

# Introduction to Numerical Software

Presented to  
**ATPESC 2023 Participants**

**Ulrike Meier Yang**  
Lawrence Livermore National Laboratory

Date 08/08/2023

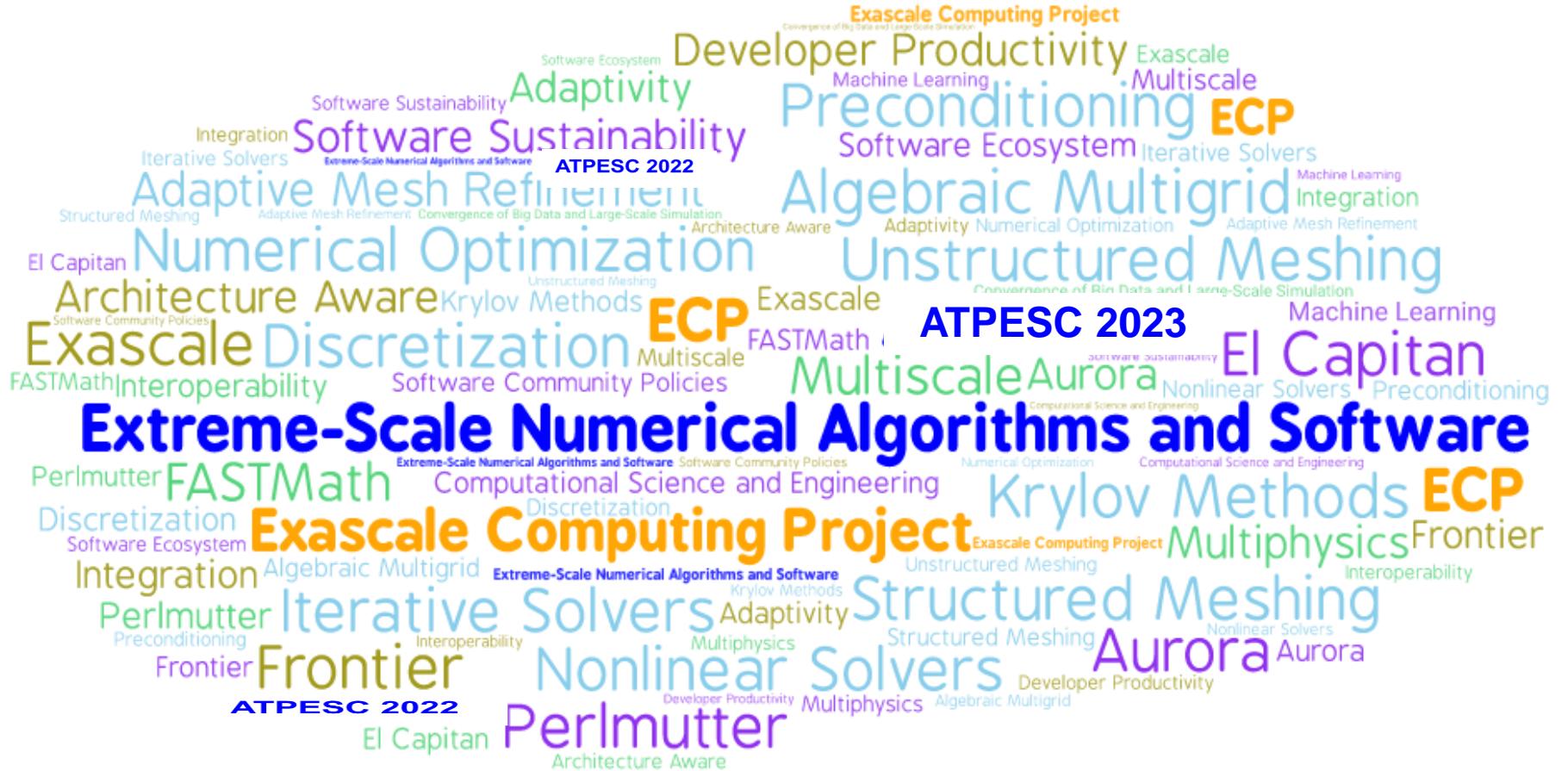


**ATPESC Numerical Software Track**



# Outline

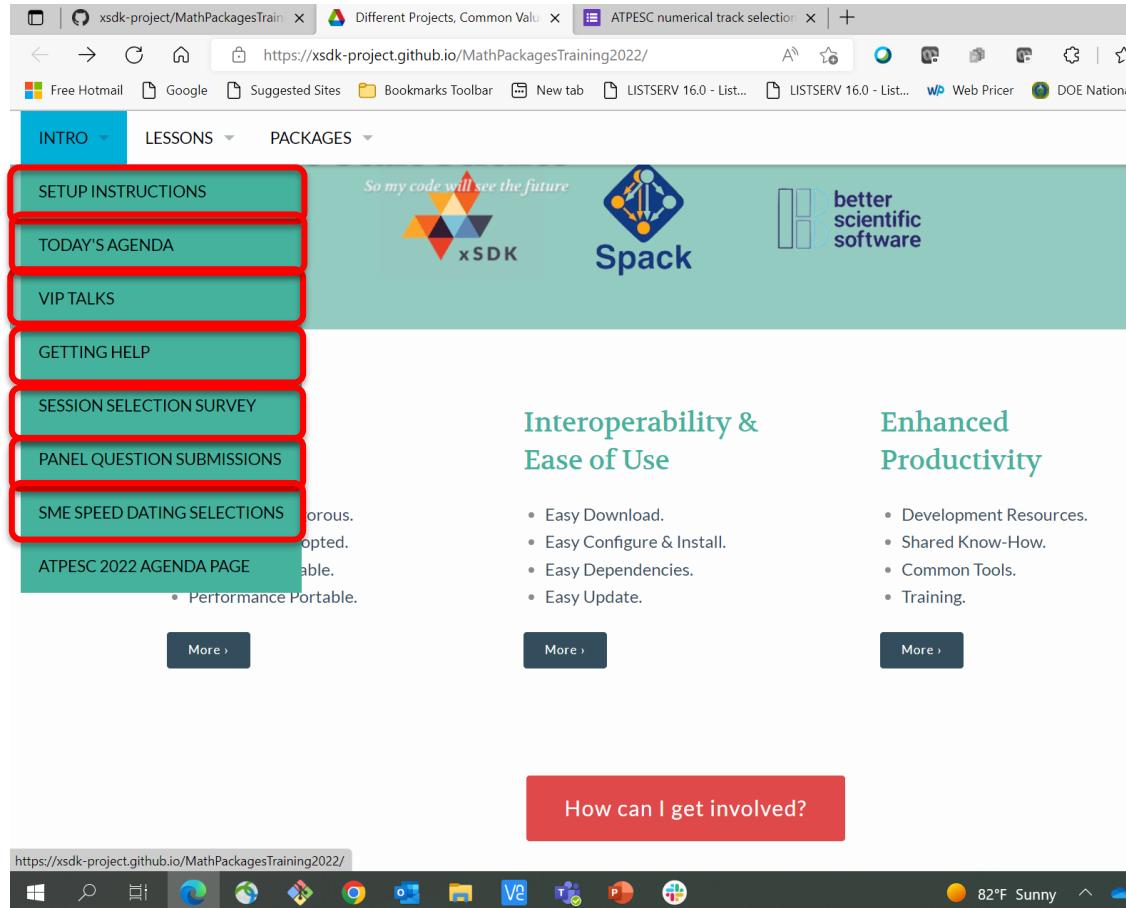
- Logistics for the day
- Intro to numerical algorithms and software for extreme-scale science



