Overview of Numerical Algorithms and Software for Extreme-Scale Science

Presented to
ATPESC 2020 Participants

Lois Curfman McInnes
Argonne National Laboratory

Mark C. Miller
Lawrence Livermore National Laboratory

Date 08/04/2020

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

Outline

- Logistics for the day
- Intro to numerical algorithms and software for extreme-scale science
- Gallery of highlights: HPC numerical software packages
- Hands-on example: “Hello world” for numerical packages

Your home bases for the day: ATPESC Track 5
Numerical Algorithms and Software for Extreme-Scale Science

- Main ATPESC Agenda
  - https://extremecomputingtraining.anl.gov/agenda-2020/#Track-5
  - slides (pdf) and presenter bios
- Math Packages Training Site
  - session abstracts, links to parallel breakout rooms, hands-on lessons, more
  - https://xsdk-project.github.io/MathPackagesTraining2020/agenda

Agenda

https://extremecomputingtraining.anl.gov/agenda-2020/#Track-5

Time	Session	Presenter(s)
9:30	Introduction to Numerical Libraries and Virtual Logistics	Lois Curfman McInnes, ANL and Mark Miller, LLNL
10:30	*Session 1*	
	• **ROOM FRONTIER:** Structured Discretization (with AMReX)	Ann Almgren, LBL and Don Willcox, LBL
	• **ROOM AURORA**: Unstructured Discretization (with MFEM/PUMI)	Mark Shephard, RPI and Aaron Fisher, LLNL
	• **ROOM PERLMUTTER:** Iterative Solvers+Preconditioners (with Trillinos/MueLu)	Jonathan Hu, SNL and Christian Glusa, SNL
	• **ROOM EL CAPITAN:** Direct Solvers (with SuperLU/Strumpack)	Sherry Li, LBL and Pieter Ghysels, LBL
11:30	*Break*	
11:45	*Session 2*	
	• **ROOM FRONTIER:** Structured Discretization (with AMRex)	Ann Almgren, LBL and Don Willcox, LBL
	• **ROOM AURORA:** Unstructured Discretization (with MFEM/PUMI)	Mark Shephard, RPI and Aaron Fisher, LLNL
	• **ROOM PERLMUTTER:** Iterative Solvers+Algebraic Multigrid (with HYPRE)	Ulrike Yang, LLNL
	• **ROOM EL CAPITAN:** Direct Solvers (with SuperLU/Strumpack)	Sherry Li, LBL and Pieter Ghysels, LBL
12:45 p.m.	*Lunch*	
1:45	*Panel Discussion*: Contributing to the Numerical Package Community	Panel Moderator: Mark Miller, LLNL Panelists: Jonathan Hu, SNL; Richard Mills, ANL; Sherry Li, LBL; Cameron Smith, RPI; and Ulrike Yang, LLNL
2:35	*Session 3*	
	• **ROOM FRONTIER:** Nonlinear Solvers (with PETSc)	Richard Mills, ANL
	• **ROOM AURORA:** Optimization (with TAO)	Alp Dener, ANL
	• **ROOM PERLMUTTER:** Time Integration (with SUNDIALS)	Dan Reynolds, SMU
	• **ROOM EL CAPITAN:** Iterative Solvers+Preconditioners (with Trillinos/MueLu)	Jonathan HU, SNL and Christian Glusa, SNL
3:25	*Break*	
3:40	*Session 4*	
	• **ROOM FRONTIER:** Nonlinear Solvers (with PETSc)	Richard Mills, ANL
	• **ROOM AURORA:** Optimization (with TAO)	Alp Dener, ANL
	• **ROOM PERLMUTTER:** Time Integration (with SUNDIALS)	Dan Reynolds, SMU
	• **ROOM EL CAPITAN:** Direct Solvers (with SuperLU/Strumpack)	Sherry Li, LBL and Pieter Ghysels, LBL
4:30	Working with Numerical Packages in Practice	Ann Almgren, LBL
5:00	Adjourn	
5:15	Optional Activity: SME speed-dating in pairs	Mark Miller, LLNL
6:35	Optional Activity Concludes	

EXASCALE COMPUTING PROJECT

Agenda Overview

Start CDT	Activity	Virtual Room
9:30	Introduction	Main room
10:30	Parallel Session #1	Four parallel rooms
11:30	Break	
11:45	Parallel Session #2	Four parallel rooms
12:45	Lunch	
1:45	Panel	Main room
2:35	Parallel Session #3	Four parallel rooms
3:25	Break	
3:40	Parallel Session #4	Four parallel rooms
4:30	Wrap-up	Main room
5:00	Break	
5:15	Subject Matter Expert (SME) Speed Dating (optional)	Individual SME rooms

Mix-n-Match topics to your interests

See Synopses from Agenda

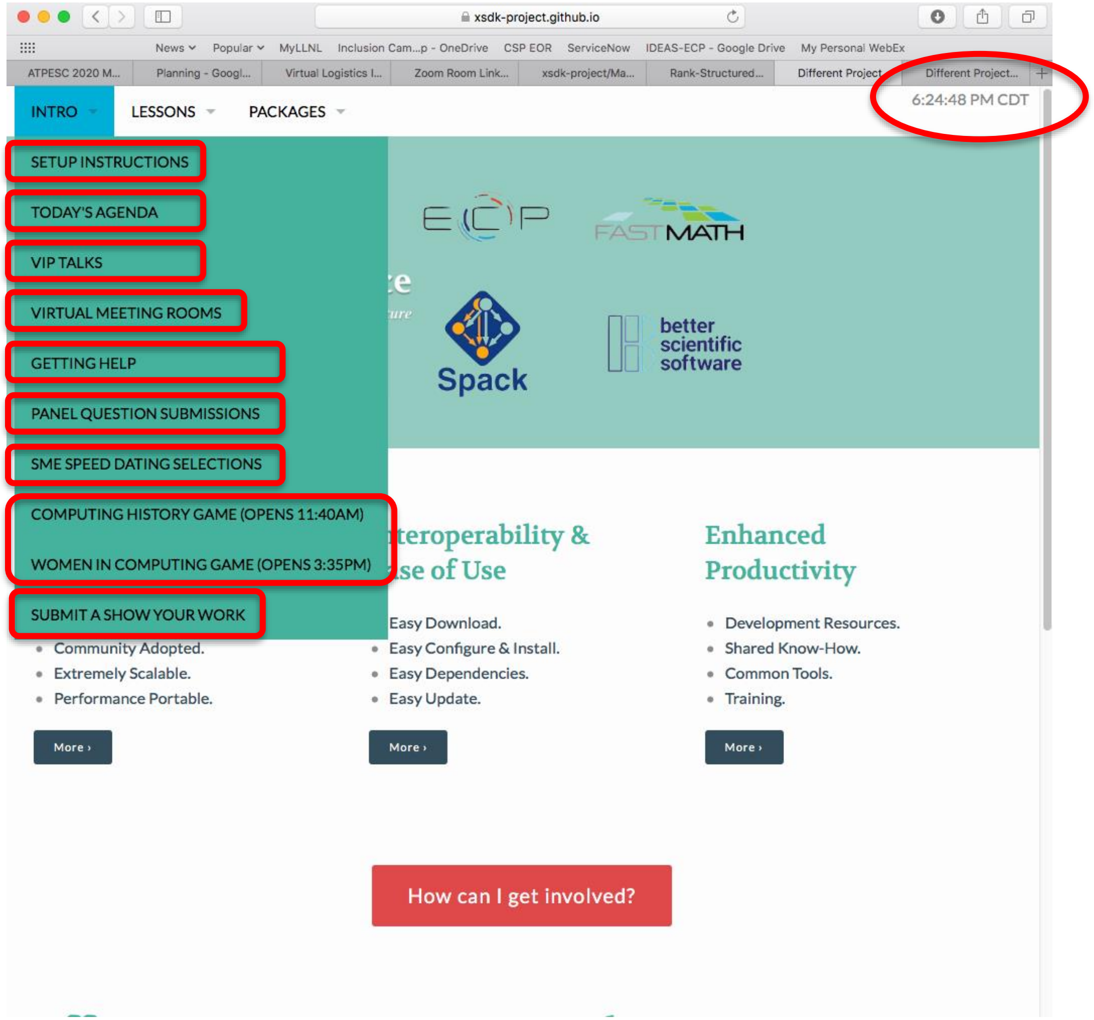
#1
- **Structured Discretization** (AMReX)
- **Unstructured Discretization** (MFEM/PUMI)
- **Iterative Solvers & Preconditioners** (Trilinos/MueLU)
- **Direct Solvers** (SuperLU/Strumpack)

#2
- **Structured Discretization** (AMReX)
- **Unstructured Discretization** (MFEM/PUMI)
- **Iterative Solvers & Preconditioners** (hypre)
- **Direct Solvers** (SuperLU/Strumpack)

#3
- **Nonlinear Solvers** (PETSc)
- **Optimization** (TAO)
- **Time Interation** (SUNDIALS)
- **Iterative Solvers & Preconditioners** (Trilinos/MueLU)

#4
- **Nonlinear Solvers** (PETSc)
- **Optimization** (TAO)
- **Time Interation** (SUNDIALS)
- **Direct Solvers** (SuperLU/Strumpack)

https://xsdk-project.github.io/MathPackagesTraining2020/



- Clock
- Setup instructions
- Today's agenda
- VIP talks
- Virtual meeting rooms
- Getting help
- Panel question submission
- SME speed dating selections
- Break games
- Submit a "Show your work"

EXASCALE COMPUTING PROJECT

Today's agenda

https://xsdk-project.github.io/MathPackagesTraining2020/agenda

Mix-n-Match topics
to your interests
See Synopses from Agenda

Structured Discretization (with AMReX)

Slides

Block-structured adaptive mesh refinement (AMR) provides a natural framework in which to focus computing power on the most critical parts of the problem in the most computationally efficient way possible. AMReX supports the development of block-structured AMR algorithms for solving systems of partial differential equations (PDE's) and other algorithms that require structured mesh and/or particle discretizations. We will begin with an overview of block-structured AMR, including several different time-stepping strategies, and then discuss the features of AMReX we might want to use to solve a multiphysics problem on machines from laptops to supercomputers. Hands-on exercises will include passive scalar advection with time-dependent adaptivity, the use of native linear solvers to impose incompressibility on a flow around obstacles, and "AMReX-Pachinko", which demonstrates the interaction of particles with objects.

Iterative Solvers & Preconditioners (with MueLu)

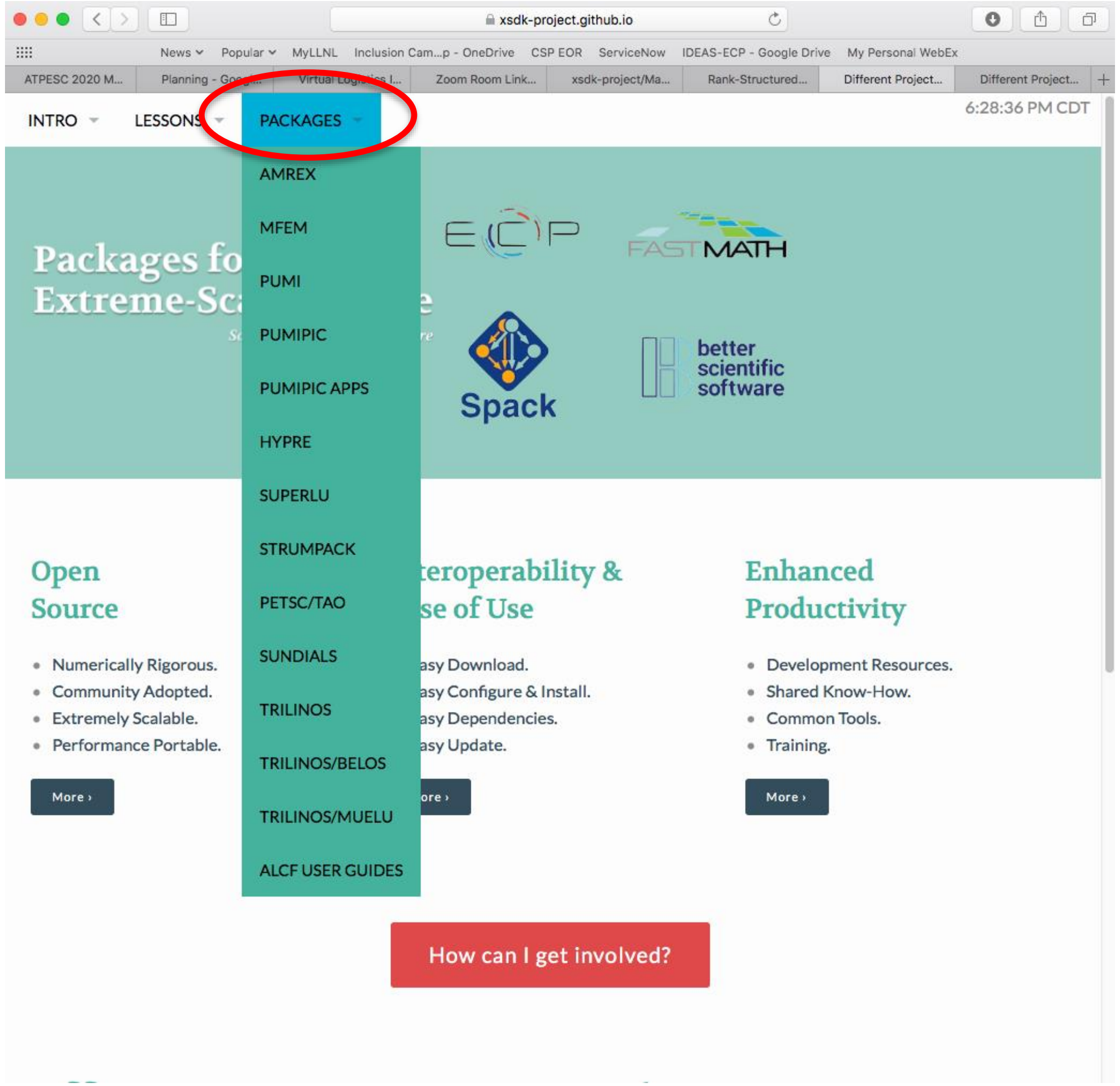
Slides

In this session, attendees will learn about linear solvers and preconditioners available in the Trilinos project. We will focus on Krylov solvers such as conjugate gradients (CG) and generalized minimum residual (GMRES); simple preconditioners like Jacobi, Gauss-Seidel, and Chebyshev polynomials; and scalable aggregation-based algebraic multigrid preconditioning. The two hands-on lessons will provide an opportunity to run a variety of stand-alone examples that demonstrate some of the many Trilinos solver capabilities on a model linear problem.

CDT Start	Mins	Topic	Speaker(s)	Virtual Venue
09:30	55	Intro. to Numerical Libraries	Lois Curfman McInnes Mark Miller	Main-Room
10:25	5	Telecon Transition		
10:30	60	**Parallel Session One**		
		Structured Discretization (with AMReX)	Ann Almgren Don Willcox	Frontier
		Unstructured Discretization (with MFEM/PUMI)	Aaron Fisher Mark Shephard	Aurora
		Iterative Solvers & Preconditioners (with MueLu)	Jonathan Hu Christian Glusa	Perlmutter
		Direct Solvers (with SuperLU/Strumpack)	Sherry Li Pieter Ghysels	El-Capitan

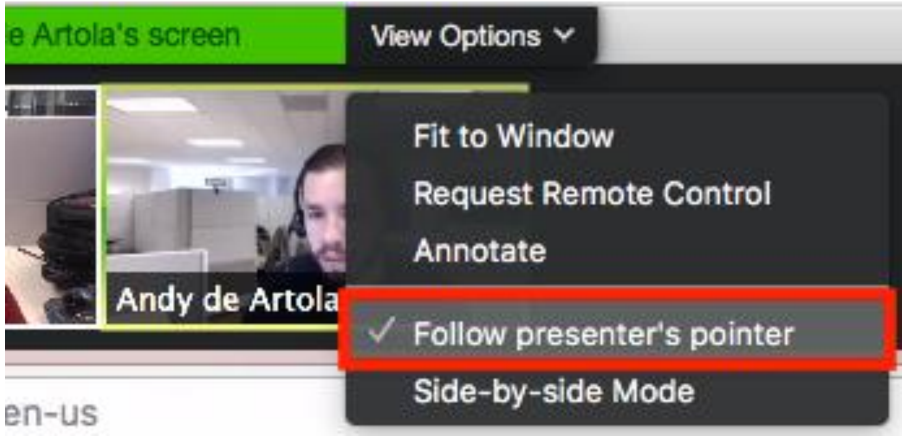
EXASCALE COMPUTING PROJECT

https://xsdk-project.github.io/MathPackagesTraining2020/



- Hands-on Lessons
- Packages

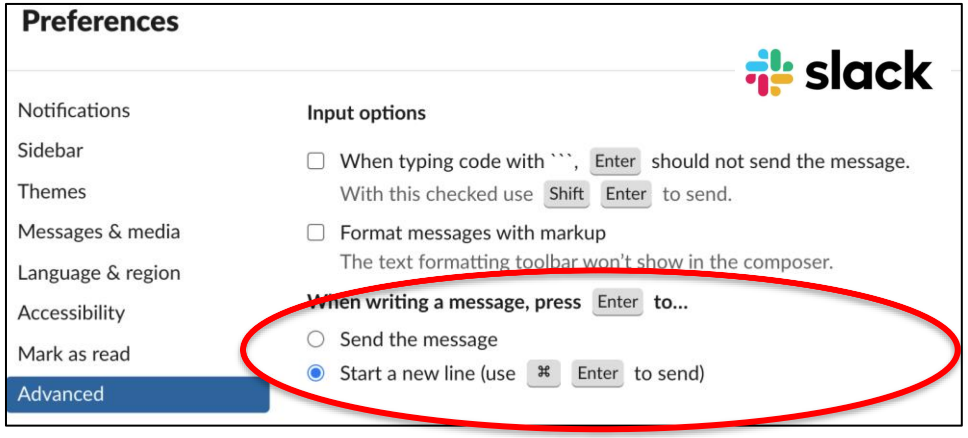
Using Zoom



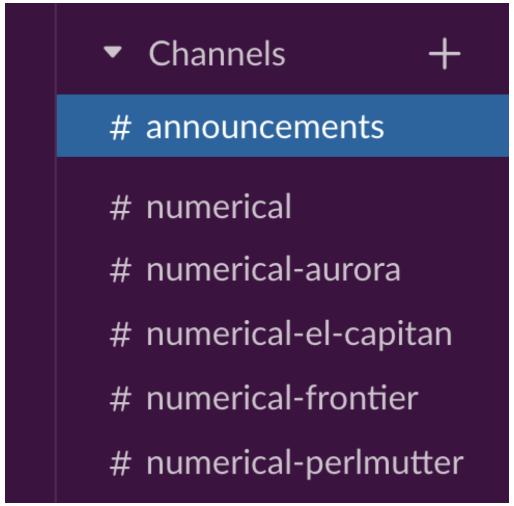
- Please stay muted unless asked to un-mute
- We're using Slack for chat, not Zoom's chat
- "Follow presenter's pointer" might be helpful
  - Available only if NOT in "Fit to Window" mode
- Download slide PDFs from ATPESC web site (agenda page) ahead of presentation as a backup
- Other useful tips
  - Better performance if also disable your video
  - Stop other streaming activity in your home if you can

Using Slack

Tip: Consider setting Preferences to customize when to send

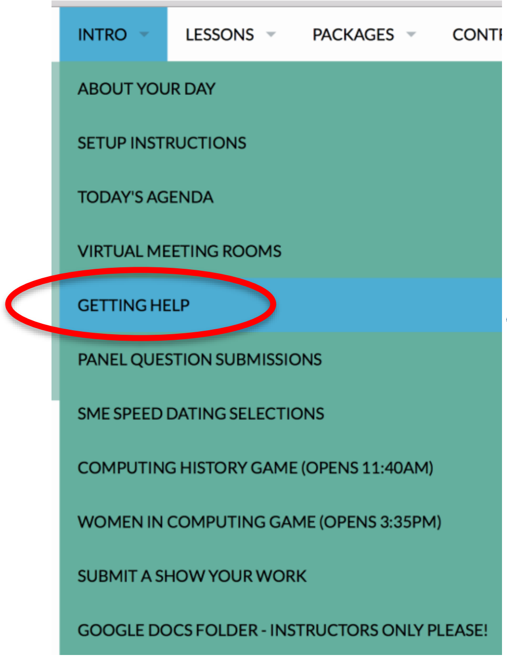


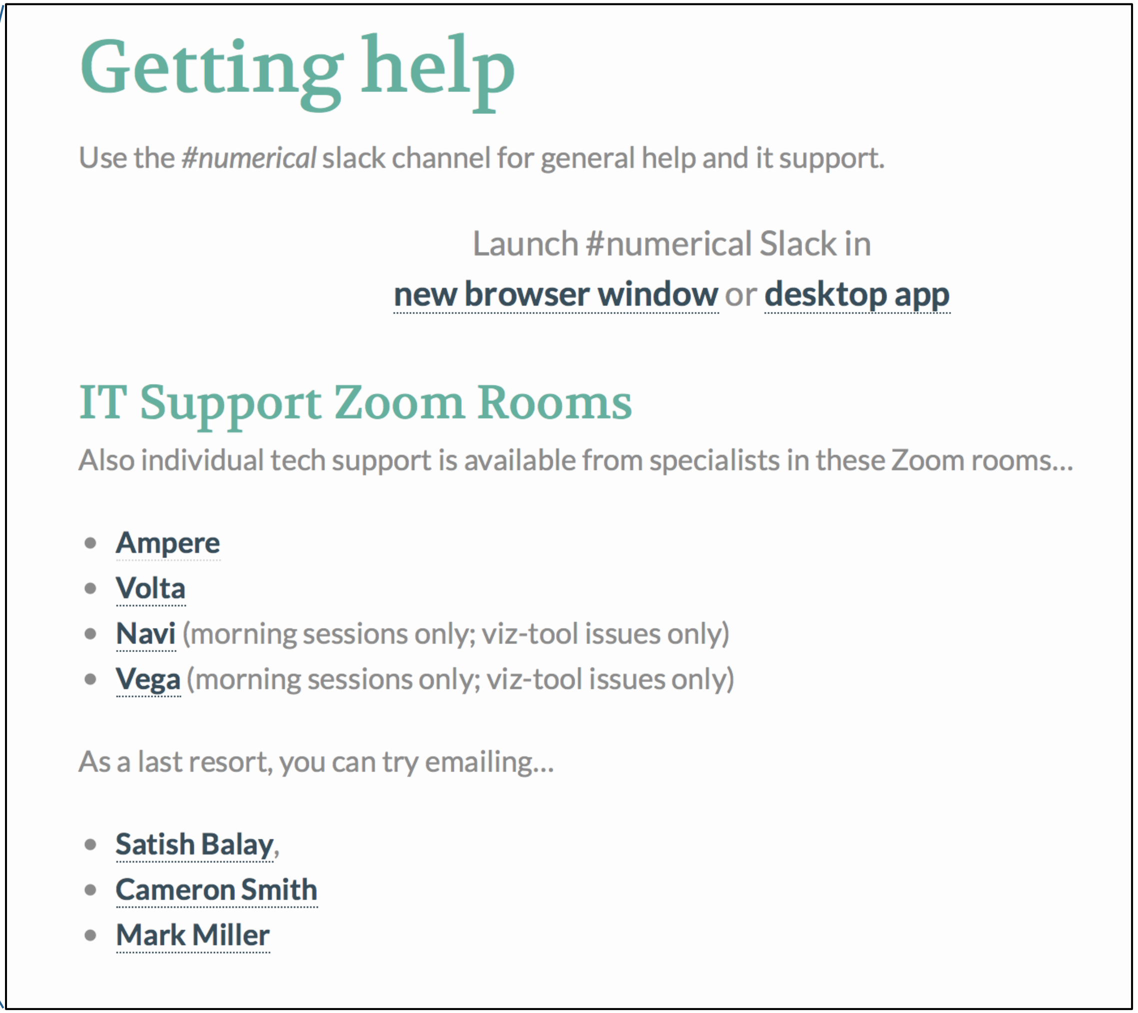
- Recommend using the desktop app, but browser ok too
- **#numerical** channel
  - For all chat during presentations in “Main room”
  - For all chat outside any specific parallel session
  - For general help
  - Recommend using the thread option to help keep track of discussions on subtopics
- **#numerical-<room-name>** (e.g., **#numerical-aurora**)
  - For all chat during presentations in the associated room
  - Room chat restricted to discussion <u>on the current presentation topic only</u>
    - To continue questions/discussion on topics presented earlier in the day, transition to the **#numerical** channel or direct slack messages to individuals in the ATPESC numerical software team



Getting help ("Getting help" menu item)

- **#numerical** Slack channel
- IT Support Rooms under the "Getting Help" menu





Getting help

Use the *#numerical* slack channel for general help and it support.

