwind15.o crankn.o -lm
```



- How might we test it?
  - We know steady state solution for bc0=A and bc1=B is line from A to B

Exercise #3 (2 mins):
Run the application to model a problem of interest

- Outside temp has been same as inside temp @ 70 °F for a long time
- Night/Storm will last 15.5 hours @ -40 °F
- Walls are 0.25 meters thick wood, pipe is 0.1 meters diameter

Material	Thermal Diffusivity, α, (m^2/s)
Wood	8.2×10^{-8}
Adobe Brick	2.7×10^{-7}
Common (“red”) brick	5.2×10^{-7}

Exercise #4 (1 min)
Analyze the results

Criterion: Will conclude pipe freezes if…
…center point drops below freezing before storm passes

```
make plot PTOOL=[visit|gnuplot|pyplot] RUNAME=<run-name>
```

What if our problem was to find the optimum wall width?

Simplifications hide challenges of math package software engineering

- **Challenges in numerical algorithms**
  - Discretizations: Dimensions, geometries, material interfaces, etc
  - Time Integrators: Adaptive, faster convergence, efficiencies, etc.
  - Solvers: Implicit, explicit, iterative, direct, preconditioners, etc.
  - Optimization: Outer loops, nonintrusive, reduced-order models, etc.
  - Validation & verification: Vetted, trusted results, community accepted
- **Challenges in software development**
  - Time and space performance
  - Scalability & performance portability
  - Encapsulation, interfaces & interoperability
  - Documentation, ease of installation, ease of use
  - Sustainable open source, supported with regular updates, bug tracking/fixing
  - Support for user-defined customization and extensibility

Sign up for 2-on-1 discussions with numerical software developers

Via Google form: See link in email:

- **Your email address**
- Select 1st, 2nd, and 3rd priorities for short 2-on-1 meetings with expert developers
- Brief description of interests

- Complete by 4:30 pm CDT

Meeting opportunities include:

- Today (evening)
- Other days/times, opportunities for communication with developers who are not attending today