# 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-2022/#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/MathPackagesTraining2022/agenda/>

# <https://xsdk-project.github.io/MathPackagesTraining2023/>



- Setup instructions
- Today's agenda
- VIP talks
- Getting help
- Session Selection Survey
- Panel question submission
- SME speed dating selections

# Agenda

<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15		Break, Subject Matter Expert (SME) Selections, Panel Questions
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30		Lunch, SME Selections, Panel Questions
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15		Break, SME Selections, <b>Panel Questions Due</b>
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:30		Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software
5:30 – 6:30		Unstructured Time: <b>SME Selections Due</b> , Informal Discussion, Continue Hands-on
6:30 – 7:30		Dinner
7:30 – 9:30		Optional Activity: SME Speed-dating

# Choose which lecture you want to attend!

- Access: <https://forms.gle/axrawtNsTgbjDTJP8>

<https://xsdk-project.github.io/MathPackagesTraining2022/>

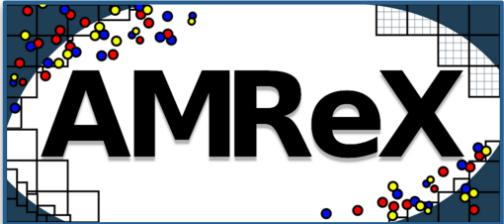
The screenshot shows a web browser window with three tabs: "xsdk-project/MathPackagesTrain", "Different Projects, Common Value", and "ATPESC numerical track selection". The main content area has a teal sidebar on the left with links: INTRO, LESSONS, PACKAGES, SETUP INSTRUCTIONS, TODAY'S AGENDA, VIP TALKS, GETTING HELP, and SESSION SELECTION SURVEY (which is highlighted with a red box). The main content area features logos for XSDK and Spack. Below the logos, two sections are shown: "Interoperability & Ease of Use" and "Enhanced Productivity", each with a bulleted list of benefits. At the bottom is a red button labeled "How can I get involved?".

The screenshot shows a survey form titled "ATPESC numerical track selections form". It asks participants to denote sessions they plan to attend. The first section is "Parallel Session One \*", with three radio button options: "Structured Meshes (with AMReX)", "Unstructured Meshes (with MFEM/PUMI)", and "Undecided (happy to be placed in either)". The second section is "Parallel Session Two \*", with two radio button options: "Krylov Solvers & Multigrid Preconditioning (with Belos and MueLu)" and "Direct Solvers (with SuperLU/STRUMPACK)". A "Switch account" link and a "Required" indicator are also present.

# Agenda

<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	<b>Structured Discretization (AMReX) – Ann Almgren, Andrew Myers</b>	<b>Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith</b>
10:45 – 11:15		Break
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30		Lunch
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15		Break
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:30	Wrap-up (Ann Almgren)/ Panel: Extreme-Scale Numerical Algorithms and Software	
6:30 – 7:30		Dinner
7:30 – 8:30		SME Speed Dating



**Block-structured adaptive mesh refinement framework.** Scalable support for hierarchical mesh and particle data, with embedded boundaries.

## ▪ Capabilities

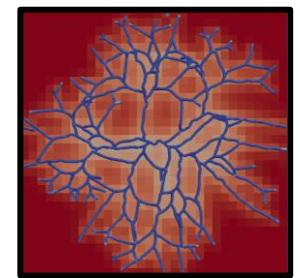
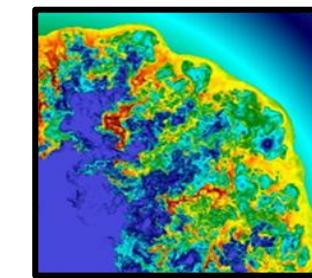
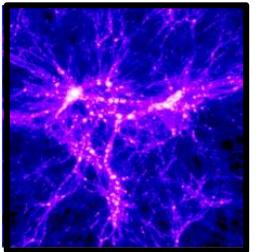
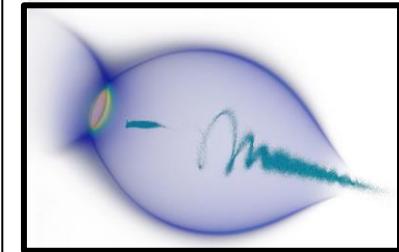
- Support for PDEs on a hierarchical adaptive mesh with particles and embedded boundary representations of complex geometry
- Support for multiple modes of time integration
- Support for explicit and implicit single-level and multilevel mesh operations, multilevel synchronization, particle, particle-mesh and particle-particle operations
- Hierarchical parallelism –
  - hybrid MPI + OpenMP with logical tiling on multicore architectures
  - hybrid MPI + GPU support for hybrid CPU/GPU systems (NVIDIA CUDA, AMD HIP, Intel SYCL)
- 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

## ▪ Open source software

- Used for diverse apps, including accelerator modeling, astrophysics, combustion, cosmology, multiphase flow, phase field modeling, atmospheric modeling and more
- Source code and development hosted on github with rigorous testing framework
- Extensive documentation, examples and tutorials