Launch #numerical Slack in
new browser window or **desktop app**

IT Support Zoom Rooms

Also individual tech support is available from specialists in these Zoom rooms...

- **Ampere**
- **Volta**
- **Navi** (morning sessions only; viz-tool issues only)
- **Vega** (morning sessions only; viz-tool issues only)

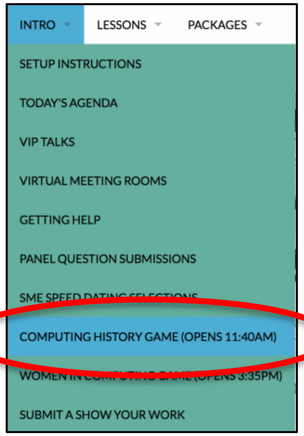
As a last resort, you can try emailing...

- **Satish Balay**,
- **Cameron Smith**
- **Mark Miller**

EXASCALE COMPUTING PROJECT

During breaks and lunch

- During mid-morning and mid-afternoon 15-minute breaks, we will keep Zoom meetings open and allow unmuting for some informal dialog for those interested.
- During lunch will do the same with the “Main Room” – again, for anyone interested.
- Some simple computing history games for those interested
  - Ignore the points, scoring



The ATPESC Team 2020 on Zoom

Row 1:
- Dan Reynolds
- Lois Curfman McInnes
- Don Willcox
- Ann Almgren
- Satish Balay

Row 2:
- Pieter Ghysels
- Christian Glusa
- Mark Miller
- Aaron Fisher
- Sherry Li

Row 3:
- Sara Osborn
- David Gardner
- Mark Shephard
- Ulrike Yang
- Alp Dener

Row 4:
- Richard Mills
- Jonathan Hu

Not shown:
- Cameron Smith
- Carol Woodward

EXASCALE COMPUTING PROJECT

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) and numerical optimization

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 provides 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

The ATPESC Team 2020

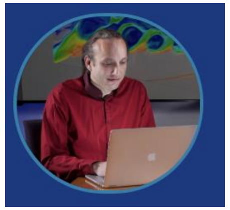
Extreme-scale numerical algorithms and software
Integrated lectures and hands-on examples, panel session, individual discussions … and more!


Mark Miller, LLNL


Cameron Smith, RPI


Carol Woodward, LLNL


Ann Almgren, LBL


Aaron Fisher, LLNL


Jonathan Hu, SNL


Richard Mills, ANL


Don Willcox, LBL


Ulrike Yang, LLNL


Satish Balay, ANL


Pieter Ghysels, LBL


Sherry Li, LBL


Dan Reynolds, SMU


Alp Dener, ANL


Christian Glusa, SNL


Lois Curfman McInnes, ANL


Mark Shephard, RPI

Thank you to David Gardner and Sara Osborn, LLNL

Additional contributors to gallery of highlights:

Various HPC package developers

EXASCALE COMPUTING PROJECT

VIPs of ATPESC Extreme-Scale Numerical Software Track

- **Jim Demmel, UC Berkeley** [bio]
  - Communication-Avoiding Algorithms for Linear Algebra, Machine Learning, and Beyond
    - ATPESC 2019 [slides, video]
    - ENLA Seminar, June 2020 [video]

- **Jack Dongarra, Univ of Tennessee** [bio]
  - Adaptive Linear Solvers and Eigensolvers, ATPESC 2019 [slides, video]

- **David Keyes, KAUST** [bio]
  - The Convergence of Big Data and Large-scale Simulation: Leveraging the Continuum, ATPESC 2019 [slides, video]
  - Algorithmic Adaptations to Extreme Scale Computing, ATPESC 2018 [slides, video]

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

EXASCALE COMPUTING PROJECT
https://exascaleproject.org

https://fastmath-scidac.llnl.gov

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

… 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 and Sustainability (Aug 6)

Community efforts: Join us!

https://xsdk.info

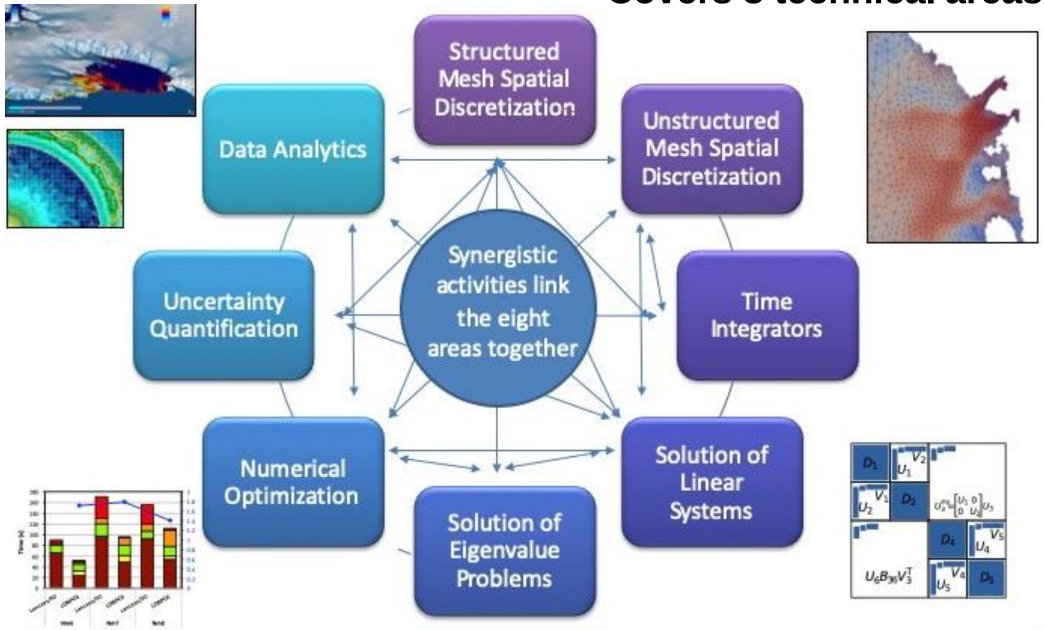
https://bssw.io

https://e4s.io

FASTMath: Frameworks, Algorithms & Scalable Technologies for Mathematics

https://fastmath-scidac.llnl.gov

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

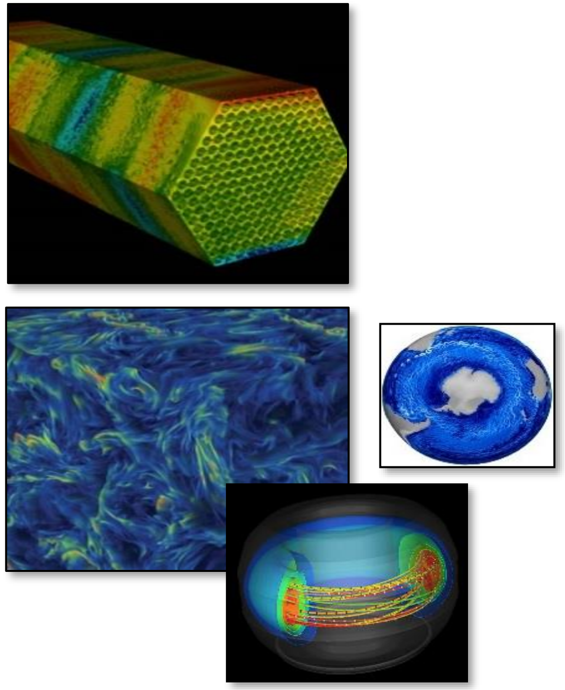
100's of person years of experience building math software

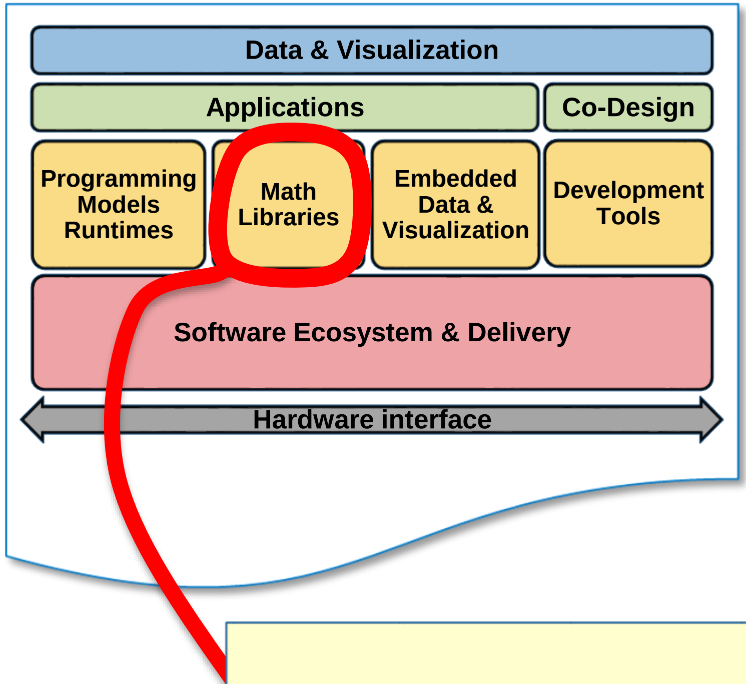
SuperLU, AMReX, sundials, ZOLTAN, mfem, PETSc, hypre, UQTk, symPACK, PUMI, Trilinos, Albany, DAKOTA

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

ECP's holistic approach uses co-design and integration to achieve exascale computing

Application Development	Software Technology	Hardware and Integration
Science and mission applications	Scalable software stack	Relationships: facilities with AD/ST, with vendors





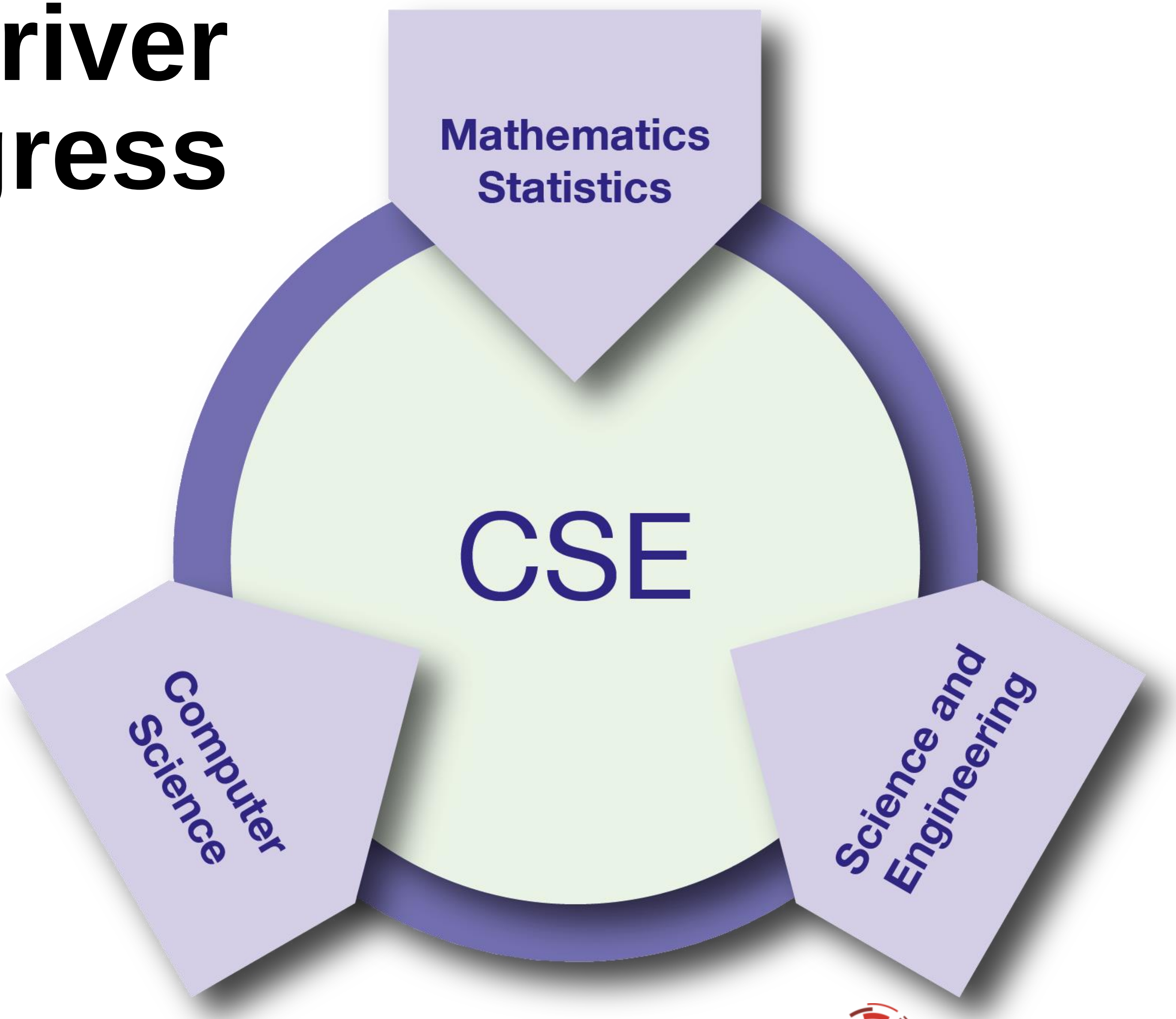
Emphasis for this presentation

CSE: Essential driver of scientific progress

CSE = Computational Science & Engineering

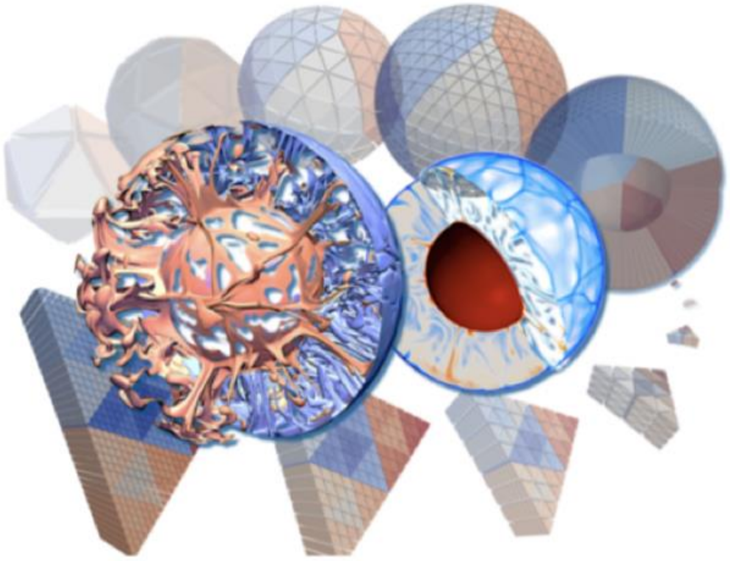
Development and use of computational methods for scientific discovery

- 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

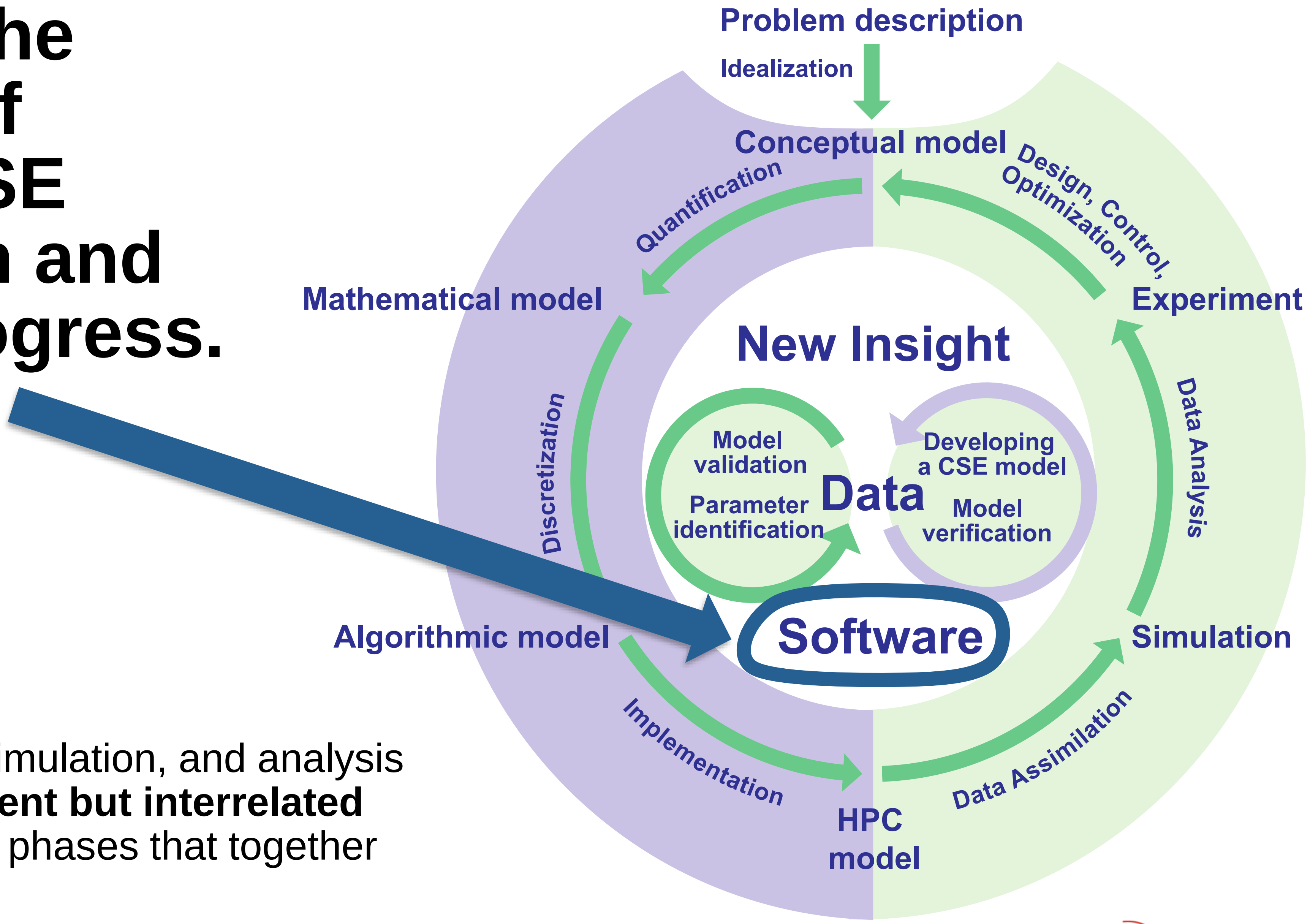
- Mathematical methods and algorithms
- CSE and HPC: Ubiquitous parallelism
- CSE and the data revolution
- CSE software
- CSE education & workforce development



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.