## Examples of AMReX applications

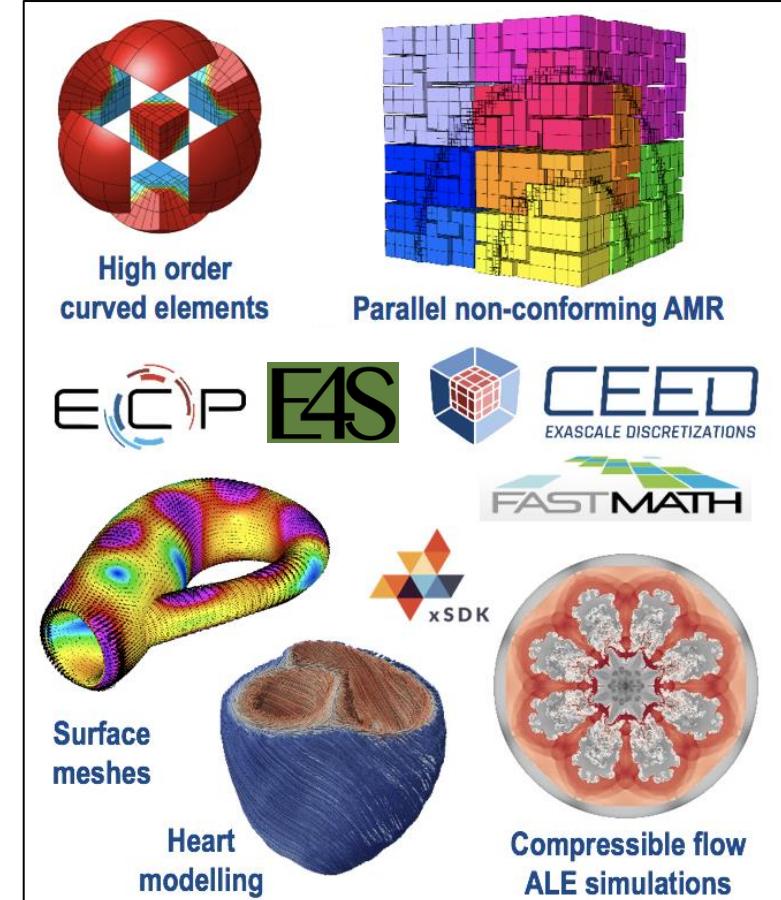


<https://www.github.com/AMReX-Codes/amrex>



**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.

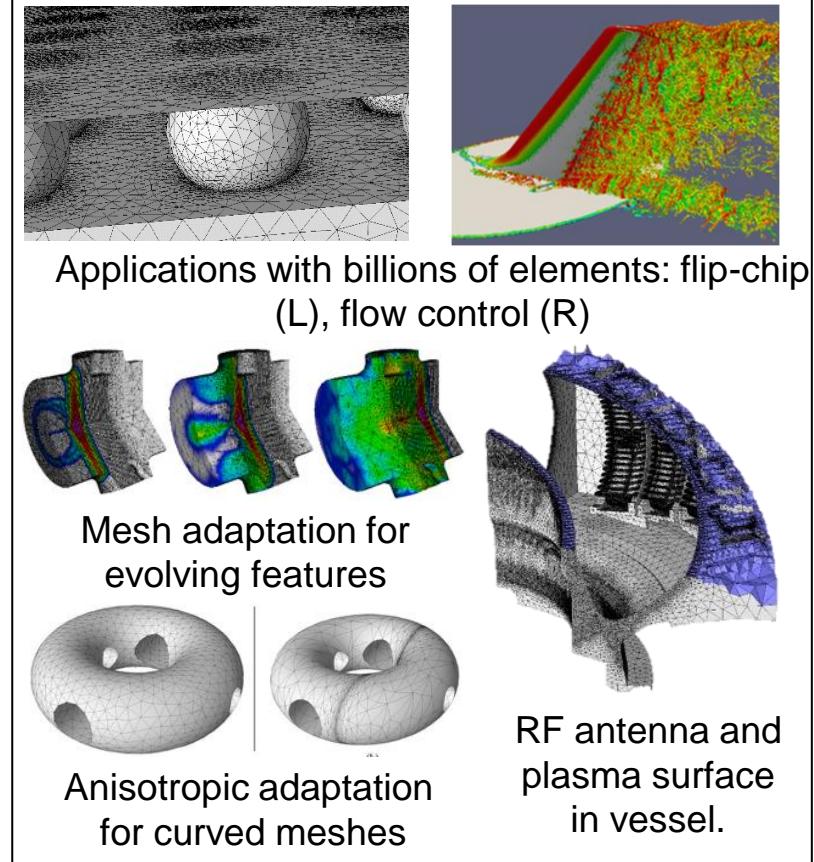


<https://mfem.org>

# 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**
  - xSDK package; installs with Slack
  - 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



**Source Code:** [github.com/SCOREC/core](https://github.com/SCOREC/core)  
**User Guide:** [scorec.rpi.edu/pumi/PUMI.pdf](http://scorec.rpi.edu/pumi/PUMI.pdf)  
**Paper:** [scorec.rpi.edu/REPORTS/2014-9.pdf](http://scorec.rpi.edu/REPORTS/2014-9.pdf)

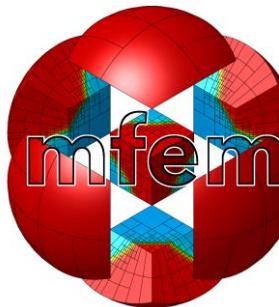
# Room Choice

- Please raise your hand if you want to attend

- Structured meshing



- Unstructured meshing



# Agenda

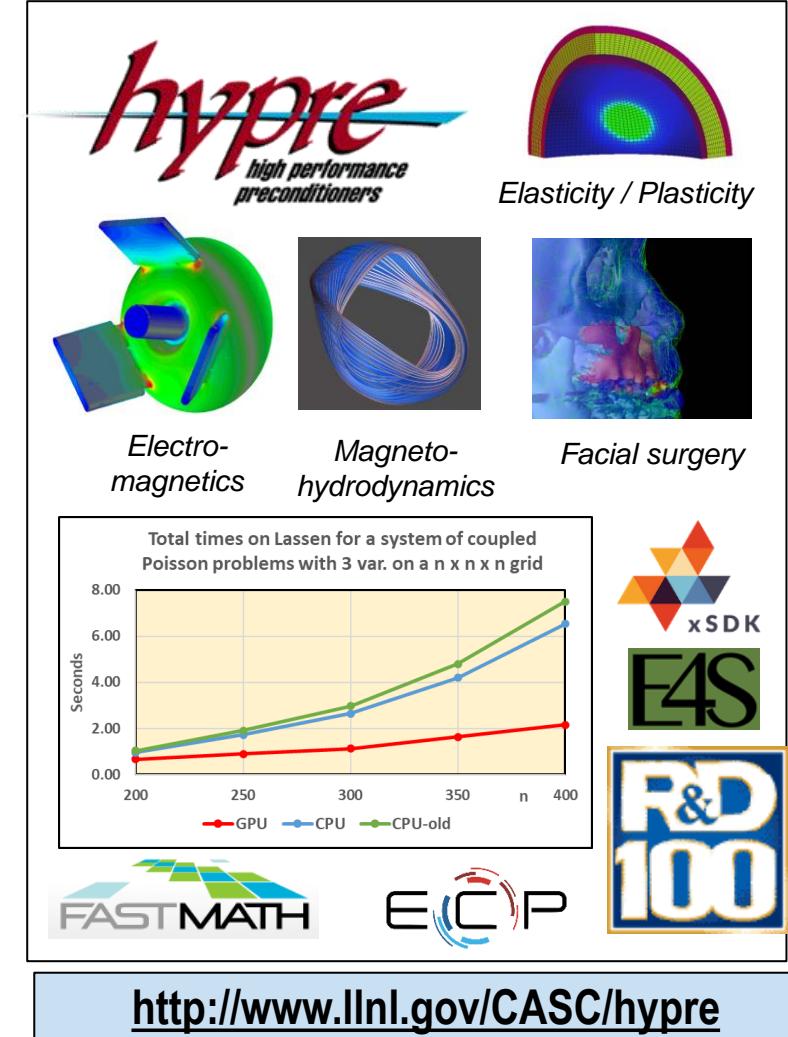
<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15		Break
11:15 – 12:30	<b>Iterative Solvers &amp; Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang</b>	<b>Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels</b>
12:30 – 1:30		Lunch
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15		Break
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:30	Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software	
6:30 – 7:30		Dinner
7:30 – 8:30		SME Speed Dating



**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 efficient (scalable) linear solvers through effective data storage schemes
- **Scalable preconditioners and solvers**
  - Structured and unstructured algebraic multigrid solvers
  - Maxwell solvers, H-div solvers
  - Multigrid solvers for nonsymmetric systems: pAIR
  - Multigrid reduction (MGR) for systems of PDEs
  - Matrix-free Krylov solvers
  - ILU and FSAI preconditioners
- **Exascale early systems GPU-readiness**
  - Nvidia GPU (CUDA), AMD GPU (HIP), Intel GPU (SYCL)
- **Open-source software**
  - Used worldwide in a vast range of applications
  - Can be used through PETSc and Trilinos
  - Available on github: <https://www.github.com/hypre-space/hypre>



# SuperLU



**Supernodal Sparse LU Direct Solver.** Flexible, user-friendly interfaces.  
Examples show various use scenarios. Testing code for unit-test. BSD license.

- **Capabilities**

- Serial (thread-safe), shared-memory (SuperLU\_MT, OpenMP or Pthreads), distributed-memory (SuperLU\_DIST, hybrid MPI+ OpenM + CUDA/HIP).
  - Written in C, with Fortran interface
- Sparse LU decomposition (can be nonsymmetric sparsity pattern), triangular solution with multiple right-hand sides
- Incomplete LU (ILUTP) preconditioner in serial SuperLU
- Sparsity-preserving ordering: minimum degree or graph partitioning applied to  $A^T A$  or  $A^T + A$
- User-controllable pivoting: partial pivoting, threshold pivoting, static pivoting
- Condition number estimation, iterative refinement, componentwise error bounds

- **Exascale early systems GPU-readiness**

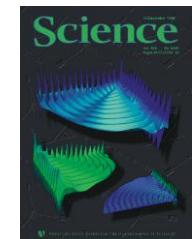
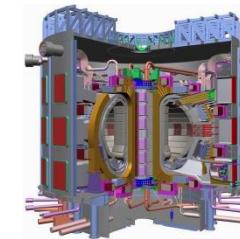
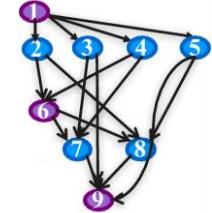
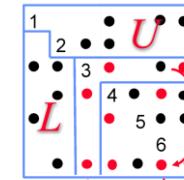
- Available: Nvidia GPU (CUDA), AMD GPU (HIP)
- In progress: Intel GPU (SYCL)

- **Parallel Scalability**

- Factorization strong scales to 32,000 cores (IPDPS'18, JPDC'19)
- Triangular solve strong scales to 4000 cores (SIAM CSC'18, SIAM PP'20, SC'23)

- **Open-source software**

- Used 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.



<https://portal.nersc.gov/project/sparse/superlu/>

# STRUMPACK

## Structured Matrix Package

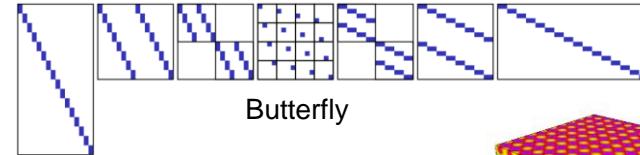
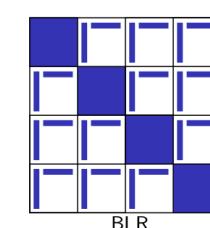
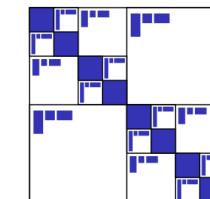
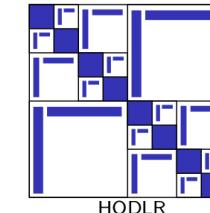


Hierarchical solvers for dense rank-structured matrices and fast algebraic sparse solver and robust and scalable preconditioners.

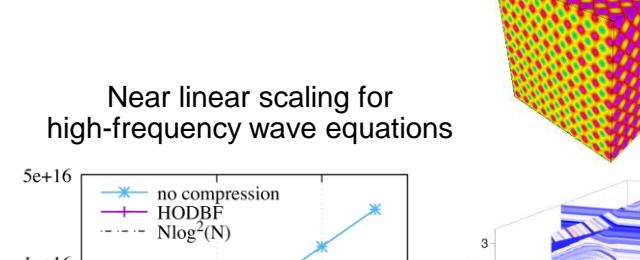


### Dense Matrix Solvers using Hierarchical Approximations

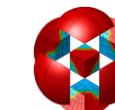
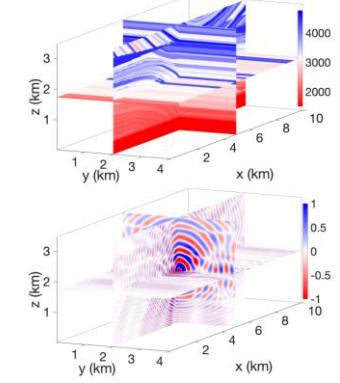
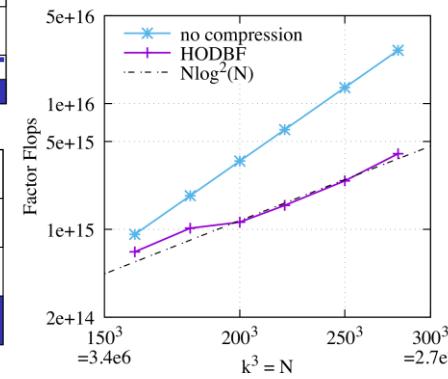
- Hierarchical partitioning, low-rank approximations
- Hierarchically Semi-Separable (HSS), Hierarchically Off-Diagonal Low-Rank (HODLR), Hierarchically Off-Diagonal Butterfly (HODBF), Block Low-Rank (BLR), Butterfly
- C++ Interface to ButterflyPACK (Fortran)
- Applications: BEM, Cauchy, Toeplitz, kernel & covariance matrices, ...
- Asymptotic complexity much lower than LAPACK/ScaLAPACK routines