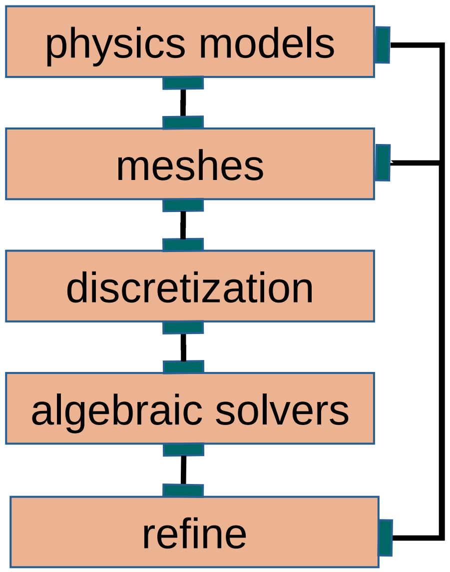
CSE cycle: Modeling, simulation, and analysis

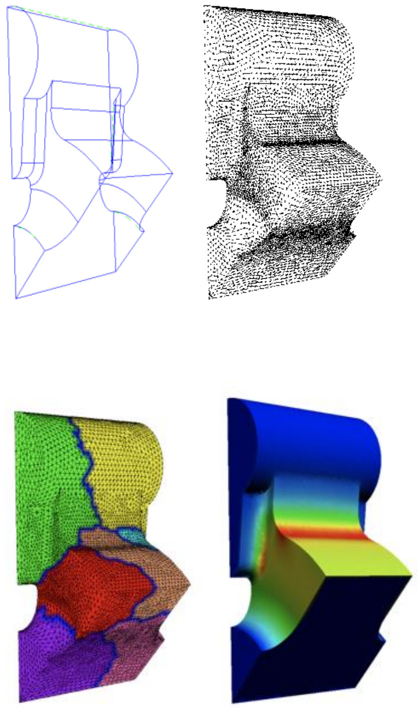
- **Software: independent but interrelated elements** for various phases that together enable CSE

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

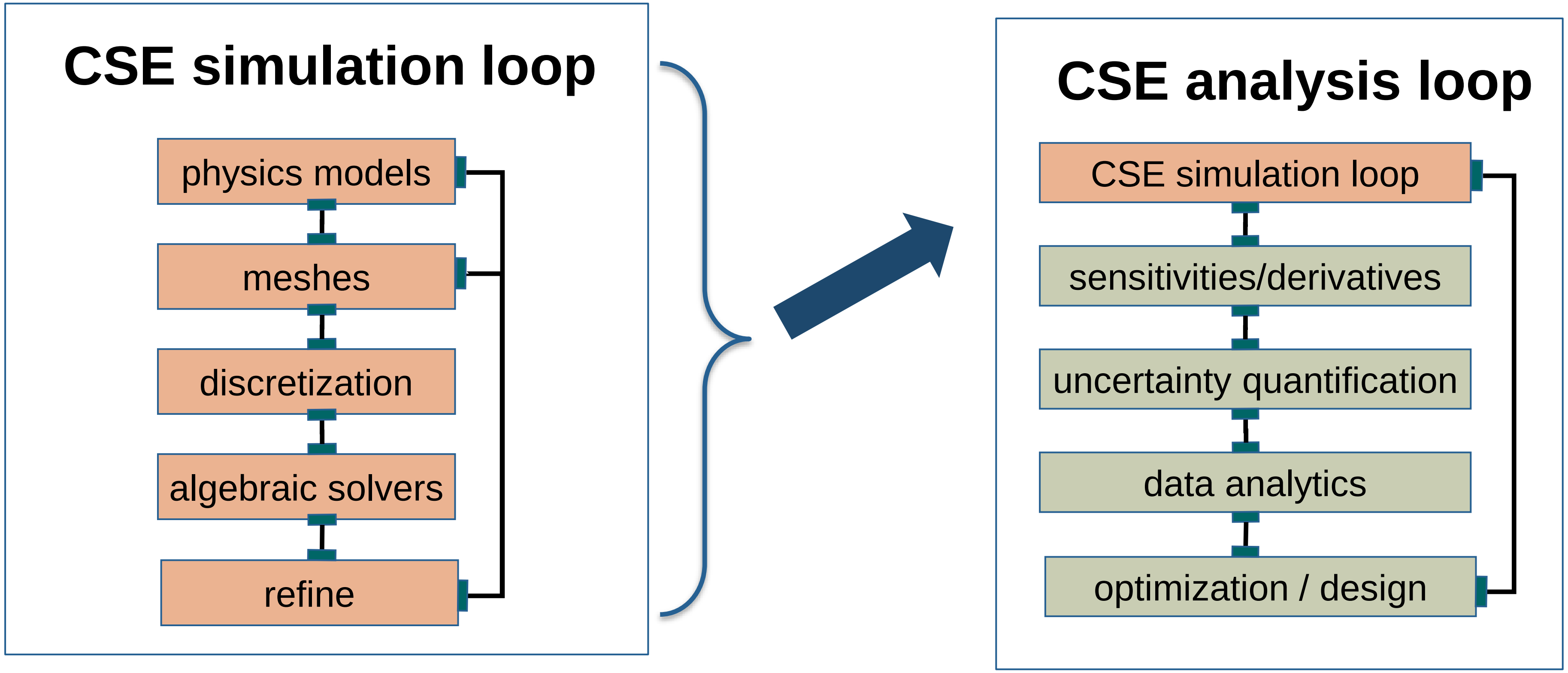
CSE simulation loop





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

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

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

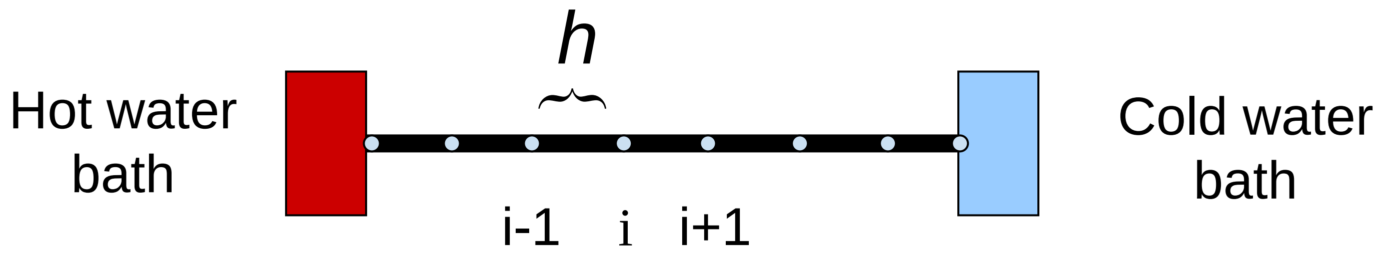
The first step is to discretize the equations

- 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

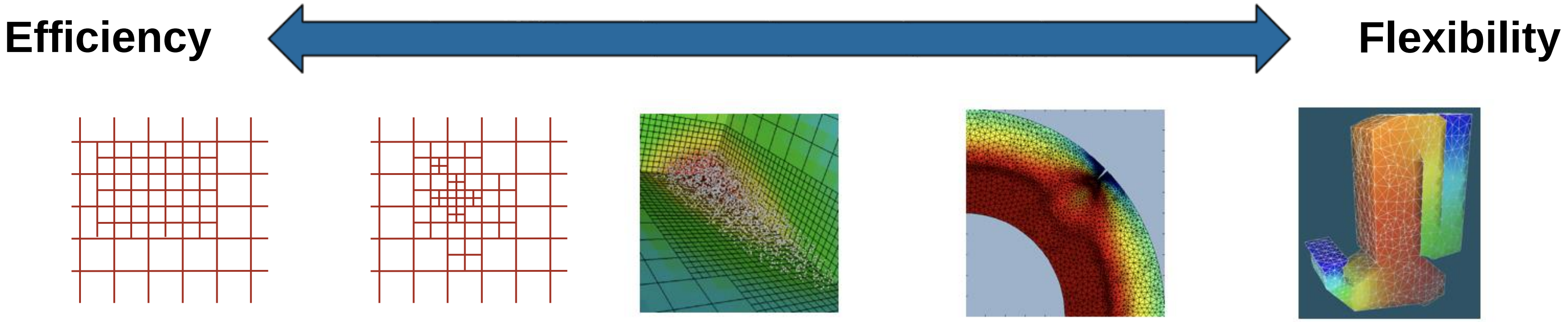
- Set up a matrix of the unknown coefficients
  - include the known boundary conditions
- Solve the linear system for T_i

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

- Visualize and analyze the results

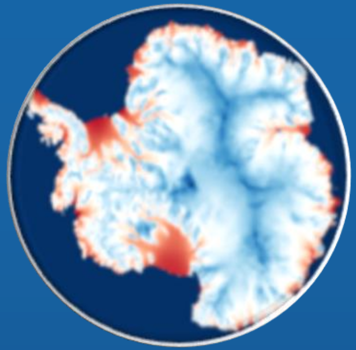
As problems get more complicated, so do the steps in the process

- Different discretization strategies exist for differing needs

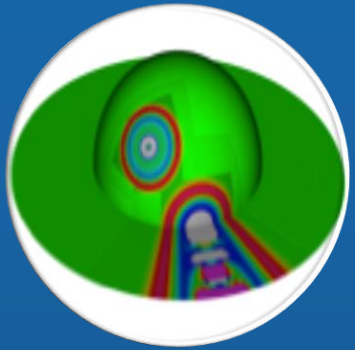


- Most problems are time dependent and nonlinear
  - Need higher algorithmic levels than linear solvers
- Increasingly combining multiple physical processes
  - Interactions require careful handling
- Goal-oriented problem solving requires optimization, uncertainty quantification

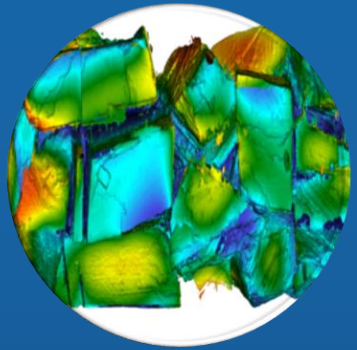
Structured grid efforts focus on high-order, mapped grids, embedded boundaries, AMR, and particles



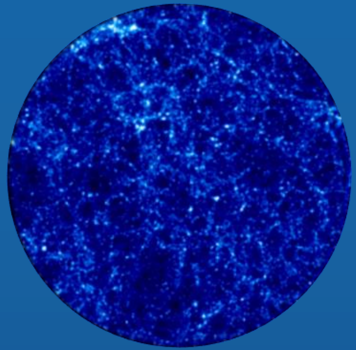
Structured AMR



Mapped-multiblock grids



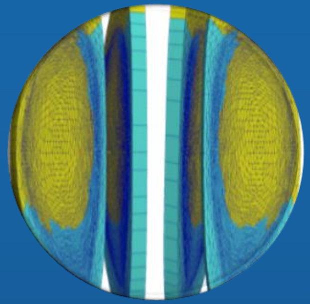
Embedded boundary methods



Particle-based methods

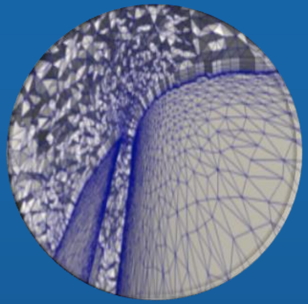
Application to cosmology, astrophysics, accelerator modeling, fusion, climate, subsurface reacting flows, low mach number combustion, etc.

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

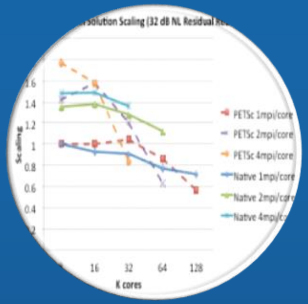


Parallel mesh infrastructures

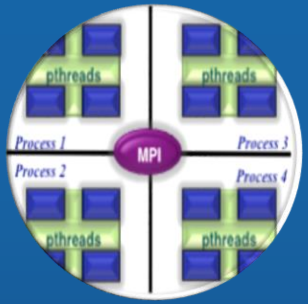
Dynamic load balancing



Mesh adaptation and quality control



Parallel performance on unstructured meshes

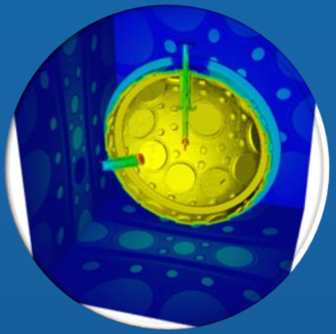


Architecture aware implementations

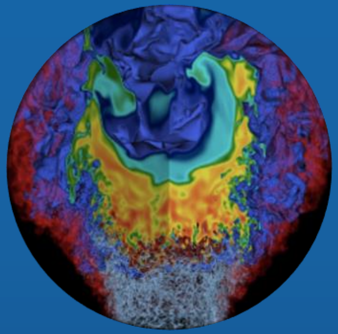
Application to fusion, climate, accelerator modeling, NNSA applications, nuclear energy, manufacturing processes, etc.

EXASCALE COMPUTING PROJECT

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

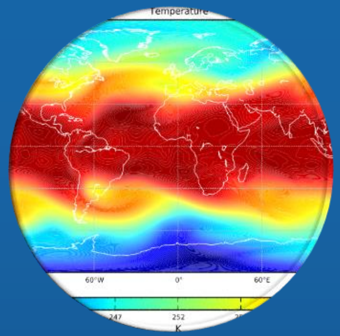


IMEX, multi-rate methods



Spectral deferred correction techniques

Parallel in time



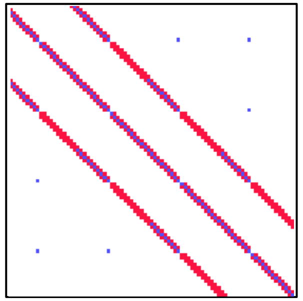
Constrained problems

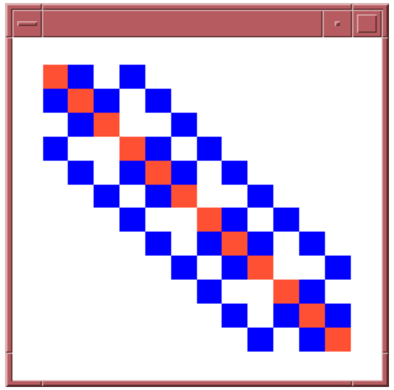
Application to atmospheric dynamics, ice sheets, combustion, power grid, cosmology, subsurface, etc.

ECP EXASCALE COMPUTING PROJECT

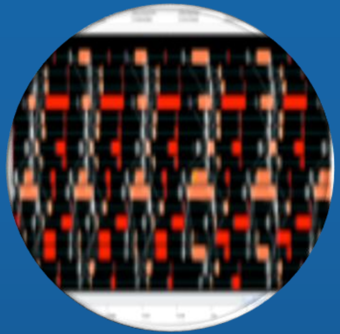
As problems grow in size, so do corresponding discrete systems

- 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.

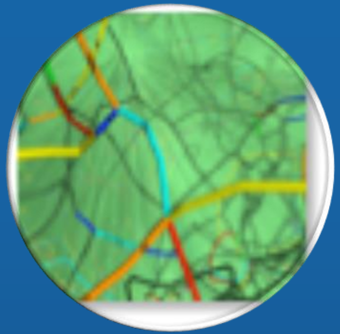




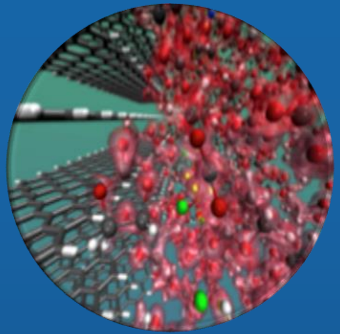
Research on algebraic systems provides key solution technologies to applications



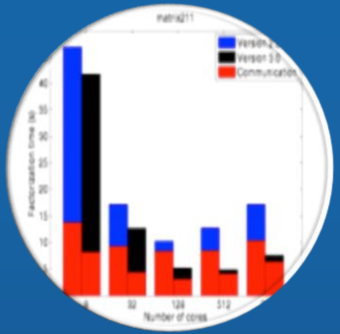
Linear system solution using direct and iterative solvers



Nonlinear system solution using acceleration techniques and globalized Newton methods



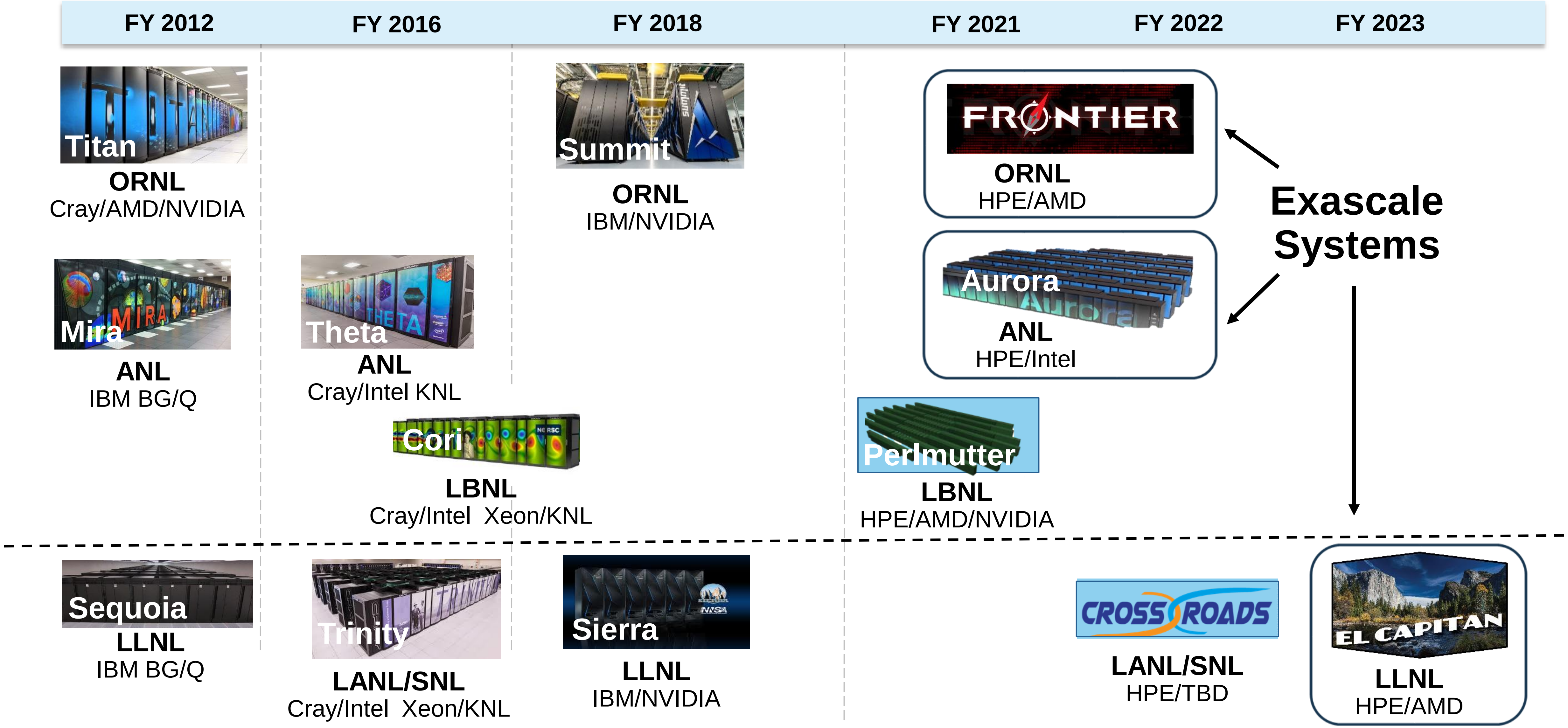
Eigensolvers using iterative techniques and optimization



Architecture aware implementations

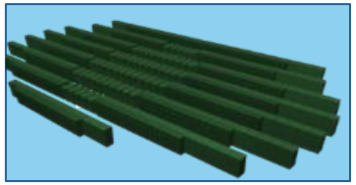
Application to fusion, nuclear structure calculation, quantum chemistry, accelerator modeling, climate, dislocation dynamics etc,

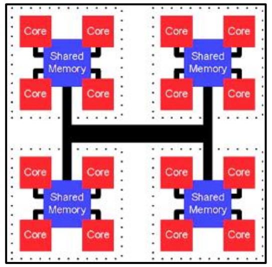
DOE HPC Roadmap to Exascale Systems

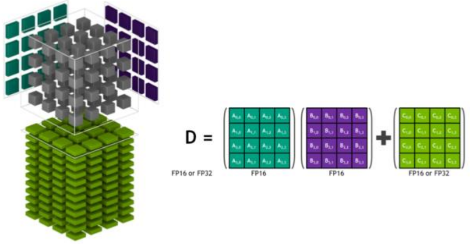


Disruptive changes in HPC architectures

- **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.







- **Research advances: On-node and inter-node capabilities**
  - Reduce communication and synchronization
  - Increase concurrency
  - Address memory footprint
  - Enable large communication/computation overlap
  - Use GPUs and multithreading
  - Compare task and data parallelism
  - Low-level kernels for vector operations that support hybrid programming models
  - Mixed precision (leverage compute power available in low-precision tensor cores)
  - Etc.

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.github.io/
- **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!

EXASCALE COMPUTING PROJECT

ECP applications need sustainable coordination among math libraries

ECP AD Teams

Combustion-Pele, EXAALT, ExaAM, ExaFEL, ExaSGD, ExaSky, ExaStar, ExaWind, GAMESS, MFIX-Exa, NWChemEx, Subsurface, WarpX, WDMApp, WarpX, ExaAM, ATDM (LANL, LLNL, SNL) apps, AMReX, CEED, CODAR, CoPA, ExaLearn

ECP Math Libraries

hypre (high performance preconditioners), PETSc, TAO, Trilinos, STRUMPACK, MFEM, SuperLU, sundials, MAGMA, PLASMA, ArborX, DTK, ECP SLATE ICL, FFTX, heFFTe, libEnsemble, Ginkgo, TASMANIAN

Examples:

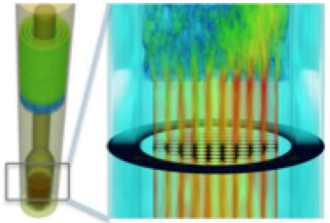
- **ExaAM:** DTK, hypre, PETSc, Sundials, Tasmanian, Trilinos, FFT, etc.
- **ExaWind:** hypre, KokkosKernels, SuperLU, Trilinos, FFT, etc.
- **WDMApp:** PETSc, hypre, SuperLU, STRUMPACK, FFT, etc.
- **CEED:** MFEM, MAGMA, hypre, PETSc, SuperLU, Sundials, etc.
- And many more …

ECP EXASCALE COMPUTING PROJECT

Multiphysics: A primary motivator for exascale

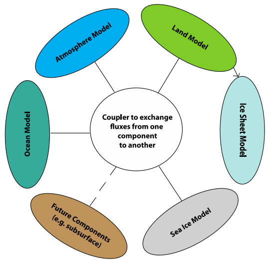
Multiphysics: greater than 1 component governed by its own principle(s) for evolution or equilibrium

– Also: broad class of coarsely partitioned problems possess similarities

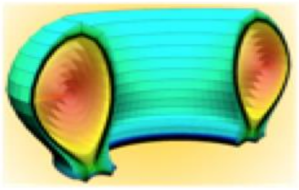


nuclear reactors
A. Siegel, ANL

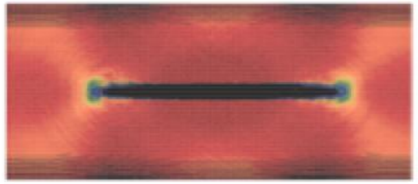
particle accelerators
K. Lee, SLAC



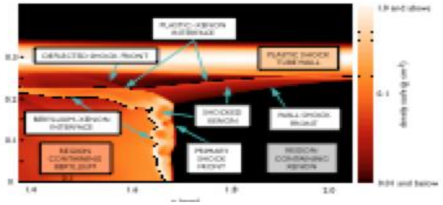
climate
K. Evans, ORNL



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
© The Author(s) 2012
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
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

EXASCALE COMPUTING PROJECT

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

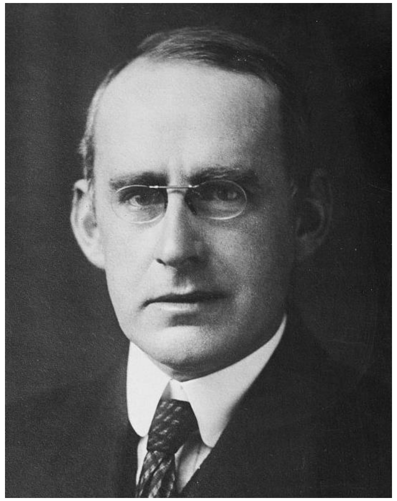
- **Need consistency** of compiler (+version, options), 3rd-party packages, etc.
- **Namespace and version conflicts** make simultaneous build/link of packages difficult
- **Multilayer interoperability** requires careful design and sustainable coordination

Need software <u>ecosystem</u> 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

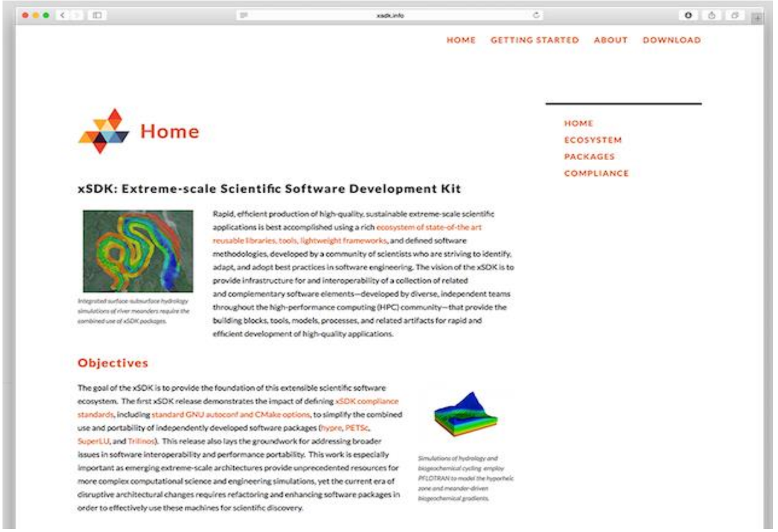
EXASCALE COMPUTING PROJECT

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



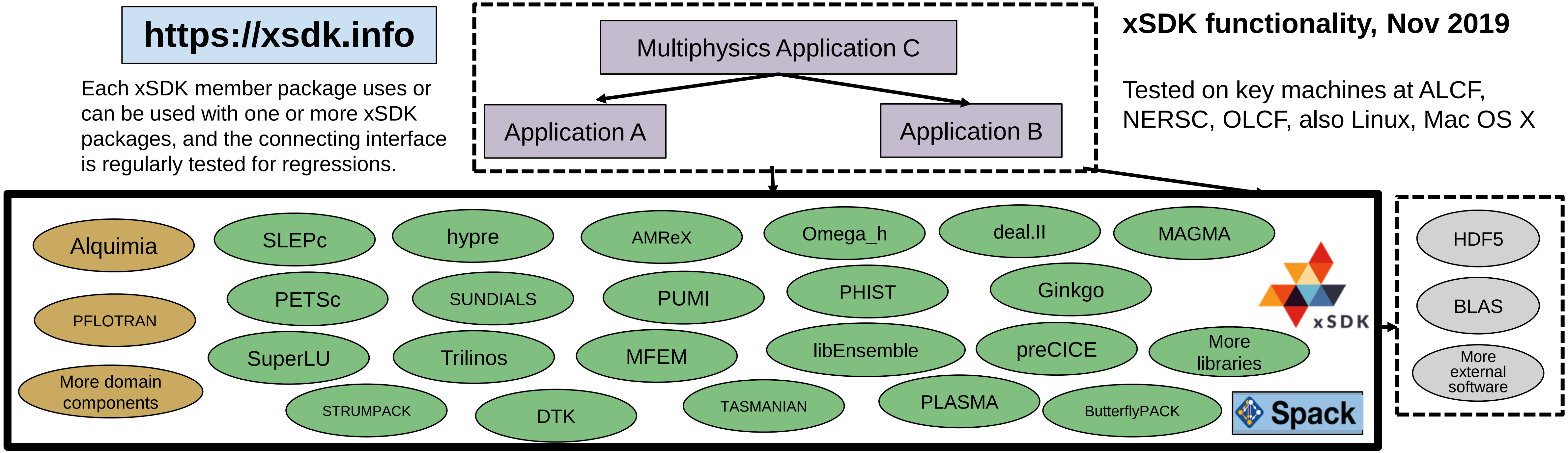
website: xSDK.info

xSDK Version 0.5.0: November 2019

https://xsdk.info

https://xsdk.info

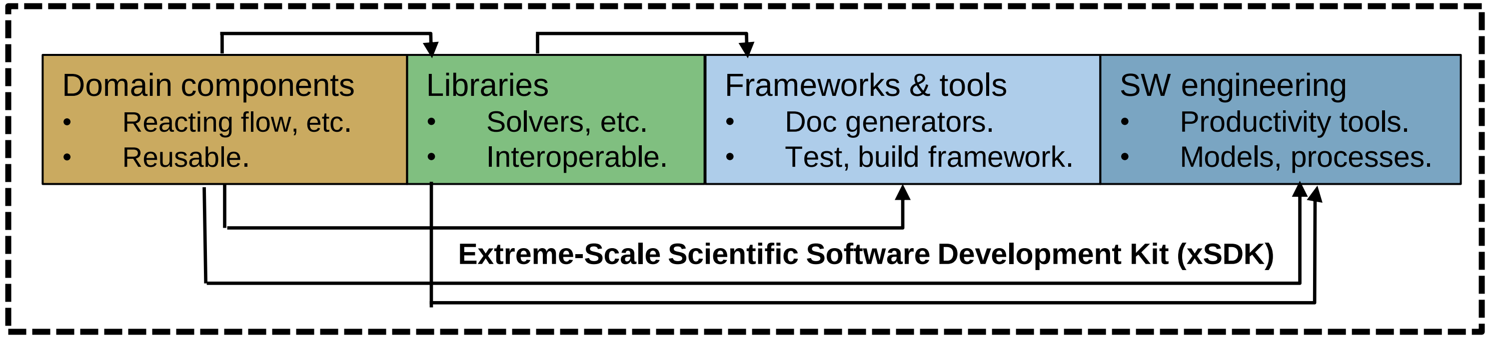
Each xSDK member package uses or can be used with one or more xSDK packages, and the connecting interface is regularly tested for regressions.



xSDK functionality, Nov 2019

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

November 2019
- 21 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

EXASCALE COMPUTING PROJECT

xSDK collaborators

xSDK Release 0.5.0, Nov 2019

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

and many more …

- **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)
  - **Ginkgo**: Hartwig Anzt (Karlsruhe Institute of Technology)
  - **hypre**: Ulrike Meier Yang, Sarah Osborn, Rob Falgout (LLNL)
  - **libEnsemble**: Stefan Wild, Steve Hudson (ANL)
  - **MAGMA** and **PLASMA**: Piotr Luszczek (UTK)
  - **MFEM**: Aaron Fisher, Tzanio Kolev (LLNL)
  - **Omega_h**: Dan Ibanez (SNL)
  - **PETSc/TAO**: Satish Balay, Alp Denner, Barry Smith (ANL)
  - **preCICE:** Frederic Simonis (Technical University Munich)
  - **PUMI**: Cameron Smith (RPI)
  - **SUNDIALS**: Cody Balos, David Gardner, Carol Woodward (LLNL)
  - **SuperLU, STRUMPACK, ButterflyPACK**: Sherry Li, Pieter Ghysels, Yang Liu (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 (PNNL)

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.
R7. Have README, SUPPORT, LICENSE, and CHANGELOG file in top directory.

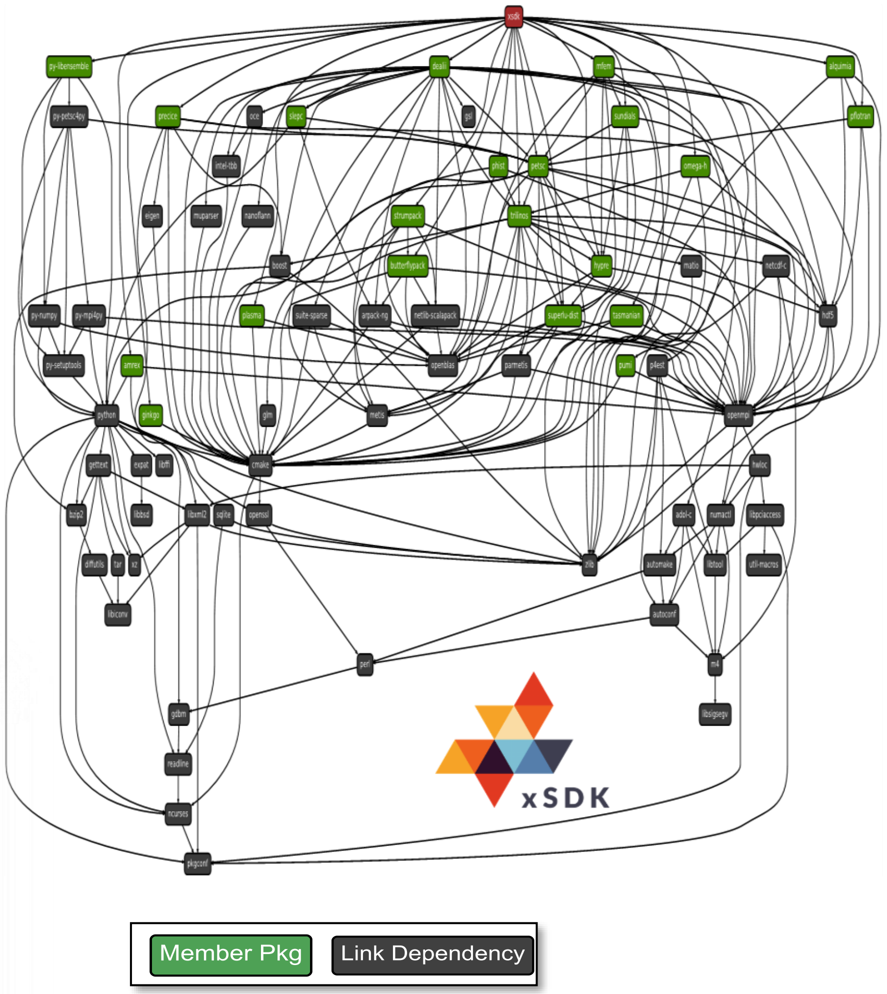
xSDK member package: Must be an xSDK-compatible package, *and* it uses or can be used by another package in the xSDK, and the connecting interface is regularly tested for regressions.

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

The xSDK is using Spack to deploy its software

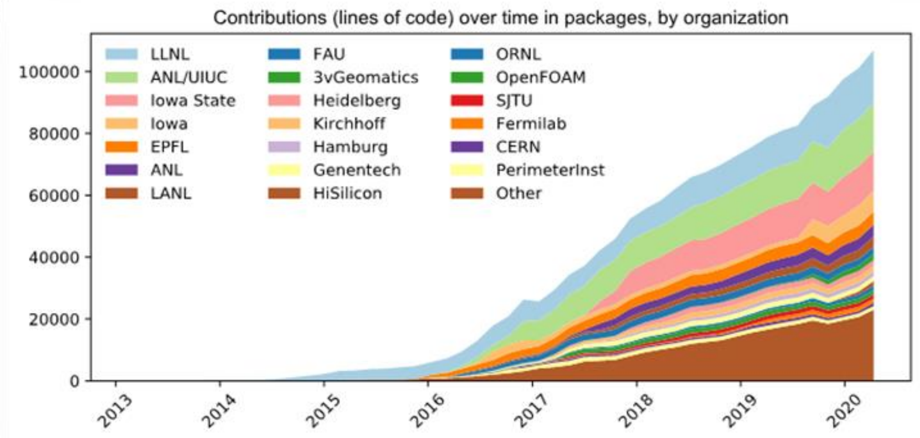
- The xSDK packages depend on a number of open source libraries
- Spack is a flexible package manager for HPC
- Spack allows the xSDK to be deployed with a single command
  - User can optionally choose compilers, build options, etc.
  - Will soon support combinatorial test dashboards for xSDK packages

Spack



Spack has grown into a thriving open source community

- Over 600 contributors
- Over4,300 software packages
- Used world-wide
- 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:

ECP Math libraries

- Performance on new node architectures
- Extreme strong scalability
- Advanced, coupled multiphysics, multiscale
- Optimization, UQ, solvers, discretizations
- Interoperability, complementarity: xSDK

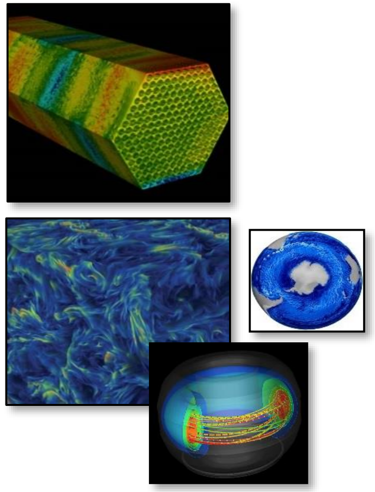
Next-generation algorithms

Advances in data structures for new node architectures

Improving library quality, sustainability, interoperability

Toward predictive scientific simulations

Increasing performance, portability, productivity

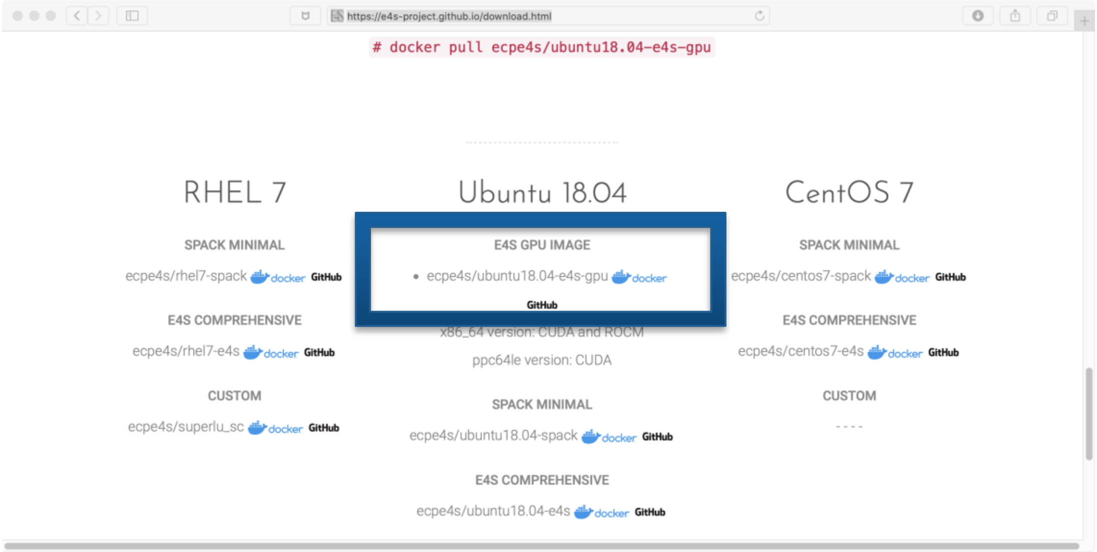


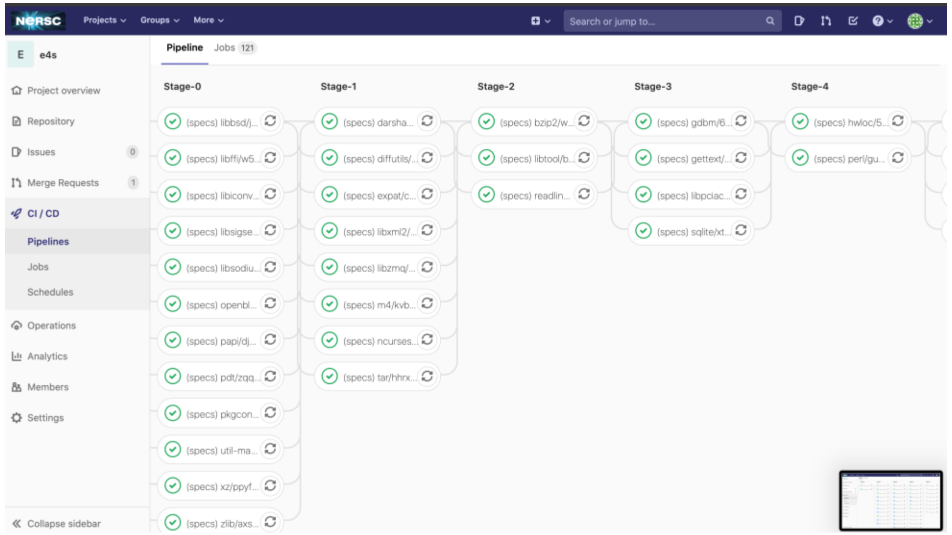
Timeline: xSDK release 1 → xSDK release 2 → . . . . . → xSDK release n

Extreme-scale Scientific Software Stack (E4S) https://e4s.io

- As our software gets more complex, it is getting harder to install tools and libraries correctly in an integrated and interoperable software stack.
- E4S is a community effort to provide open source software packages for developing, deploying, and running scientific applications on HPC platforms.
  - Delivering a modular, interoperable, and deployable software stack based on Spack [spack.io].
  - E4S provides both source builds and containers of a broad collection of HPC software packages.
  - E4S exists to accelerate the development, deployment and use of HPC software, lowering the barriers.
- E4S provides containers and turn-key, from-source builds of 50+ popular HPC software packages:
  - MPI: MPICH and OpenMPI
  - Development tools: TAU, HPCToolkit, and PAPI
  - Math libraries: hypre, PETSc, SUNDIALS, SuperLU, Trilinos
  - Data and Viz tools: Adios, HDF5, and Paraview
- E4S containers support Docker, Singularity, Shifter, and Charliecloud HPC container runtimes.
- E4S Spack build cache has over 10,000 binaries.
- Platforms: x86_64, ppc64le, and aarch64. GPUs runtimes: NVIDIA (CUDA) and AMD (ROCm).
- E4S DocPortal provide a single online location for *accurate* product descriptions for software products.
- E4S helps applications reduce the burden to install dependencies:
  - WDMapp installation speeds up from hours to minutes on Rhea at OLCF [https://wdmapp.readthedocs.io/en/latest/machines/rhea.html]

Download E4S v1.1 GPU image





E4S build pipeline
- Cori, NERSC

- 50 ECP ST products
- CUDA
- ROCm
- Tensorflow
- PyTorch

https://e4s.io

https://spack.io

EXASCALE COMPUTING PROJECT

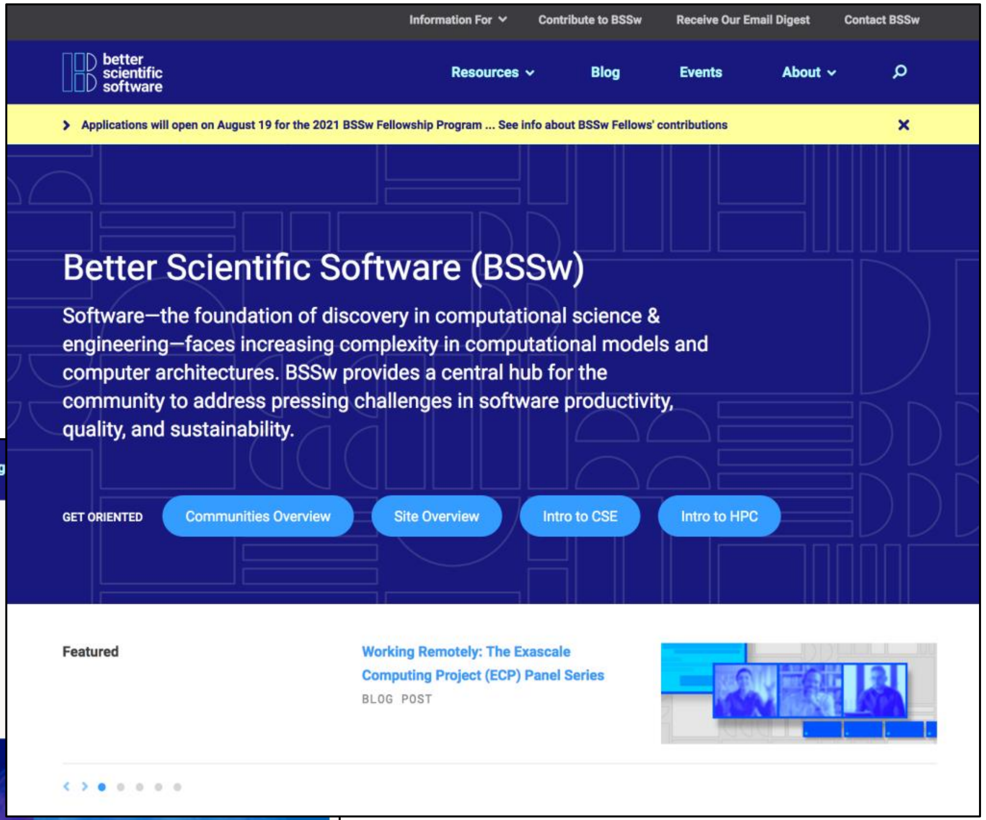
E4S Summary

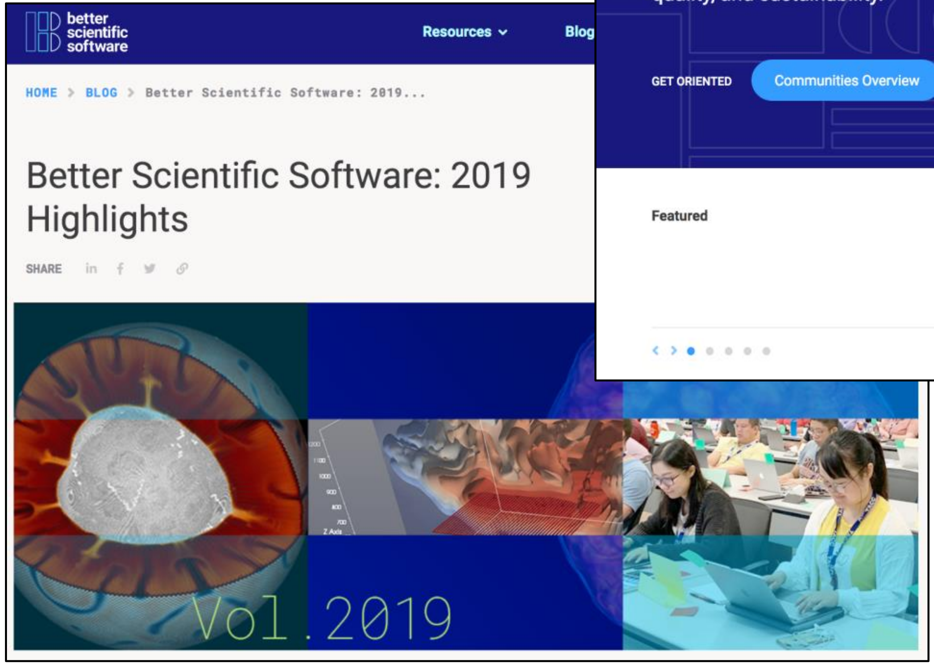
What E4S is not	What E4S is
A closed system taking contributions only from DOE software development teams.	Extensible, open architecture software ecosystem accepting contributions from US and international teams. Framework for collaborative open-source product integration.
A monolithic, take-it-or-leave-it software behemoth.	A full collection if compatible software capabilities **and** A manifest of a la carte selectable software capabilities.
A commercial product.	Vehicle for delivering high-quality reusable software products in collaboration with others.
A simple packaging of existing software.	The conduit for future leading edge HPC software targeting scalable next-generation computing platforms. A hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.