Butterfly



Near linear scaling for high-frequency wave equations



[github.com/pgphysels/STRUMPACK](https://github.com/pgphysels/STRUMPACK)

# Agenda

<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15		Break
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30		Lunch
1:30 – 2:45	<b>Nonlinear Solvers (PETSc) – Richard Mills</b>	<b>Time Integration (SUNDIALS) – David Gardner</b>
2:45 – 3:15		Break
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:30	Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software	
6:30 – 7:30		Dinner
7:30 – 8:30		SME Speed Dating

# 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 and DAE time integrators:**

- CVODE: adaptive order and step BDF (stiff) & Adams (non-stiff) methods for ODEs
- ARKODE: adaptive step implicit, explicit, IMEX, and multirate Runge-Kutta methods for ODEs
- IDA: adaptive order and step BDF methods for DAEs
- CVODES and IDAS: provide forward and adjoint sensitivity analysis capabilities

- **Nonlinear Solvers:** KINSOL – Newton-Krylov; accelerated Picard and fixed point

- **Modular Design:** Easily incorporated into existing codes; Users can supply their own data structures and solvers or use SUNDIALS provided modules

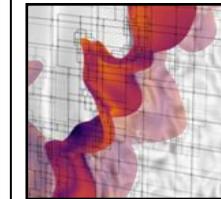
- **Support on NVIDIA, AMD, and Intel GPUs:**

- Vectors: CUDA, HIP, OpenMP Offload, RAJA, SYCL (DPC++)
- Linear solvers: cuSOLVER, MAGMA, matrix-free Krylov methods

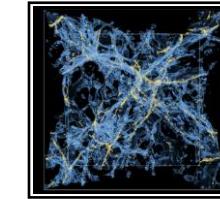
- **Open Source:** BSD License; Download from LLNL site, GitHub, or Spack

- Supported by extensive documentation; user email list with an active community
- Available through MFEM, AMReX, deal.II, and PETSc

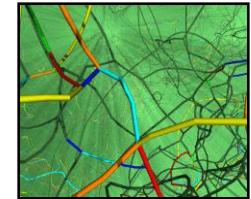
**SUNDIALS is used worldwide in applications throughout research and industry**



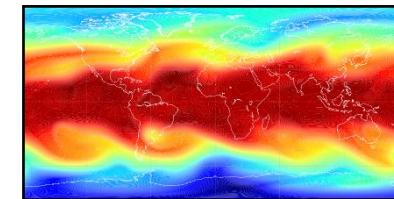
Combustion  
(Pele)



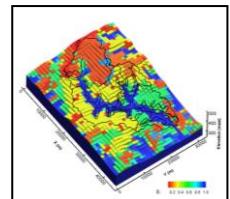
Cosmology  
(Nyx)



Dislocation dynamics  
(ParaDIS)



Atmospheric Dynamics  
(Tempest)



Subsurface flow  
(ParFlow)

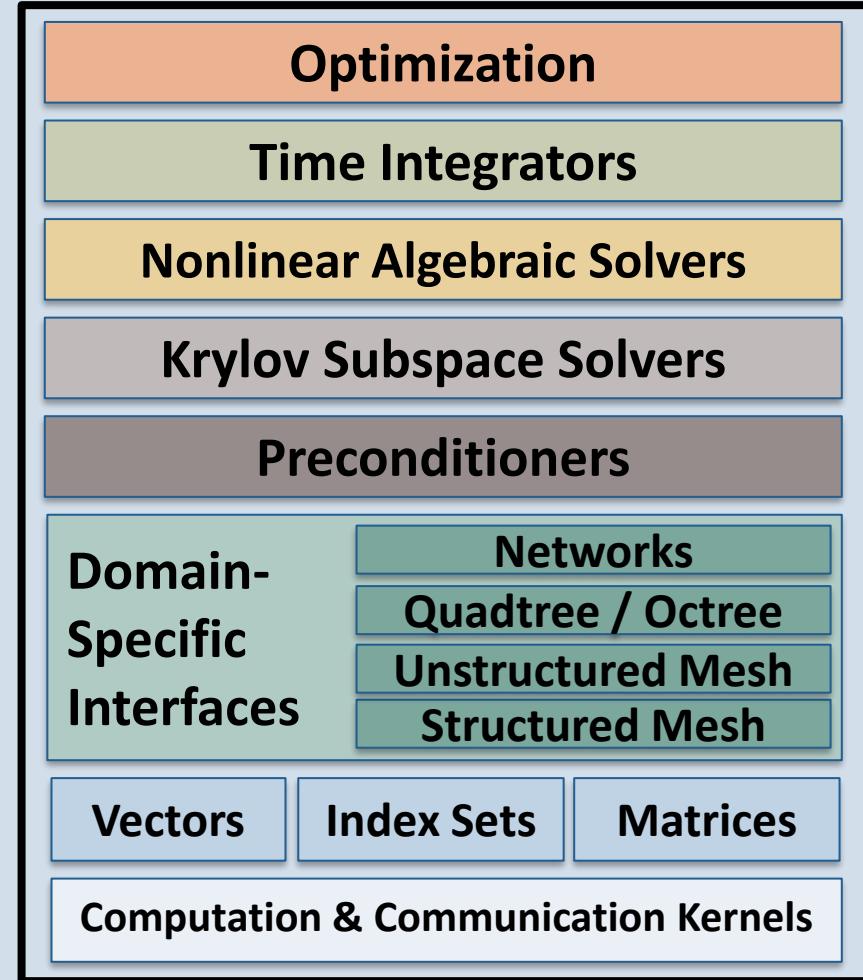
 Lawrence Livermore  
National Laboratory

 ECP

 FASTMATH

 xSDK

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

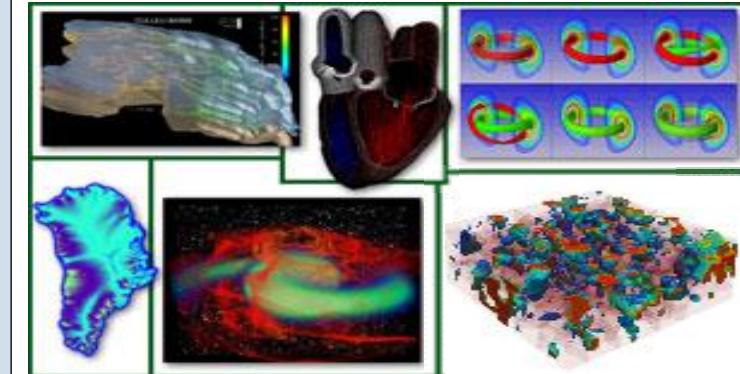


- **Easy customization and composable solvers at runtime**

- 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; NVIDIA, AMD, and Intel GPUs
- Thousands of users worldwide