Further reading

https://bssw.io

- Building community through software policies, Piotr Luszczek and Ulrike Yang
- SuperLU: How advances in software practices are increasing sustainability and collaboration, Sherry Li
- Porting the Ginkgo package to AMD's HIP ecosystem, Hartwig Anzt
- Scientific software packages and their communities, Rene Gassmoeller
- Leading a scientific software project: It's all personal, Wolfgang Bangerth
- The art of writing scientific software in an academic environment, Hartwig Anzt
- Working Remotely: The Exascale Computing Project (ECP) panel series, Elaine Raybourn et al.
- Better Scientific Software: 2019 highlights, Rinku Gupta
- And many more …





See also Track 7:

Software Productivity & Sustainability (Aug 6)

Gallery of highlights

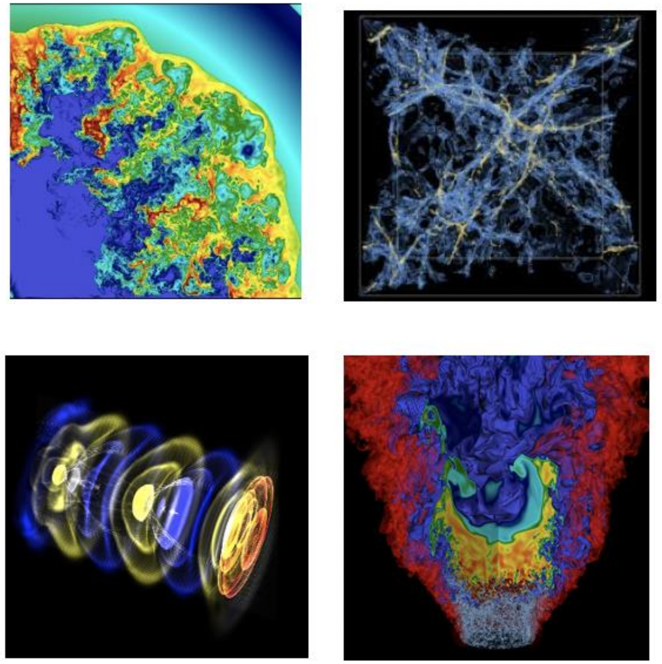
- Overview of some HPC numerical software packages
- 1 slide per package, emphasizing key capabilities, highlights, and where to go for more info
  - Listed first
    - Packages featured in ATPESC 2020 lectures and hands-on lessons
      - Developers are available for optional discussions
  - Listed 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



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

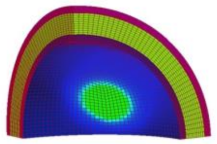
hypre

Lawrence Livermore National Laboratory

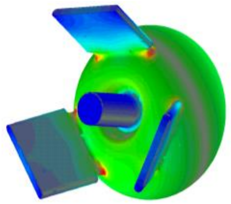
Highly scalable multilevel solvers and preconditioners. Unique user-friendly interfaces. Flexible software design. Used in a variety of applications. Freely available.

- **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
  - Provide CPU and GPU support
  - Available on github: https://www.github.com/LLNL/hypre

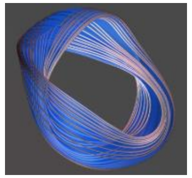
hypre high performance preconditioners



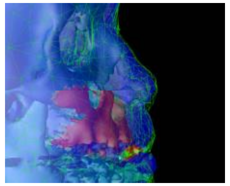
Elasticity / Plasticity



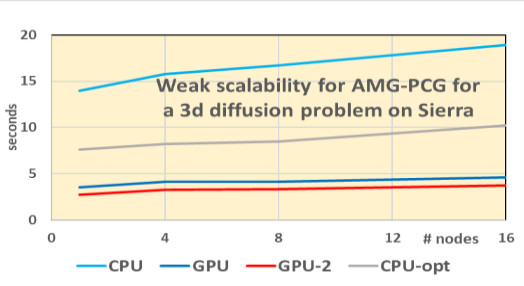
Electro-magnetics



Magneto-hydrodynamics



Facial surgery



xSDK

FASTMATH ECP R&D 100

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

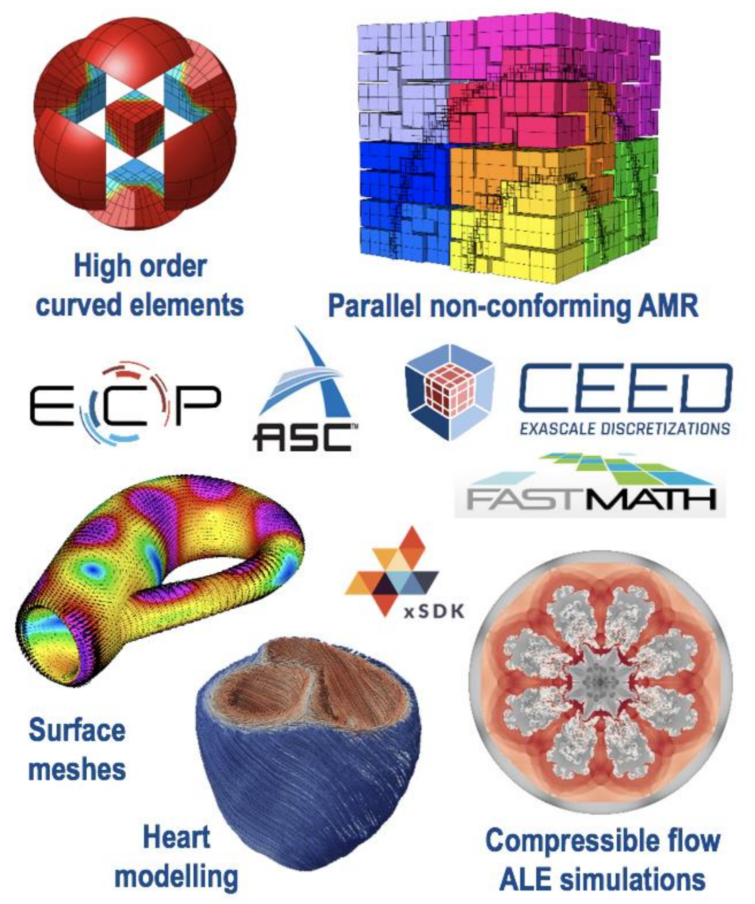
ECP EXASCALE COMPUTING PROJECT

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

PETSc TAO

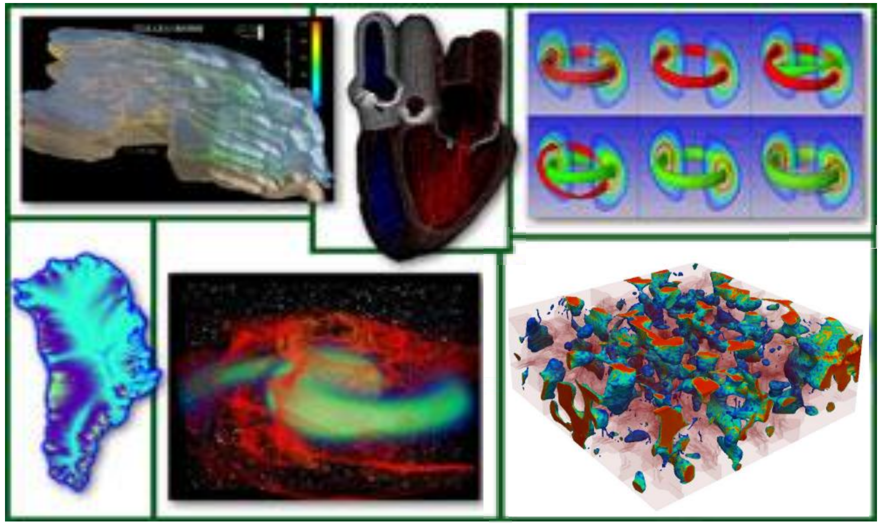
Portable, Extensible Toolkit for Scientific Computation / Toolkit for Advanced Optimization

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

- Optimization
- Time Integrators
- Nonlinear Algebraic Solvers
- Krylov Subspace Solvers
- Preconditioners
- Domain-Specific Interfaces
  - Networks
  - Quadtree / Octree
  - Unstructured Mesh
  - Structured Mesh
- Vectors
- Index Sets
- Matrices
- Computation & Communication Kernels

- **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

Parallel Unstructured Mesh Infrastructure

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

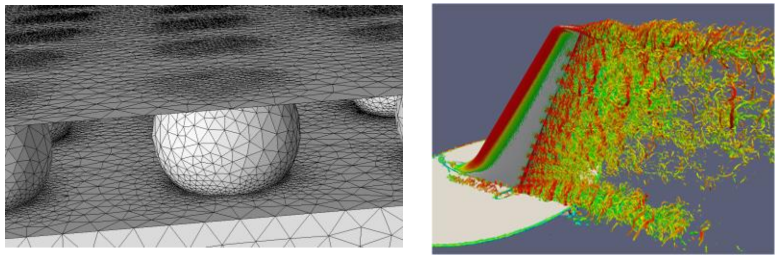
- **Core functionality**
  - Distributed, conformant mesh with entity migration, remote read only copies, fields and their operations
  - Link to the geometry and attributes
  - Mesh adaptation (straight and curved), mesh motion
  - Multi-criteria partition improvement
  - Distributed mesh support for 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
  - PetraM: Adaptive RF fusion
  - 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

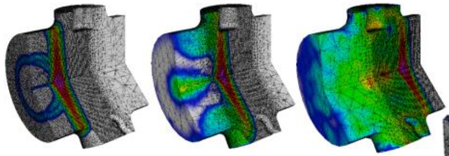
PUMi

SCOREC

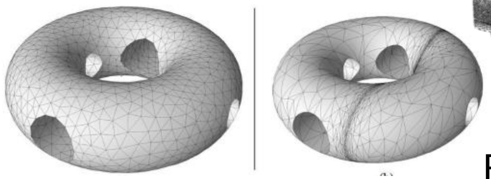
Rensselaer



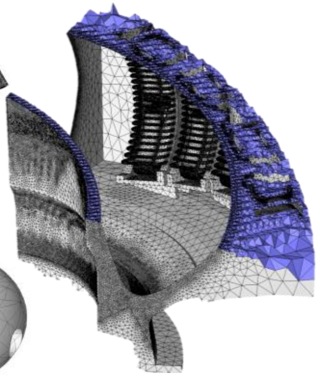
Applications with billions of elements: flip-chip (L), flow control (R)



Mesh adaptation for evolving features



Anisotropic adaptation for curved meshes



RF antenna and plasma surface in vessel.

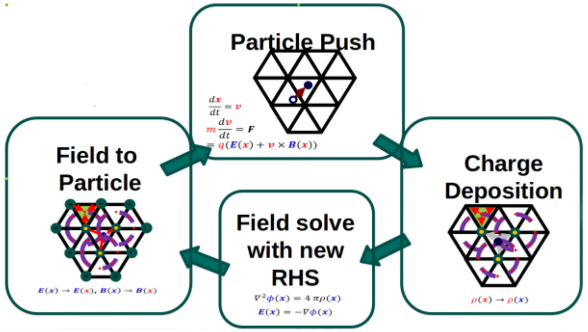
Source Code: github.com/SCOREC/core
Paper: www.scorec.rpi.edu/REPORTS/2014-9.pdf

PUMIPic Parallel Unstructured Mesh Infrastructure for Particle-in-Cell

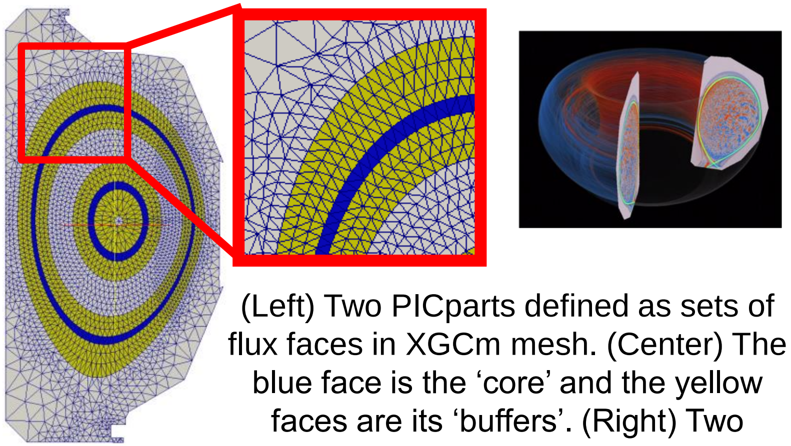
Parallel management of unstructured meshes with particles. Framework for GPU accelerated particle-in-cell applications using unstructured meshes.

- **Core functionality**
  - Unstructured mesh-based approach
    - Particles accessed through mesh
    - Particle search through mesh adjacencies
    - Effective coupling to PDE solvers
    - Partitioning using bounding flux surfaces, graph, or geometric methods
    - PICpart: owned elements (defined by partition) + copied elements from topologically or spatially neighboring processes
    - Stored on GPU using Omega_h library: *github.com/SNLComputation/omega_h*
  - Particles
    - Supports multiple species each with distinct combinations of 'Plain Old Data' per particle
    - Group particles by the mesh element that they are spatially located within
    - Stored on GPU using Sell-C-Sigma structure [Kreutzer 2014] that provides coalesced accesses for 'warp' sized blocks of work
  - Parallel kernel launch function abstracts underlying particle and mesh storage

- **Applications Supported**
  - GITRm: impurity transport
  - XGCm: core+edge fusion plasma physics
  - Weak scaling on up to 24,000 GPUs of Summit with 1.15 trillion particles running push, particle-to-mesh, and mesh-to-particle operations with an XGCm tokamak mesh and domain decomposition



Stages of a PIC application supported by PUMIPic



(Left) Two PICparts defined as sets of flux faces in XGCm mesh. (Center) The blue face is the 'core' and the yellow faces are its 'buffers'. (Right) Two poloidal planes in a toroidal domain.

Source Code: github.com/SCOREC/pumi-pic
Paper: www.scorec.rpi.edu/REPORTS/2019-2.pdf

PUMIPic Applications

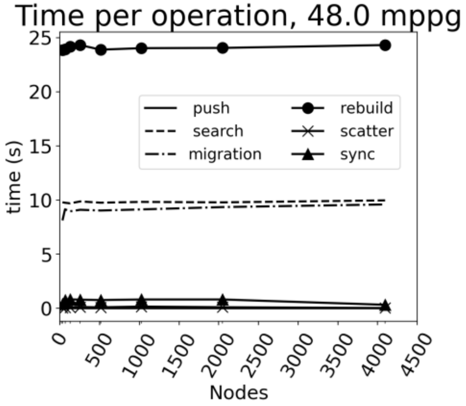
Unstructured mesh particle-in-cell fusion applications using PUMIPic. Supporting the analysis of tokamak plasma physics and impurity transport using extensions to the PUMIPic framework.

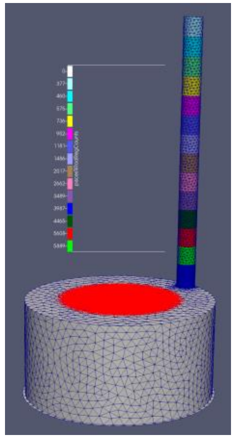
- **XGCm**
  - Core and edge fusion plasma physics with ions and kinetic electrons
  - Tokamak: 2D mesh partitioned into PICParts (see PUMIPic slide) based on bounding flux surfaces
  - A group of processes is assigned to a PICPart and 1/Pth of the torus in the toroidal direction – group size controls particle load on each GPU
  - Initial focus on performance and scaling with pseudo operations
  - Weak scaling on up to 24,000 GPUs of Summit with 1.15 trillion particles running push, particle-to-mesh, and mesh-to-particle operations
  - Current focus on implementing physically correct operations
- **GITRm**
  - Impurity transport
  - 3D meshes PICParts formed using graph based partitions
  - Tracking wall collisions and multiple species
  - Initial focus on verifying implementation of all physics model terms
  - Statistical and numerical verification complete
  - Current focus on performance and scalability

HBPS High-fidelity Boundary Plasma Simulation

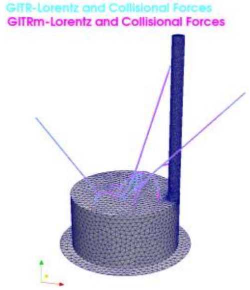
XGCm weak scaling on Summit

D3D, 2M elm. mesh, 192 PICParts/plane, 1 to 128 planes, 48M ptcls/GPU, 6 GPUs/node





Counts of impacting particles



Particle paths match

GITRm PISCES initial test case

Contact: Mark S. Shephard
shephard@rpi.edu

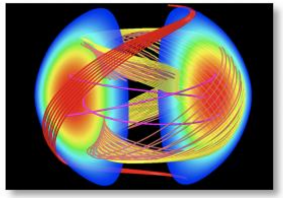
SUNDIALS

Suite of Nonlinear and Differential /Algebraic Equation Solvers

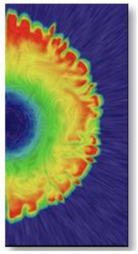
Adaptive time integrators for ODEs and DAEs and efficient nonlinear solvers
Used in a variety of applications. Freely available. Encapsulated solvers & parallelism.

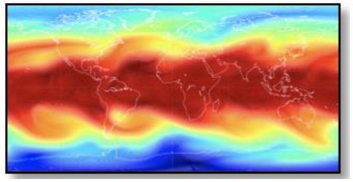
- **ODE integrators:**
  - CVODE: adaptive order and step BDF (stiff) & Adams (non-stiff) methods
  - ARKode: adaptive step implicit, explicit, IMEX, and multirate Runge-Kutta methods
- **DAE integrators**: IDA – adaptive order and step BDF integrators
- **Sensitivity Analysis:** CVODES and IDAS provide 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

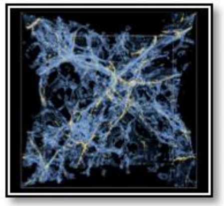
SUNDIALS is used by thousands worldwide in applications from research and industry


Magnetic Reconnection

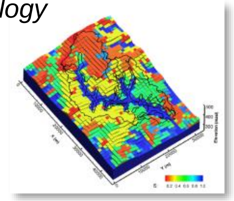

Dislocation Dynamics


Core Collapse Super-nova


Atmospheric Dynamics


Cosmology

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

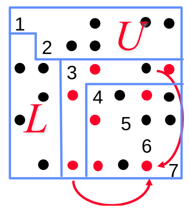

Subsurface Flow

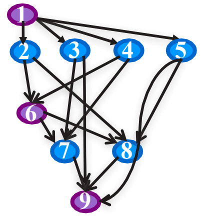
http://www.llnl.gov/casc/sundials

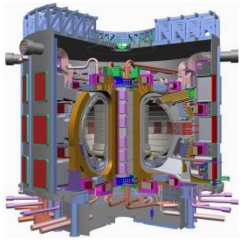
SuperLU

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

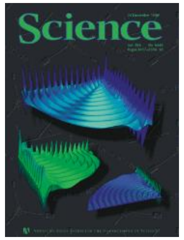
- **Capabilities**
  - Serial (thread-safe), shared-memory (SuperLU_MT, OpenMP or Pthreads), distributed-memory (SuperLU_DIST, hybrid MPI+ OpenM + CUDA).
    - Implemented in C, with Fortran interface
  - Sparse LU decomposition, triangular solution with multiple right-hand sides
  - Incomplete LU (ILU) 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.

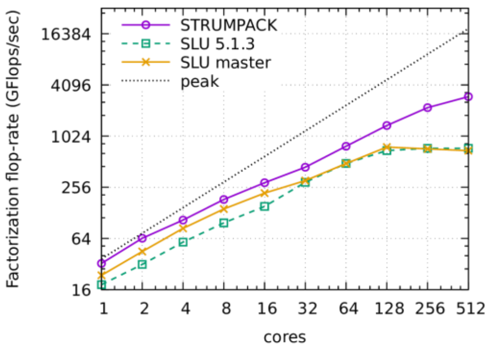
http://crd-legacy.lbl.gov/~xiaoye/SuperLU

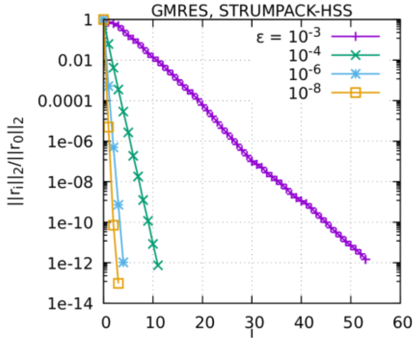
STRUMPACK

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

STRUMPACK
STRUctured Matrix PACKage





github.com/pghysels/STRUMPACK

EXASCALE COMPUTING PROJECT

Trilinos

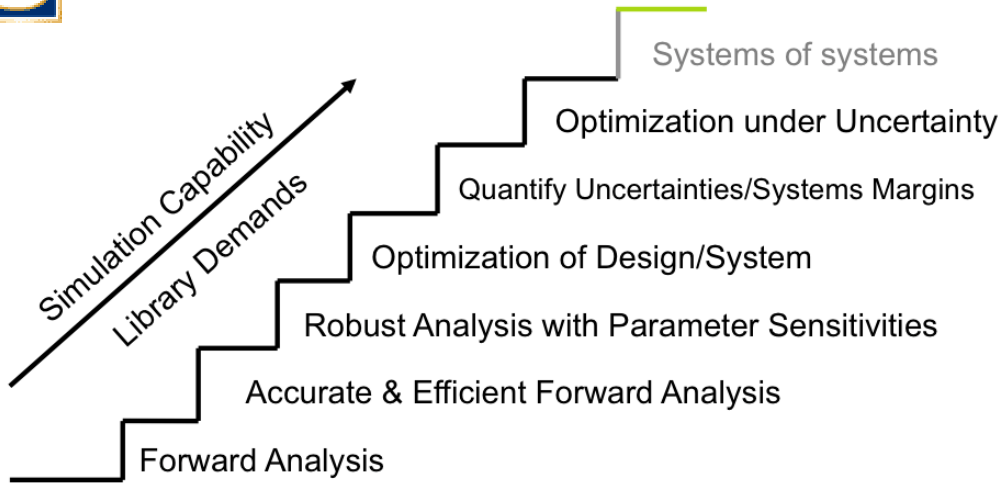
Sandia National Laboratories

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**
  - Scalable linear, nonlinear, eigen, transient, optimization, UQ solvers
  - Discretization, geometry, meshing
  - Load balancing
  - ***Performance Portability across multiple platforms (GPU, multicore) provided by Kokkos***
- **60+ packages**
  - Other distributions: Cray LIBSCI, Github repo
  - Thousands of users, worldwide distribution
  - Laptops to leadership systems: MPI, GPU, multicore

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.github.io/

xSDK FASTMATH TRILINOS

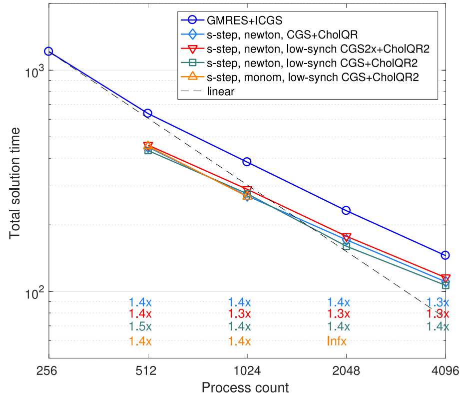
EXASCALE COMPUTING PROJECT

Trilinos/Belos

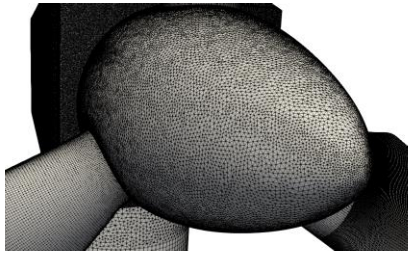
Iterative Krylov-based solvers. Templated C++ allows for generic scalar, ordinal, and compute node types.

- **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
  - Pipelined and s-step methods
  - Stable polynomial preconditioning
- **Performance portability via Kokkos (CPUs, NVIDIA/Intel/AMD GPUs, Phi)**
- **Ongoing research**
  - Communication avoiding methods

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.



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



Thomas et al., "High-fidelity simulation of wind- turbine incompressible flows", SISC, 2019.

Sandia National Laboratories

ASC

ECP EXASCALE COMPUTING PROJECT

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

ECP EXASCALE COMPUTING PROJECT

Trilinos/MueLu

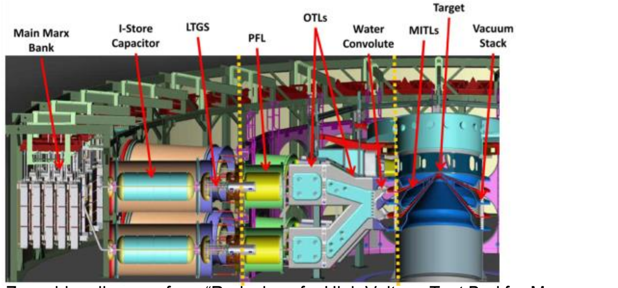
Structured and unstructured aggregation-based algebraic multigrid (AMG) preconditioners

- **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, ℓ_1 GS, polynomial, ILU, sparse direct
  - Load-balancing for good parallel performance
  - Structured coarsening, geometric multigrid
  - Setup and solve phases can run on GPUs.
  - Performance portability via Kokkos (CPUs, NVIDIA/Intel/AMD GPUs, Xeon Phi)

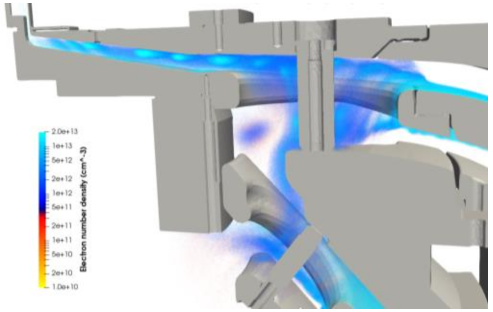
- **Research Areas**
  - AMG for multiphysics
  - Multigrid for coupled structured/unstructured meshes
  - Algorithm selection via machine learning

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.



Z machine diagram, from "Redesign of a High Voltage Test Bed for Marxes on Z", W.M. White et al., 2018.

AMG preconditioning for H(curl) systems is key enabling technology in Z machine simulations for determining power from Marx banks to Target.


Plasma density in Z machine Target simulation, courtesy of D. Sirajuddin (SNL).

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

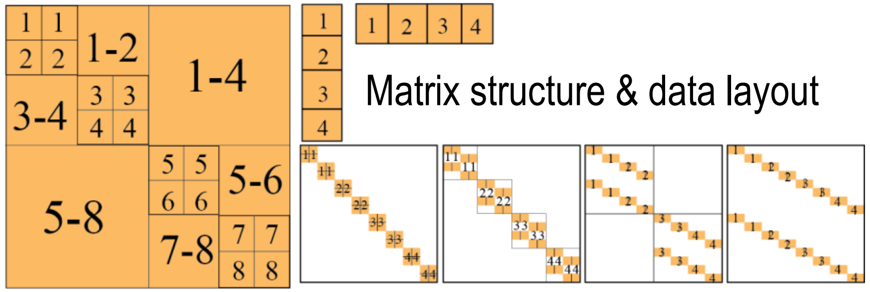
Gallery of highlights

- Overview of HPC numerical software packages
- 1 slide per package, emphasizing key capabilities, highlights, and where to go for more info
  - Listed first (alphabetically)
    - Packages featured in ATPESC 2020 lectures and hands-on lessons
  - Listed next (alphabetically)
    - Additional highlighted packages (<u>not a comprehensive list</u>)

ButterflyPACK

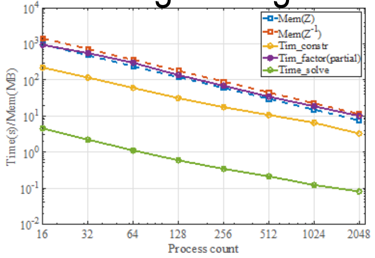
Fast direct solvers. Low-rank and butterfly compression. Distributed-memory parallel. Particularly for highly-oscillatory wave equations.

- **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

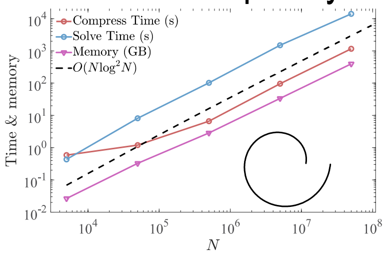


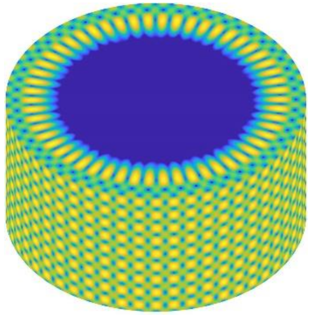
Matrix structure & data layout

Strong scaling



Solver complexity





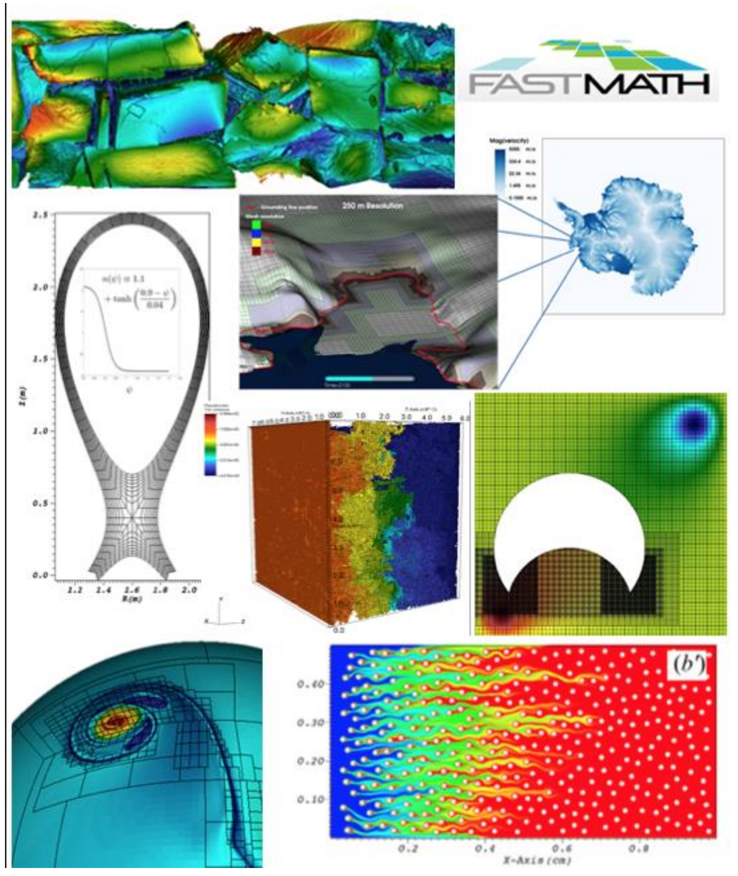
Accelerator cavity

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

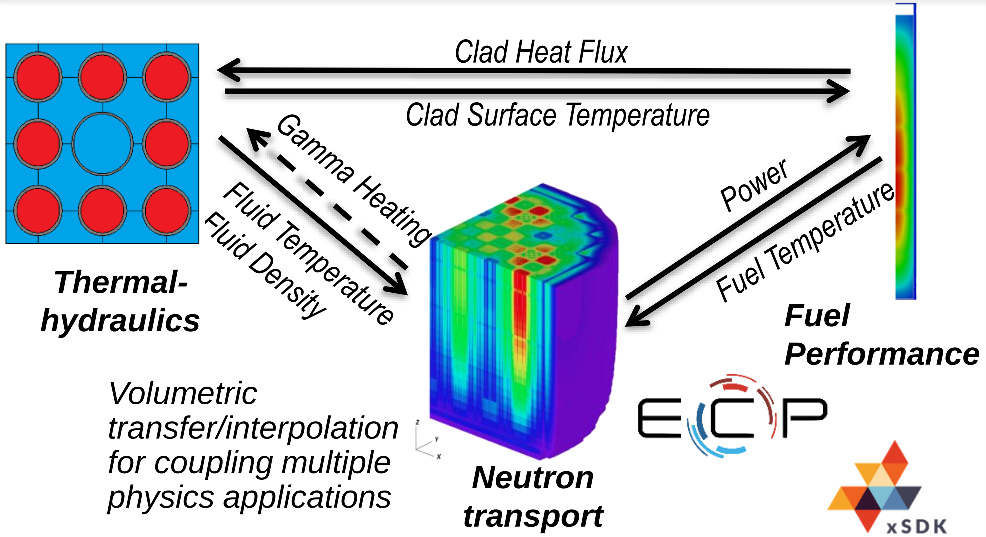


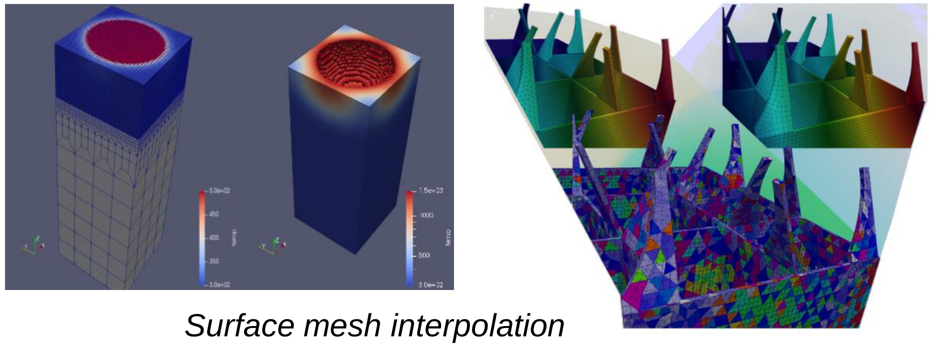
http://Chombo.lbl.gov

DataTransferKit

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

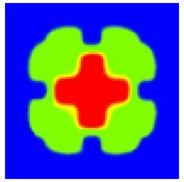




Surface mesh interpolation

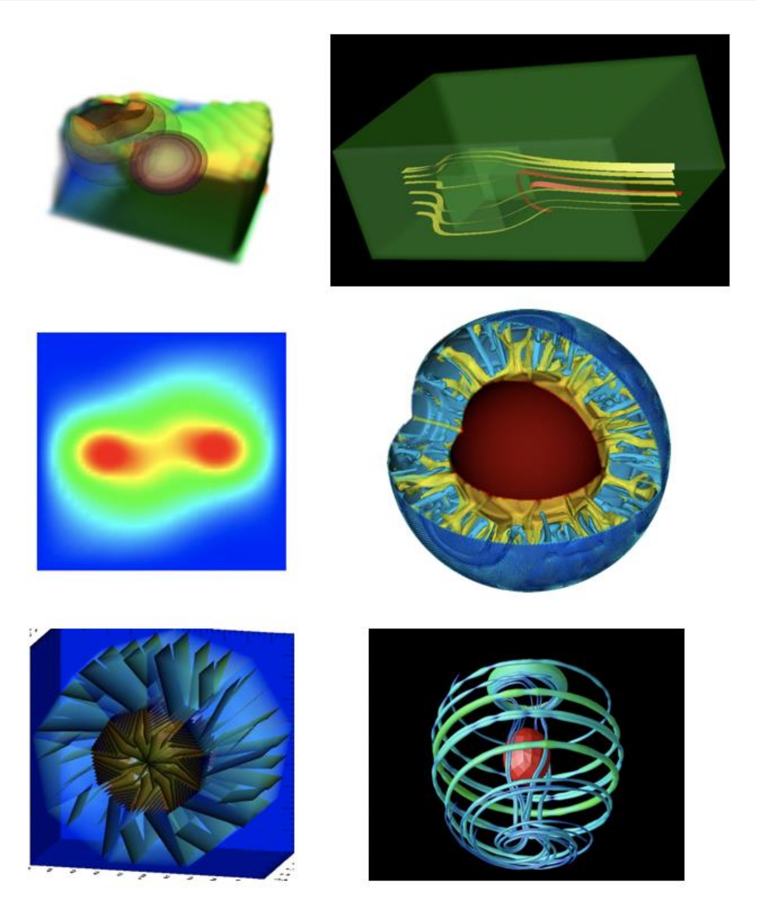
https://github.com/ORNL-CEES/DataTransferKit

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

EXASCALE COMPUTING PROJECT

libEnsemble

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

libEnsemble aims for:

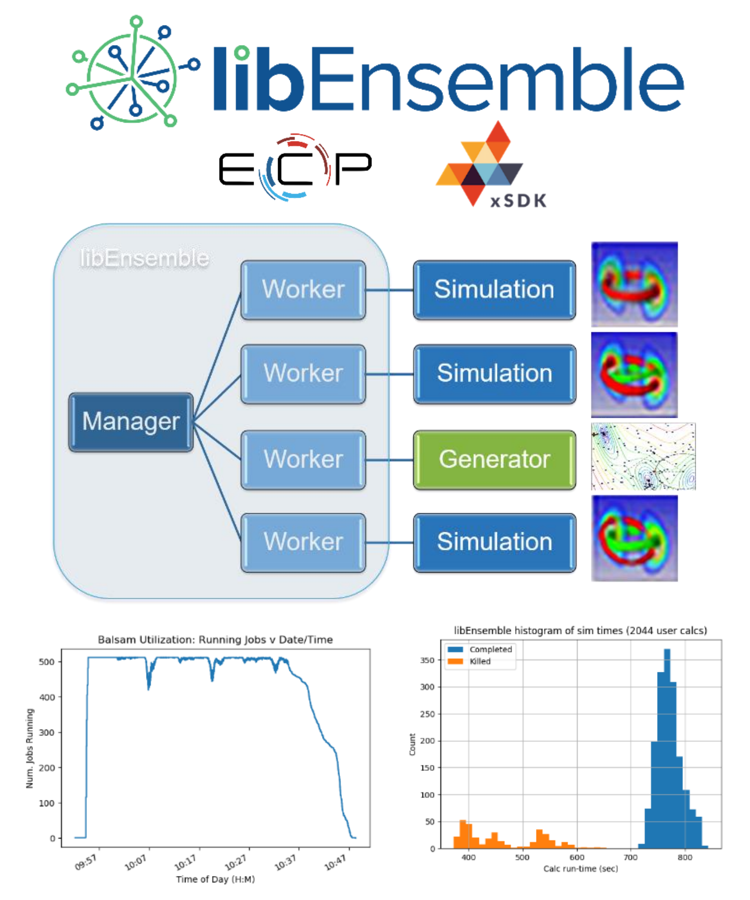
- Extreme scaling
- Resilience and fault tolerance
- Monitoring and killing tasks and recovering resources
- Portability and flexibility

libEnsemble features:

- Communications using MPI, multiprocessing, or TCP
- Support for calculations using parallel resources, including user-provided executables
- Executor auto-detects system resources and launches user executables
- Support on Summit (ORNL), Theta (ALCF), Cori (NERSC), Bridges (PSC)

Dynamic ensembles:

- Workers are allocated simulations or generate input for simulations
- One use case: an optimization method generates parameters to be evaluated by a computationally expensive simulation
- Example interfaces with PETSc, SciPy, and NLopt solvers are available



https://libensemble.readthedocs.io

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**

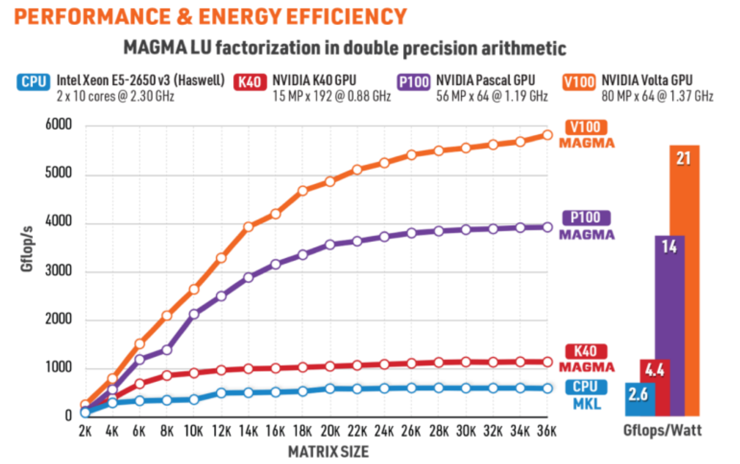
MAGMA SPARSE

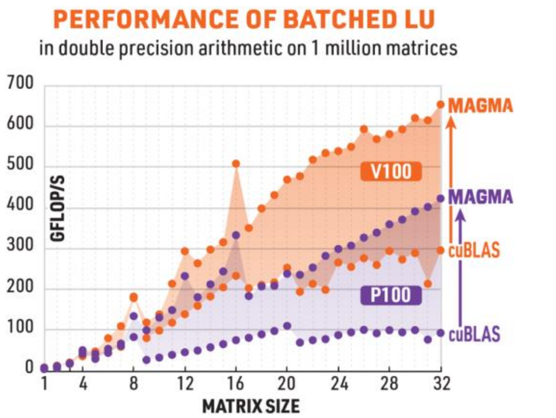
ROUTINES	BiCG, BiCGSTAB, Block-Asynchronous Jacobi, CG, CGS, GMRES, IDR, Iterative refinement, LOBPCG, LSQR, QMR, TFQMR
PRECONDITIONERS	ILU / IC, Jacobi, ParILU, ParILUT, Block Jacobi, ISAI
KERNELS	SpMV, SpMM
DATA FORMATS	CSR, ELL, SELL-P, CSR5, HYB

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





http://icl.utk.edu/magma

EXASCALE COMPUTING PROJECT

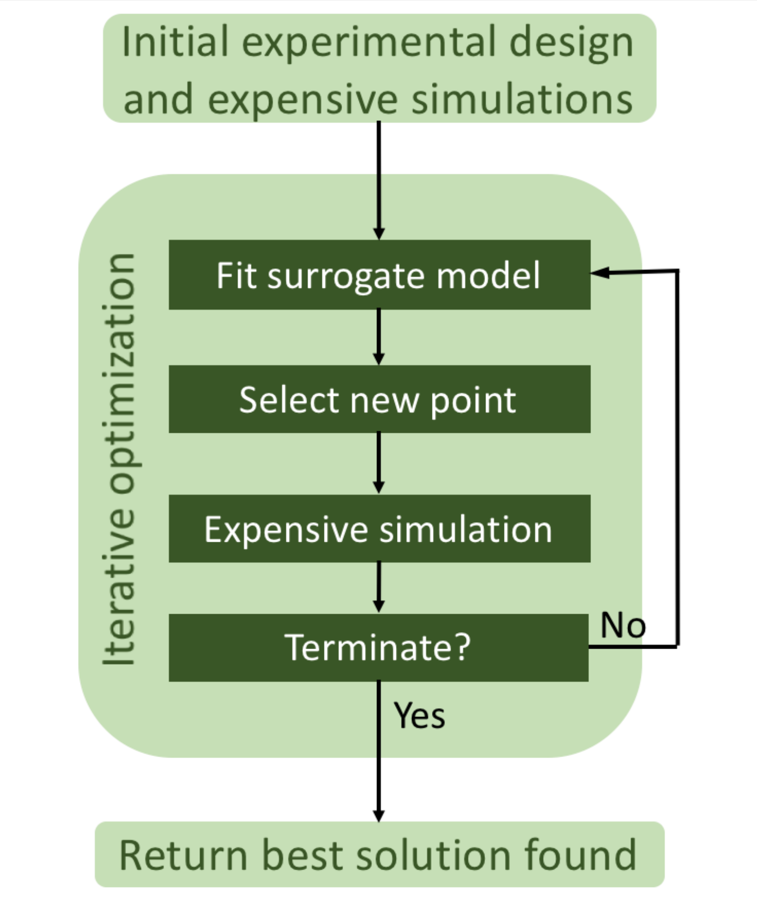
MATSuMoTo

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

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

required flexibility ↑

TRILINOS PETSc

full interoperability

PHIST (builtin)

optimized (Fortran, MPI, OpenMP)
row-major block vectors
CRS format

GHOST

C, MPI, OpenMP, SIMD, CUDA
SELL-C-sigma format

↓ hardware awareness

Funded by the DFG project ESSEX

SPPEXA

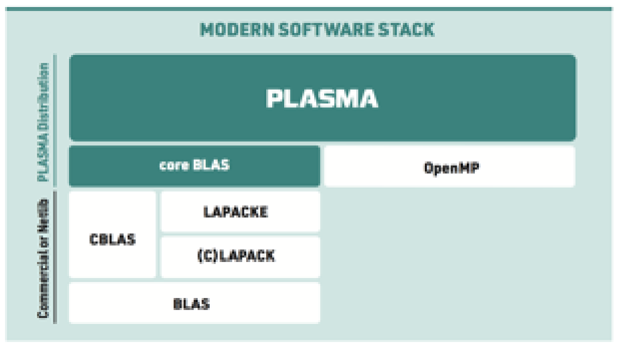
https://bitbucket.org/essex/phist

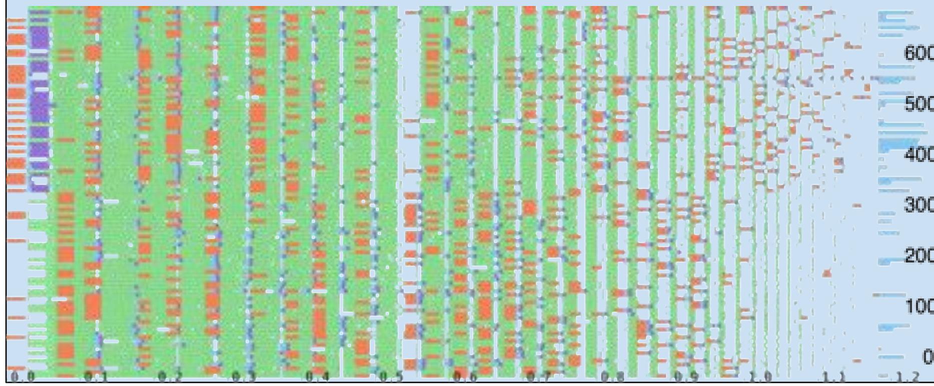
PLASMA

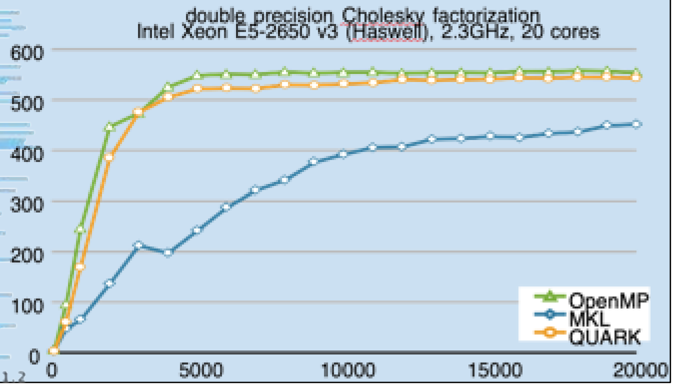
THE UNIVERSITY OF TENNESSEE KNOXVILLE

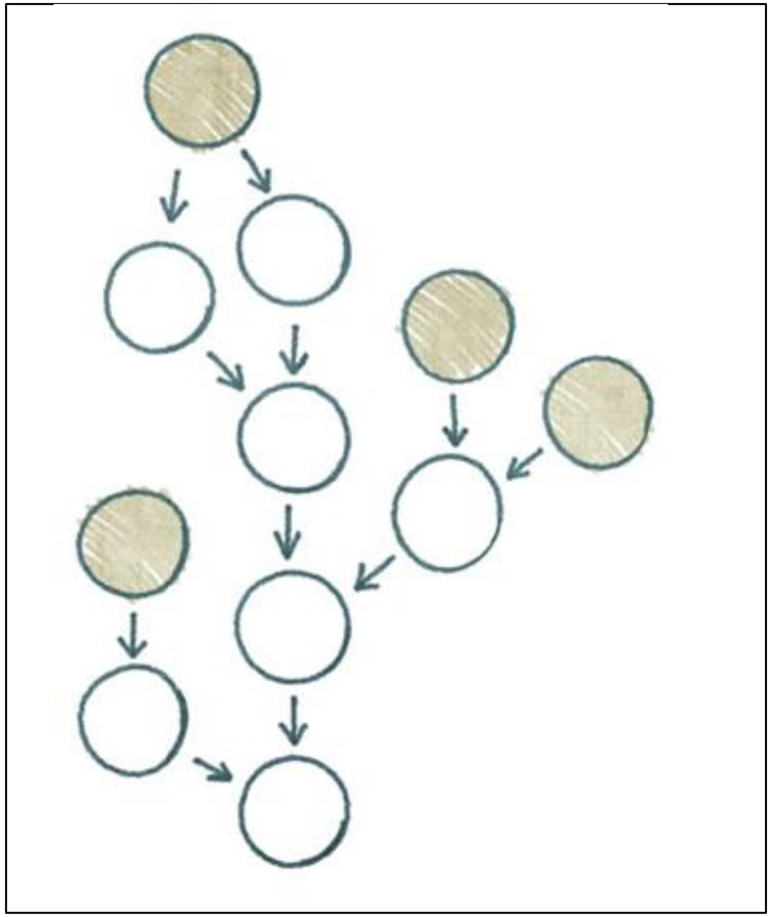
Linear algebra solvers and spectral decompositions for multicore processors.
Portable and scalable dense solvers for large core counts.