PETSc provides the backbone of diverse scientific applications.  
 clockwise from upper left: hydrology, cardiology, fusion, multiphase steel, relativistic matter, ice sheet modeling



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

Argonne  
NATIONAL LABORATORY

# Agenda

<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

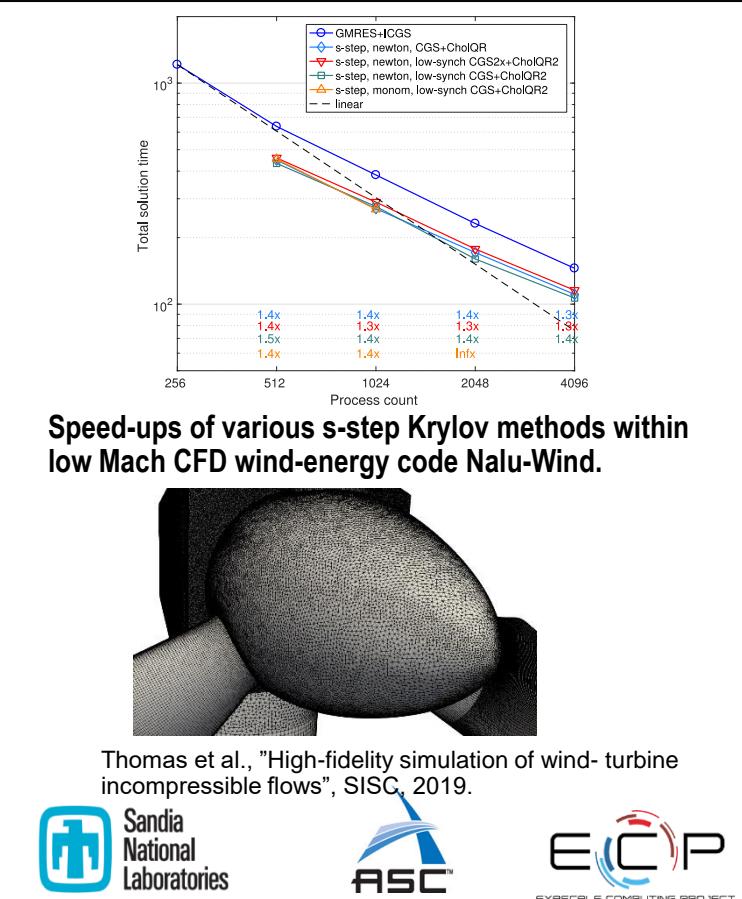
Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15		Break
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30		Lunch
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15		Break
3:15 – 4:30	<b>Optimization (TAO) – Toby Isaac</b>	<b>Iterative Solvers &amp; Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper</b>
4:30 – 5:30	Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software	
6:30 – 7:30	Dinner	
7:30 – 8:30	SME Speed Dating	

# Trilinos/Belos

**Iterative Krylov-based solvers.** Templatized 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:  $AX_i = B_i, i=1,\dots,t$
  - Sequences of multiple-RHS systems:  $A_iX_i = B_i, i=1,\dots,t$
- **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.



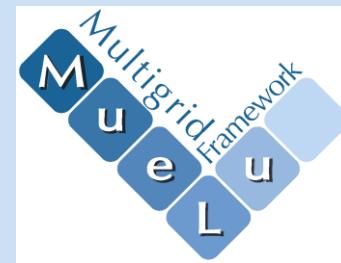
<https://trilinos.github.io/belos.html>

# 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)



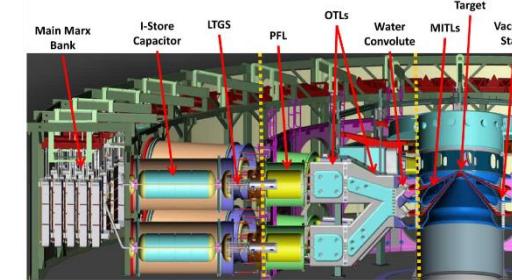
### ▪ Capabilities

- Aggregation-based coarsening
- **Smoothers**: Jacobi, GS,  $\sqrt{1}$  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

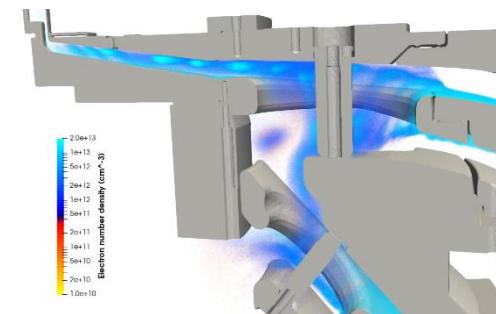
- AMG for multiphysics
- Multigrid for coupled structured/unstructured meshes
- Algorithm selection via machine learning

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  $H(\text{curl})$  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>

# <https://xsdk-project.github.io/MathPackagesTraining2023/>

The screenshot shows a web browser window with the URL <https://xsdk-project.github.io/MathPackagesTraining2023/>. The browser's address bar and tabs are visible at the top. A red circle highlights the 'LESSONS' tab in the navigation menu on the left. The main content area lists several numerical packages and their applications:

- STRUCTURED MESHING & DISCRETIZATION WITH AMREX
- UNSTRUCTURED MESHING & DISCRETIZATION WITH MFEM
- KRYLOV SOLVERS & ALGEBRAIC MULTIGRID WITH HYPRE
- KRYLOV SOLVERS & PRECONDITIONING WITH MUELU
- SPARSE, DIRECT SOLVERS WITH SUPERLU
- RANK STRUCTURED SOLVERS WITH STRUMPACK
- NONLINEAR SOLVERS WITH PETSC
- NUMERICAL OPTIMIZATION WITH TAO
- TIME INTEGRATION WITH SUNDIALS

Below the package list, there are two sections:

- Enhanced Productivity**
  - Numerically Rigorous.
  - Community Adopted.
  - Extremely Scalable.
  - Performance Portable.
  - Easy Download.
  - Easy Configure & Install.
  - Easy Dependencies.
  - Easy Update.
  - Development Resources.
  - Shared Know-How.
  - Common Tools.
  - Training.

A red button at the bottom center says "How can I get involved?".