- **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

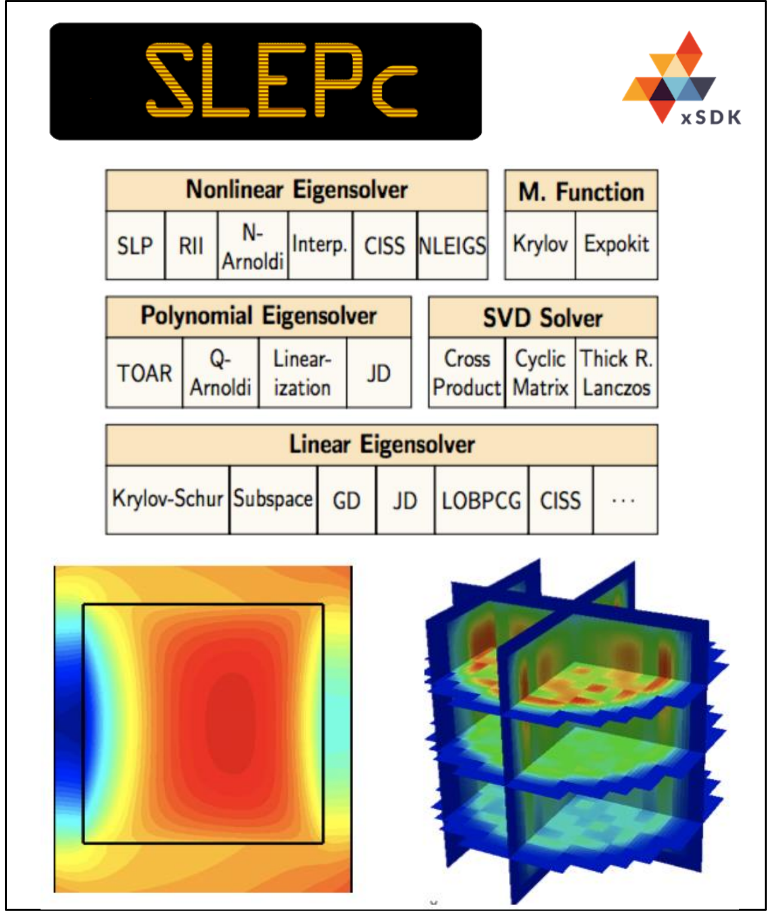
EXASCALE COMPUTING PROJECT

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



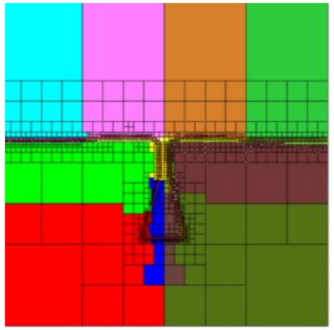
http://slepc.upv.es

EXASCALE COMPUTING PROJECT

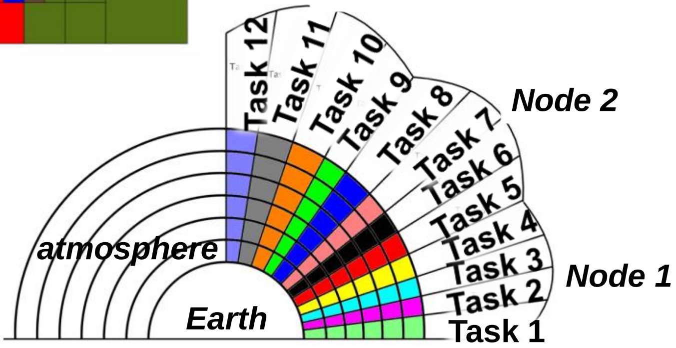
Zoltan/Zoltan2

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)
  - MPI+X geometric partitioning using Kokkos for GPU and multicore
- **Architecture-aware MPI task placement reduces application communication time**
  - Places interdependent MPI tasks on “nearby” nodes in network
  - 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

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

HandsOn Lessons

- Hand-coded heat equation intro
- Structured meshing & discretization
- Unstructured meshing & discretization
- Krylov solvers & preconditioners
- Sparse direct solvers
- Nonlinear solvers
- Time integration
- Numerical optimization

ATPESC 2020 Hands On Lessons

Meshing and Discretization with AMReX	A Block Structured Adaptive Mesh Refinement Framework
Hand Coded Heat	Why use numerical packages...
Krylov Solvers and Algebraic Multigrid with hypre	Demonstrate utility of multigrid

And more …

Github pages site:

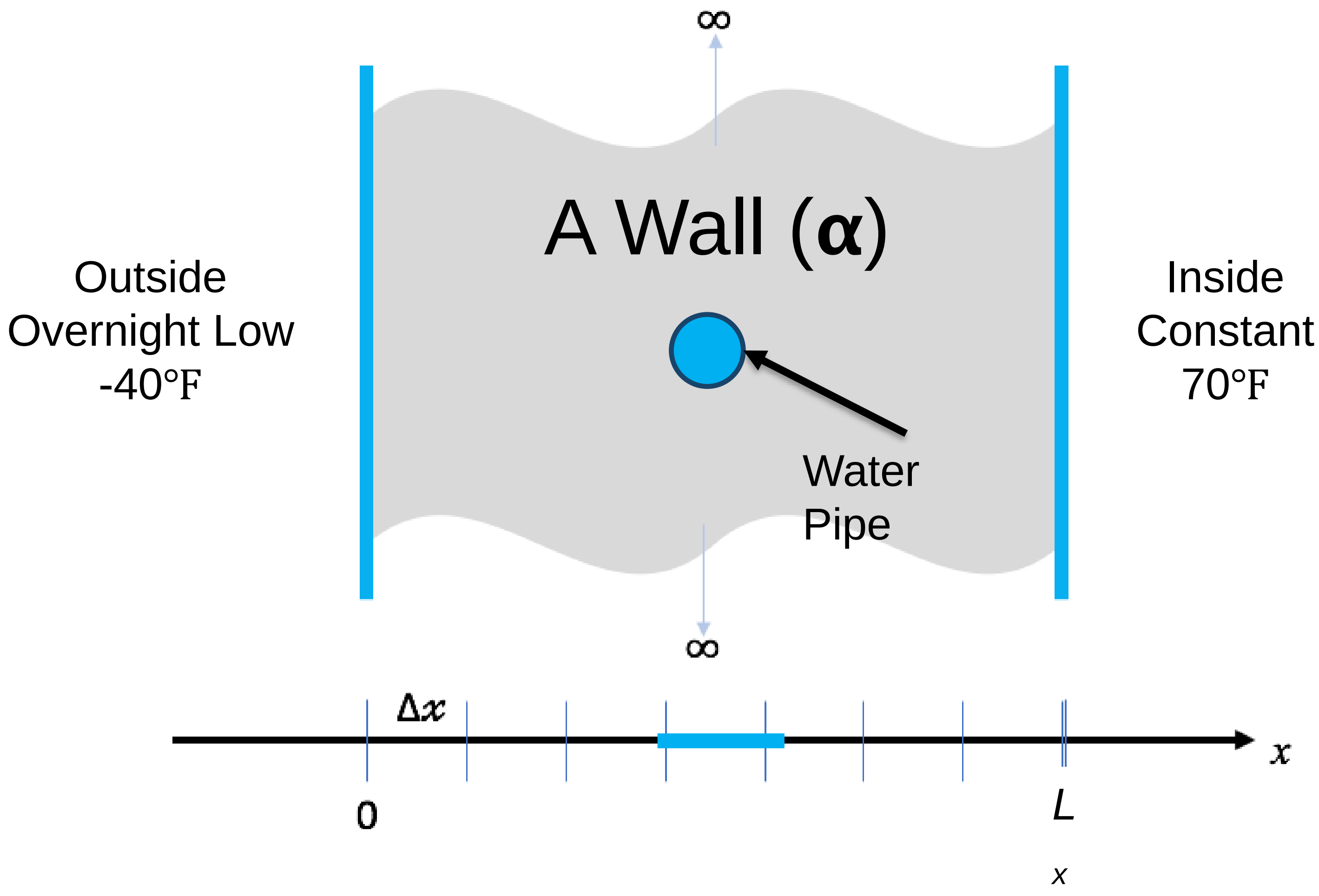
https://xsdk-project.github.io/MathPackagesTraining2020/lessons/

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: $u(x,0)=U(x)$

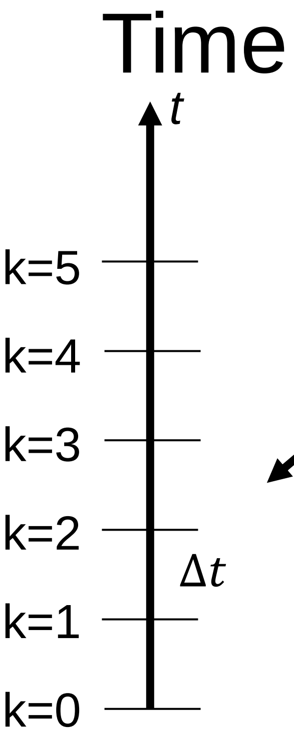
We seek a numerical software solution for $u(x,t)$ (all space & time)

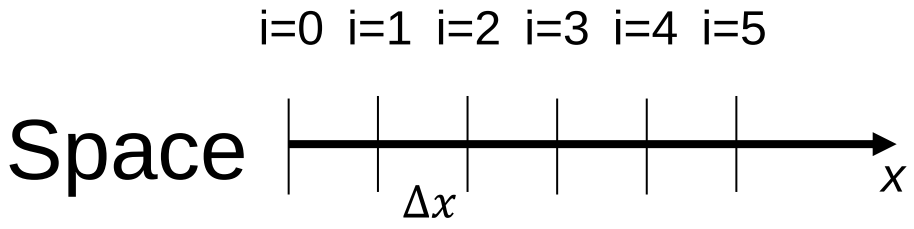
Discretize: Continuous → Discrete

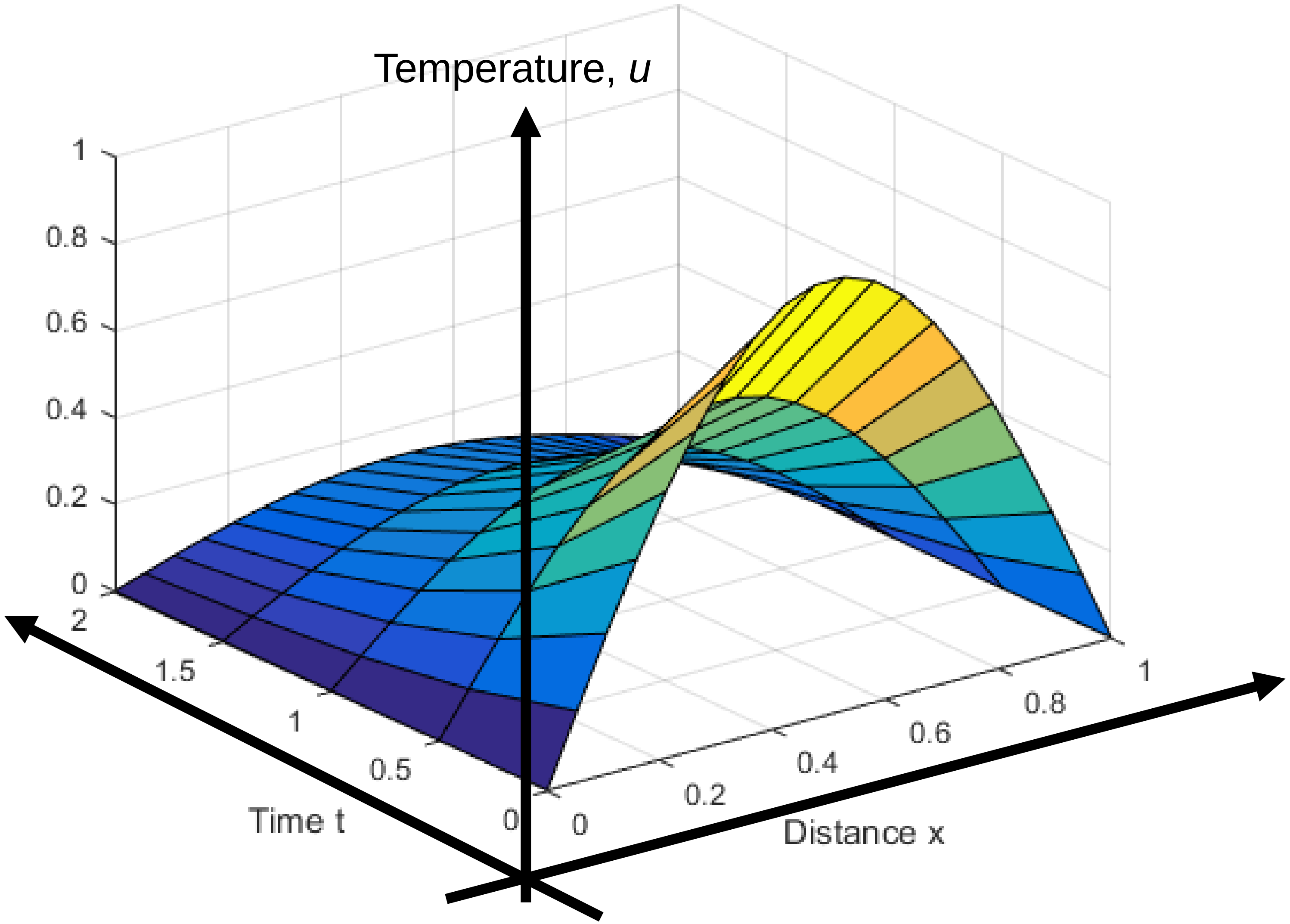
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

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

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






Temperature, u
1
0.8
0.6
0.4
0.2
0
2
1.5
1
0.5
0
Time t
0
0.2
0.4
0.6
0.8
1
Distance 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.
  - For more sophisticated cases, need a full-fledged solver.
- Known to be **unstable** for $r > \frac{1}{2}$

EXASCALE COMPUTING PROJECT

Exercise #1 (3 mins)

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

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

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

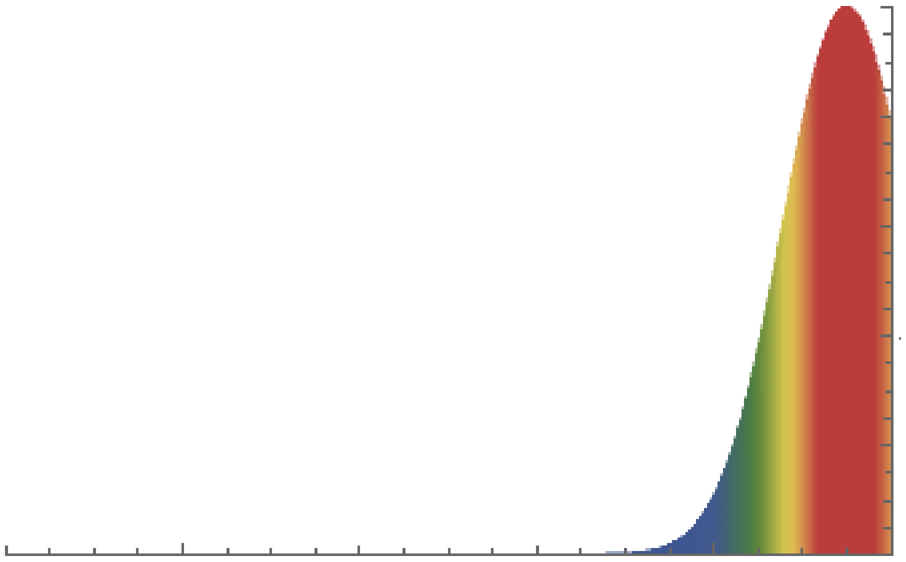
```
bool                                // true if valid, false if not
update_solution_ftcs(
    int n,                          // number of values
    Double *uk1,                    // new values: u(i) i=0…n-1 @ t=k+1
    Double const *uk0,              // last values: u(i) i=0…n-1 @ t=k
    Double 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>
```

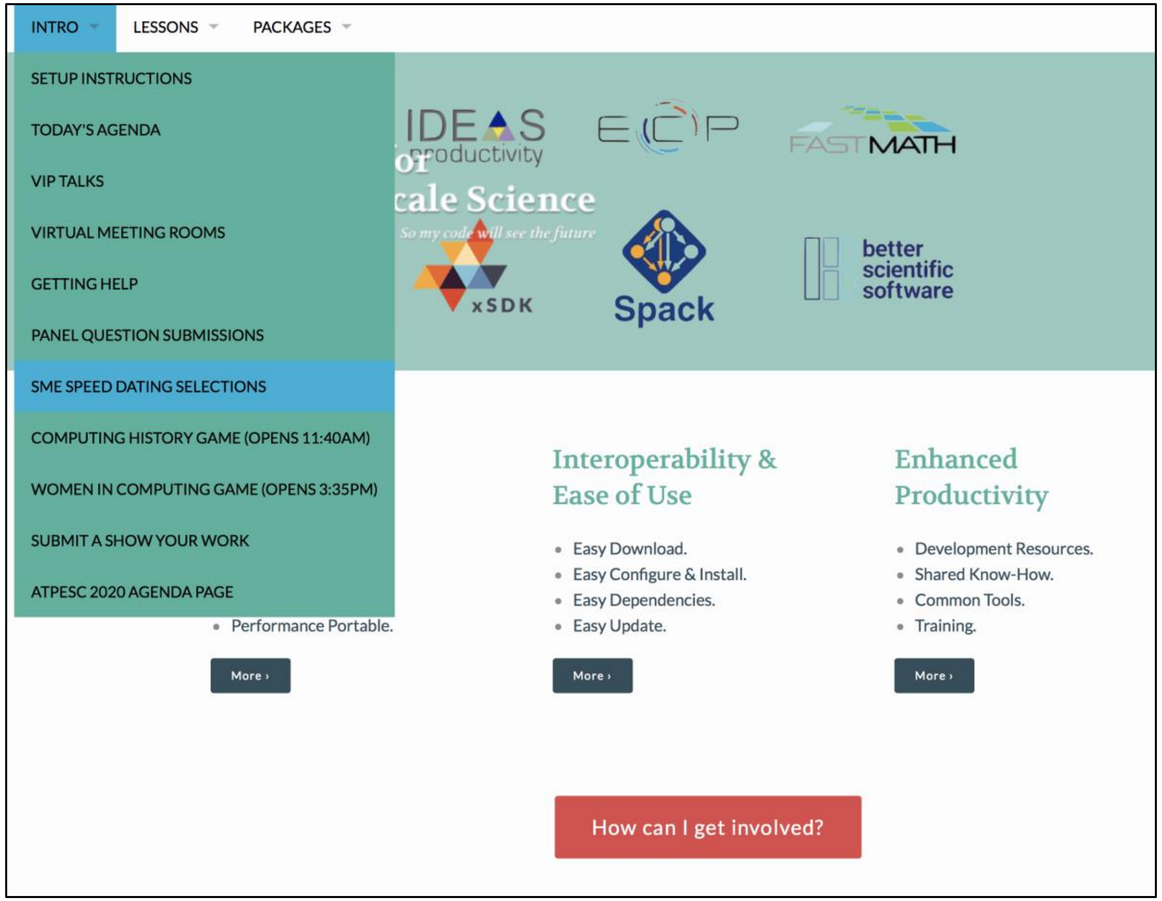
What if our problem was to find the optimum wall width?

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

Next steps

- **Attend parallel sessions**
  - **10:30 CDT and 11:45 CDT**
  - **Access Zoom rooms via**
  - https://xsdk-project.github.io/MathPackagesTraining2020/agenda
- **During breaks and lunch**
  - **Submit questions for panelists** (optional)
  - **Sign up for discussions with numerical software developers** (optional)
    - Your email address
    - Select 1st, 2nd, and 3rd priorities
    - Brief description of interests
    - Complete by 3:30 pm CDT
- **Panel session: Main Room @ 1:45 pm CDT**



EXASCALE COMPUTING PROJECT

Today’s agenda

https://xsdk-project.github.io/MathPackagesTraining2020/agenda

Mix-n-Match topics
to your interests
See Synopses from Agenda

Unstructured Discretization (with MFEM/PUMI)

Slides

Unstructured meshes can yield required levels of accuracy using fewer degrees of freedom at the cost of more complex parallel data structures and algorithms. To support the ability of application code developers to take advantage of unstructured meshes, FASTMath develops core tools to support the development of unstructured mesh simulation capabilities. This lecture will first introduce the highly extendible MFEM high order finite element solver library and then overview the PUMI unstructured mesh tools developed to support mesh adaptation, load balancing and PIC calculations.