## • Hands-on Lessons

# <https://xsdk-project.github.io/MathPackagesTraining2023/>

The screenshot shows a web browser window with the URL <https://xsdk-project.github.io/MathPackagesTraining2023/>. The page is titled "Math Packages Training 2023". At the top, there is a navigation bar with links like News, Popular, MyLLNL, Inclusion Cam..., OneDrive, CSP EOR, ServiceNow, IDEAS-ECP - Google Drive, and My Personal WebEx. Below the navigation bar, there are tabs for INTRO, LESSONS, and PACKAGES. The PACKAGES tab is highlighted with a red circle. On the left side, there is a sidebar with the title "Open Source" and a list of packages: AMREX, MFEM, PUMI, PUMIPIC, PUMIPIC APPS, HYPRE, SUPERLU, STRUMPACK, PETSC/TAO, SUNDIALS, TRILINOS, TRILINOS/BELOS, TRILINOS/MUELU, and ALCF USER GUIDES. Below this sidebar, there is a red button labeled "How can I get involved?". The main content area features logos for ECP, FASTMATH, and Spack, along with sections for "Interoperability & Ease of Use" and "Enhanced Productivity".

- Hands-on Lessons
- Packages

# Agenda

<https://extremecomputingtraining.anl.gov/agenda-2023/#Track-5>

Time	Room?	Room?
8:30 – 9:30		Introduction to Numerical Software – Ulrike Yang
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15		Break, Subject Matter Expert (SME) Selections, Panel Questions
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30		Lunch, SME Selections, Panel Questions
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15		Break, SME Selections, <b>Panel Questions Due</b>
3:15 – 4:30	Optimization (TAO) – Tobin Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos, Belos & MueLu) – Christian Glusa, Graham Harper
4:30 – 5:30	<b>Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software</b>	
5:30 – 6:30	Unstructured Time: <b>SME Selections Due</b> , Informal Discussion, Continue Hands-on	
6:30 – 7:30	Dinner	
7:30 – 9:30	Optional Activity: SME Speed-dating	

**Next steps:** <https://xsdk-project.github.io/MathPackagesTraining2023/agenda>

- WrapUp (Ann Almgren) @ 4:30pm

Panel: Main Room @ 4:45 pm

- SME Speed Dating: @7:30pm

- During breaks and lunch

- Provide Panel Questions

**Due: 3:15 pm**

- Sign up for discussions with numerical software developers (optional)

- Your email address

**Due 6:30 pm**



### Subject Matter Expert (SME) 2-on-1 interviews

This is an optional activity. It is a great opportunity to spend some time chatting with various subject matter experts (SMEs).

In the form below, you may enter your first, second and third priorities for up to three, 20 minute, two-on-one discussions with various SMEs during the evening session.

The screenshot shows a navigation bar with 'INTRO' (highlighted in blue), 'LESSONS', and 'PACKAGES'. Below the navigation are links for 'SETUP INSTRUCTIONS', 'TODAY'S AGENDA', 'VIP TALKS', 'GETTING HELP', and 'SESSION SELECTION SURVEY'. A red box highlights the 'PANEL QUESTION SUBMISSIONS' button. Another red box highlights the 'SME SPEED DATING SELECTIONS' button. To the right, there's a logo for 'So my code will see the future xSDK Spack better scientific software'. Below the navigation, there are sections for 'Interoperability & Ease of Use' and 'Enhanced Productivity', each with a bulleted list of benefits.

Interoperability & Ease of Use	Enhanced Productivity
<ul style="list-style-type: none"><li>• Easy Download.</li><li>• Easy Configure &amp; Install.</li><li>• Easy Dependencies.</li><li>• Performance Portable.</li><li>• Easy Update.</li></ul>	<ul style="list-style-type: none"><li>• Development Resources.</li><li>• Shared Know-How.</li><li>• Common Tools.</li><li>• Training.</li></ul>

# Panel: Extreme-Scale Numerical Algorithms and Software

- **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

- Panelists



Sherry Li, LBL



Toby Isaac, ANL



Graham Harper, SNL



Andrew Myers, LBL

- Moderator



Ann Almgren, LBL

## Panel Question Submission Form

Please enter here a question you would like to ask our panelists during the 45 minute panel session.

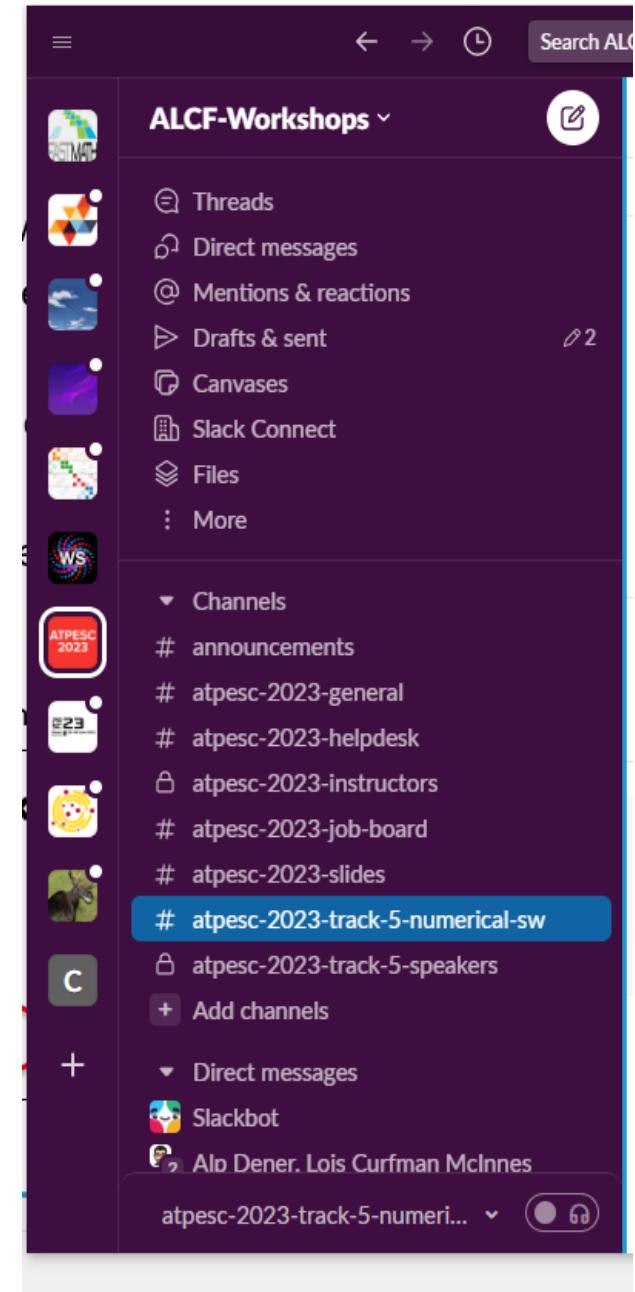
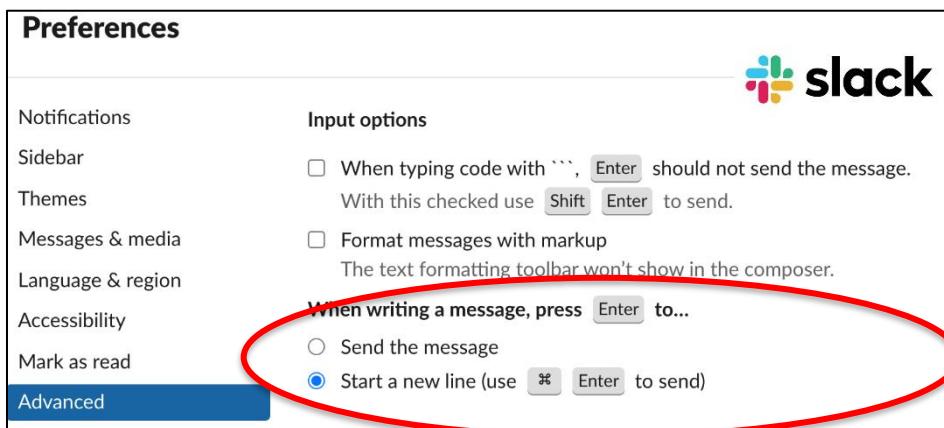
We ask that you please include your name in case we may need to call upon you to clarify your question.