Direct Solvers (with SuperLU/Strumpack)

Slides

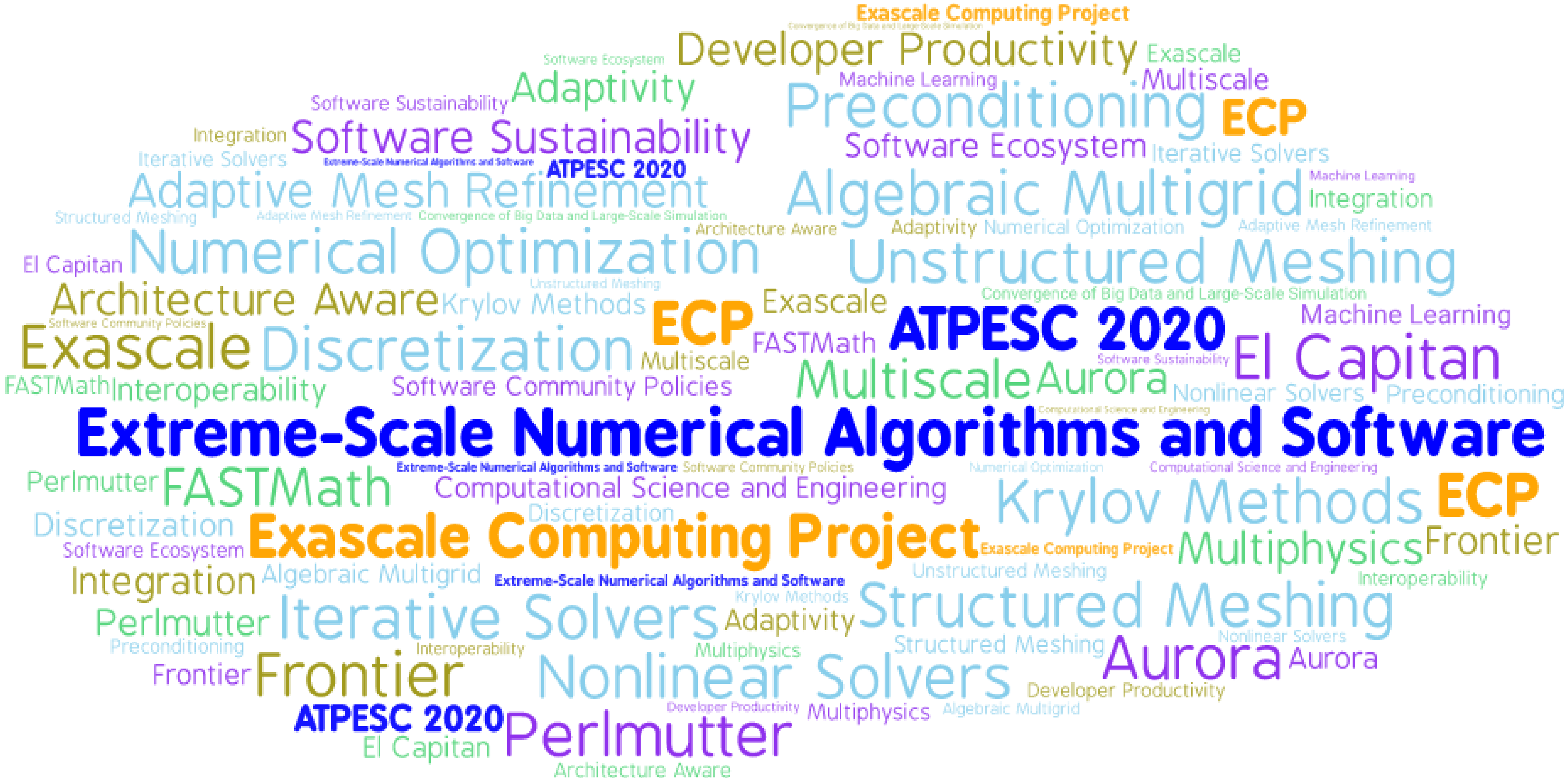
Direct Solvers are presented in three different time slots, each with a slightly different emphasis...

- Session 1 (10:30am-11:30am):
  - Sparse direct solvers (both SuperLU and Strumpack), 30 minutes (Sherry)
  - Low rank approximation techniques in Strumpack, 15 minutes (Pieter)
  - SuperLU hands-on demo, 15 minutes (Sherry)
- Session 2 (11:45am-12:45pm):
  - Sparse direct solvers (both SuperLU and Strumpack), 30 minutes (Sherry)
  - Low rank approximation techniques in Strumpack, 15 minutes (Pieter)
  - Strumpack hands-on demo, 15 minutes (Pieter)
- Session 4 (3:40pm - 4:30pm):
  - Sparse direct solvers (both SuperLU and Strumpack), 30 minutes (Sherry)
  - Low rank approximation techniques in Strumpack, 15 minutes (Pieter)
  - Q&A. (Sherry, Pieter)

CDT Start	Mins	Topic	Speaker(s)	Virtual Venue
09:30	55	Intro. to Numerical Libraries	Lois Curfman McInnes Mark Miller	Main-Room
10:25	5	Telecon Transition		
10:30	60	**Parallel Session One**		
		Structured Discretization (with AMReX)	Ann Almgren Don Willcox	Frontier
		Unstructured Discretization (with MFEM/PUMI)	Aaron Fisher Mark Shephard	Aurora
		Iterative Solvers & Preconditioners (with MueLu)	Jonathan Hu Christian Glusa	Perlmutter
		Direct Solvers (with SuperLU/Strumpack)	Sherry Li Pieter Ghysels	El-Capitan

EXASCALE COMPUTING PROJECT

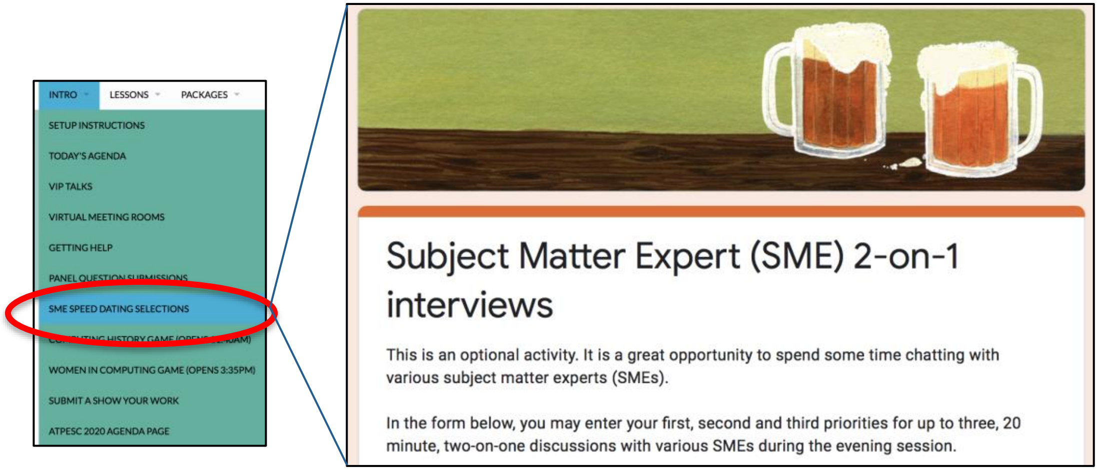
Addendum … ATPESC afternoon sessions



Next steps: Extreme-Scale Numerical Algorithms and Software

https://xsdk-project.github.io/MathPackagesTraining2020/agenda

- **Panel session: Main Room @ 1:45 pm CDT**
- **During breaks and lunch**
  - **Sign up for discussions with numerical software developers** (optional)
    - Your email address
    - Select 1st, 2nd, and 3rd priorities
    - Brief description of interests
    - Complete by 3:30 pm CDT



The ATPESC Team 2020 on Zoom

Row 1:
- Dan Reynolds
- Lois Curfman McInnes
- Don Willcox
- Ann Almgren
- Satish Balay

Row 2:
- Pieter Ghysels
- Christian Glusa
- Mark Miller
- Aaron Fisher
- Sherry Li

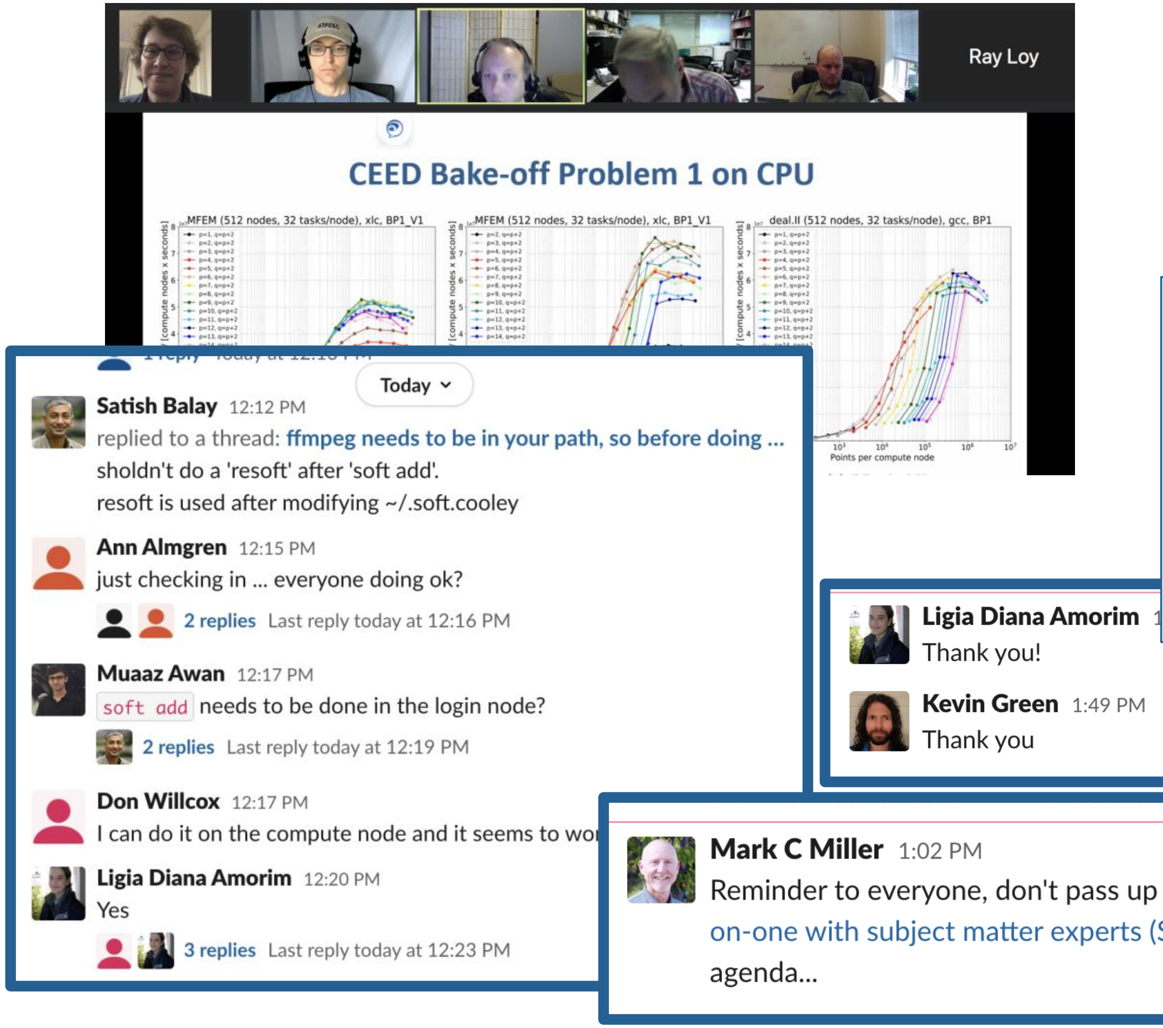
Row 3:
- Sara Osborn
- David Gardner
- Mark Shephard
- Ulrike Yang
- Alp Dener

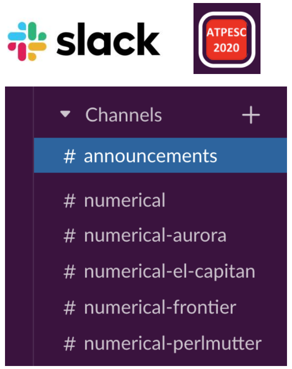
Row 4:
- Richard Mills
- Jonathan Hu

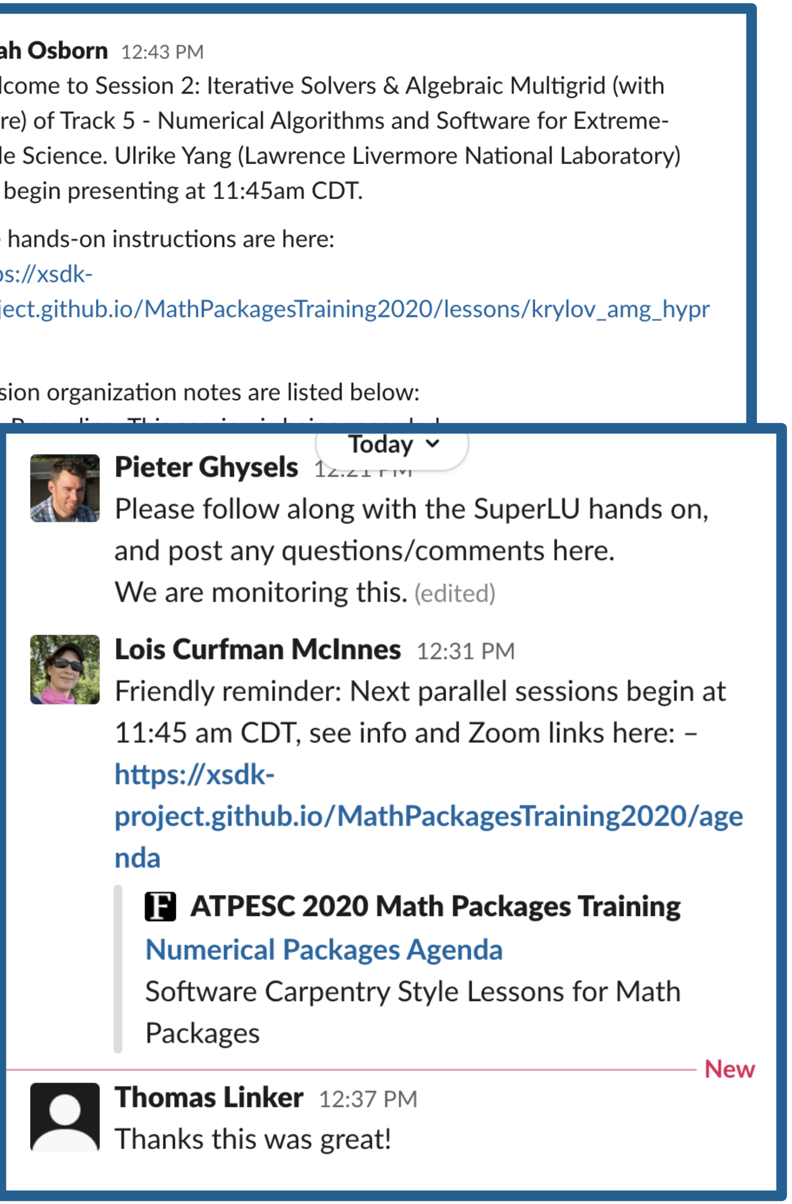
Not shown:
- Cameron Smith
- Carol Woodward

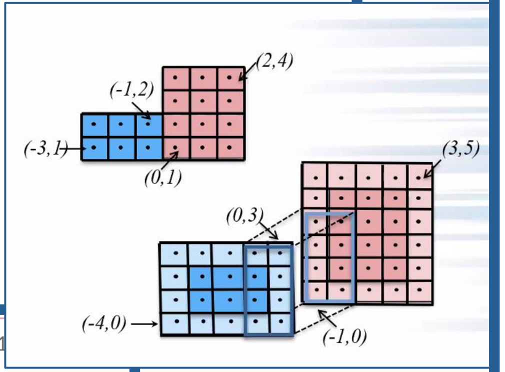
EXASCALE COMPUTING PROJECT

Parallel morning sessions
A few highlights









Panel: Contributing to the Numerical Software Community

1:45 pm CDT, Main Room in Zoom

- **Q&A Session:** ATPESC learners ask questions about working with numerical packages and the community of numerical package developers
  – Questions in **#numerical** slack channel (and via Google form earlier today)

- Panelists

Jonathan Hu, SNL

Sherry Li, LBL

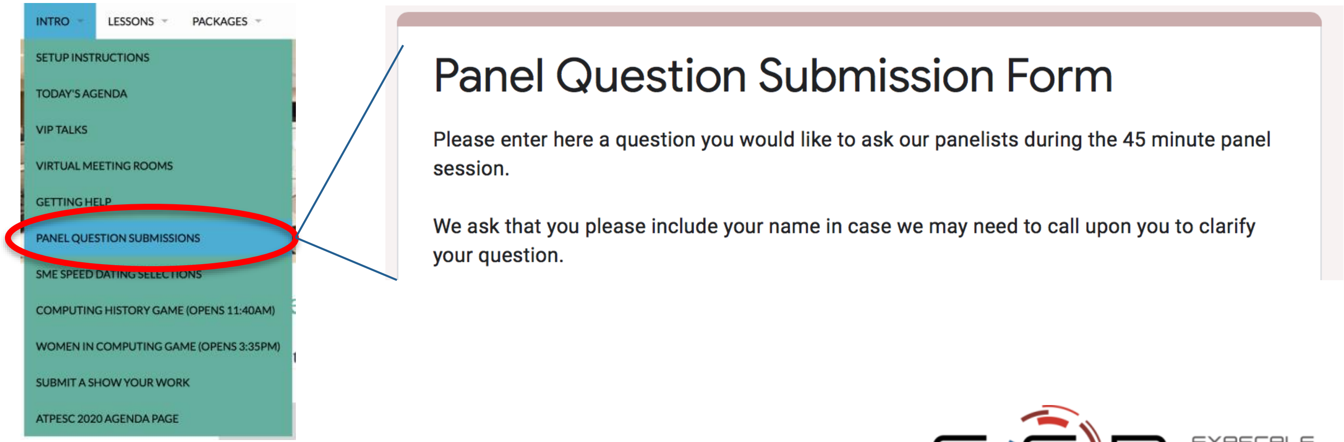
Richard Mills, ANL

Cameron Smith, RPI

Ulrike Yang, LLNL

- Moderator

Mark Miller, LLNL



EXASCALE COMPUTING PROJECT

Afternoon agenda

https://xsdk-project.github.io/MathPackagesTraining2020/agenda

Nonlinear Solvers (with PETSc)

Slides

We will begin with a quick overview of iterative solvers for nonlinear systems, and then take a deeper look into Newton-Krylov methods and how to use them via the PETSc Scalable Nonlinear Equation Solvers (SNES) component. We will do some hands-on exploration with a classic computational fluid dynamics benchmark, the lid-driven cavity pro[blem. We will] end by looking at how nonlinear composition and preconditioning can be us[ed ...] wide array of nonlinear solvers from the algorithmic building blocks in SNE[S ...] strate how these techniques can handle particularly difficult nonlinearities.

Time Integration (with SUNDIALS)

Slides

In this lecture we will discuss the role and impact of high order, adaptive, and flexible time integration libraries in solution accuracy and computational efficiency of large-scale simulations. Due to the wide variety of backgrounds among ATPESC participants, we will briefly discuss

- the location of time integrators in the HPC landscape, and their reliance on scalable nonlinear and linear solver libraries,
- the different categories of time integration methods (explicit/implicit/IMEX),
- ...properties of time integration methods (order of accuracy, linear ...
- ...daptivity for improving accuracy and efficiency,
- ...ime integration packages.

...tely half of the time period in lecture, followed by hands-on exer-...ity, accuracy, temporal adaptivity, and the role of problem-specific ...e hands-on exercises focus on time-dependent PDEs, and use the ...rary for time integration, along with the AMReX library for spatial

Optimization (with TAO)

Slides

This lecture will provide an introduction to numerical optimization with a theoretical focus on simulation-based problems. We will introduce the user interfaces for the Toolkit for Advanced Optimization (TAO) package within the PETSc library and exercise several gradient-based algorithms on scalable synthetic test problems. We will observe and discuss the relative convergence of different classes of algorithms and sensitivity analysis methods in a parallel environment.

02:35	50	**Parallel Session Three**		
		Nonlinear Solvers (with PETSc)	Richard Tran Mills	Frontier
		Optimization (with TAO)	Alp Dener	Aurora
		Time Integration (with SUNDIALS)	Dan Reynolds	Perlmutter
		Iterative Solvers & Preconditioners (with MueLu)	Jonathan Hu Christian Glusa	El-Capitan
03:25	15	**Break** *SME Selections Due*		
03:40	50	**Parallel Session Four**		
		Nonlinear Solvers (with PETSc)	Richard Tran Mills	Frontier
		Optimization (with TAO)	Alp Dener	Aurora
		Time Integration (with SUNDIALS)	Dan Reynolds	Perlmutter
		Direct Solvers (with SuperLU/Strumpack)	Sherry Li Pieter Ghysels	El-Capitan
04:30	5	Telecon Transition		
04:35	25	Working with Numerical Packages in Practice	Ann Almgren	Main-Room
		END OF OFFICIAL AGENDA		
05:00	15	**Break**		
05:15	80	SME Speed Dating OPTIONAL ACTIVITY		

EXASCALE COMPUTING PROJECT

Thank you to all ATPESC staff

Special thanks to Ray Loy and Yasaman Ghadar

For their outstanding work in running the 2-week ATPESC program

And thank you to all ATPESC attendees for engaging questions and discussions!

From Ligia Diana Amorim to Everyone:
And the great tutorials like the ones here at ATPESC also help a lot!

From Mark C. Miller to Everyone
Ditto

zoom

From Neil Mehta to Everyone:
Thank you!

From Valeria Barra to Everyone:
I also want to add that I really appreciated the website put together for today's tutorials. Thanks for the nice work!

From Ligia Diana Amorim to Everyone:
Thank you all for the great sessions today! Yes, indeed, the website was great to follow all the hands on

From Zhi Jackie Yao to Everyone:
Thank you!

From Juan Colmenares to Everyone:
yes

From Kevin Green to Everyone:
Thanks for all of the hands on material today! Was a lot of info to take in a once, but I'm looking forward to slowly digesting it all over the next evenings :)

From Mukund Raj to Everyone:
Thanks all for the presentation and great materials!

Some Zoom and Slack chats during sessions on extreme-scale numerical algorithms and software

Severin Diederichs 3:53 PM
How well does the adaptive time step selection work with multi step methods like a 5th order Adams-Bashforths integrator?
1 reply Today at 3:57 PM

Komal Kumari 4:04 PM
Unstable?
👍 1

Zac Johnston 4:05 PM
about 21?

Juan Diego Colmenares Fernandez 4:05
yeah around 22

Aaron Fisher 3:23 PM
There are Python bindings for MFEM (edited)

Kevin Green 3:23 PM
Doesn't scipy interface to SuperLU?

Sherry Li 3:24 PM
@Kevin Green Indeed! I think they wrote the Python interface, it's not in superlu release.
2 replies Last reply today at 3:27 PM

Kevin Green 3:27 PM
Thanks panelists
👍 9

Bruce A. Perry 5:34 PM
Thanks for the presentation!

Daniel R. Reynolds 5:34 PM
You're welcome!

Hugo Brunie 5:34 PM
Thanks, great tutorial

Philippe Blain 5:35 PM
Thanks Dan, very nice presentation! (edited)

slack

Channels
announcements
numerical
numerical-aurora
numerical-el-capitan
numerical-frontier
numerical-perlmutter

ECP EXASCALE COMPUTING PROJECT