# Using Slack



- Recommend using the desktop app, but browser ok too
- **# atpesc-2023-track-5-numerical-sw** channel
  - For all chat during presentations
  - For all chat outside any specific parallel session
  - For general help
  - Recommend using the thread option to help keep track of discussions on subtopics

Tip: Consider setting Preferences to customize when to send



# 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 2023

## Extreme-scale numerical algorithms and software

Integrated lectures and hands-on examples, panel session, individual discussions ... and more!



Ann Almgren, LBL



Andrew Myers, LBL



Tzanio Kolev, LLNL



Mark Shephard, RPI



Cameron Smith, RPI



Ulrike Yang, LLNL



Christian Glusa, SNL



Graham Harper, SNL



Sherry Li, LBL



Pieter Ghysels, LBL



Satish Balay, ANL



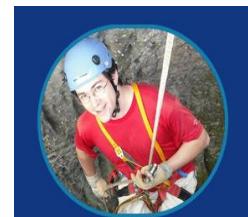
Sarah Osborn, LLNL



Toby Isaac, ANL

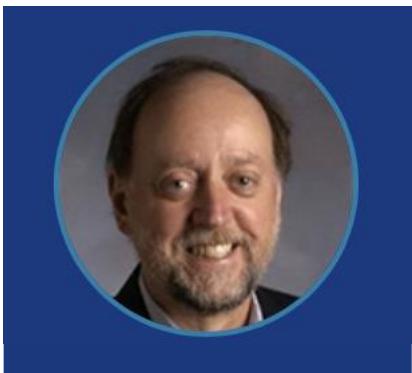


David Gardner, LLNL



Richard Mills, ANL

# VIPs of ATPESC Extreme-Scale Numerical Software Track



- **Jim Demmel, UC Berkeley** [[bio](#)]
  - **Communication-Avoiding Algorithms**
    - ATPESC 2023, Thursday, August 10, 7:30pm
    - ENLA Seminar, June 2020 [[video](#)]
- **Jack Dongarra, Univ of Tennessee** [[bio](#)]
  - **Growing up at Argonne National Lab**
    - ATPESC 2023, Monday, August 7, 7:30pm
    - Adaptive Linear Solvers and Eigensolvers, ATPESC 2019 [[video](#)]
- **David Keyes, KAUST** [[bio](#)]
  - **Efficient Computation thorough Tuned Approximation**
    - ATPESC 2023, Tuesday, August 1, 7:30pm
    - Adaptive Nonlinear Preconditioning for PDEs with Error Bounds on Output Functionals, University of Manchester, 2021 [[video](#) ]

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



<https://exascaleproject.org>



<https://scidac5-fastmath.lbl.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 11)

**Community efforts:**  
Join us!



<https://xsdk.info>

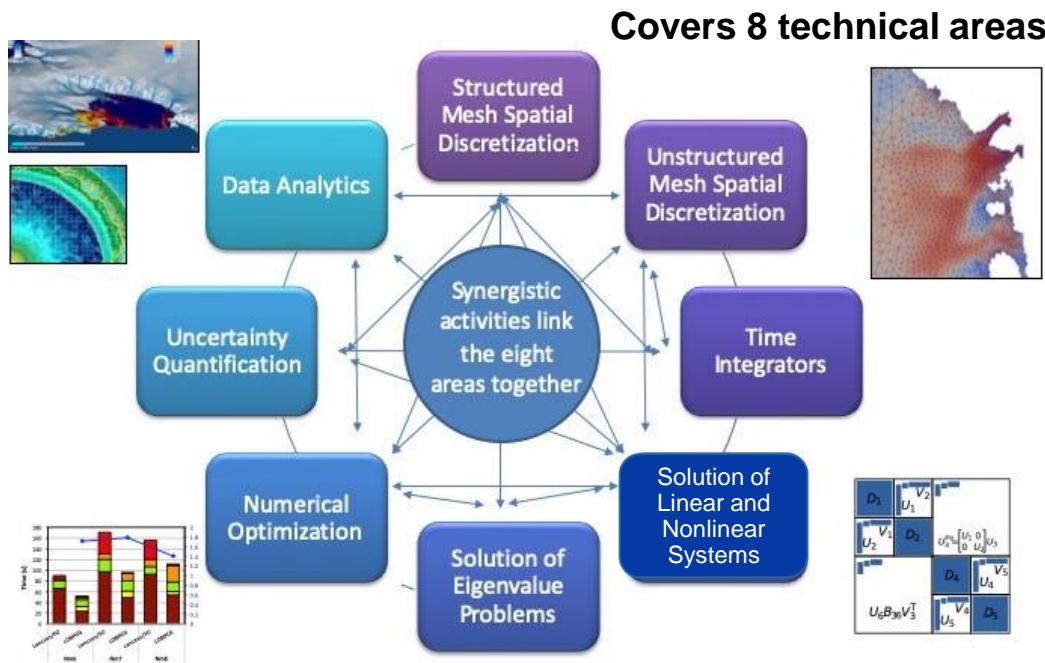


<https://e4s.io>



# FASTMath: Frameworks, Algorithms & Scalable Technologies for Mathematics

<https://scidac5-fastmath.lbl.gov/>



## 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



Argonne  
NATIONAL LABORATORY



OAK  
RIDGE  
National Laboratory

Sandia  
National  
Laboratories



USC Rensselaer SMU

100's of person years of experience building math software

*SuperLU*



*AMReX*

*PETSc*

*PUMI*  
Parallel Unstructured Mesh Infrastructure



*UQTk*

*Trilinos*



*symPACK*

*Albany*

*ZOLTAN*

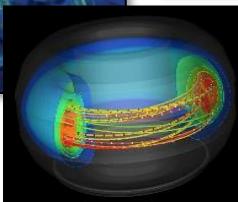
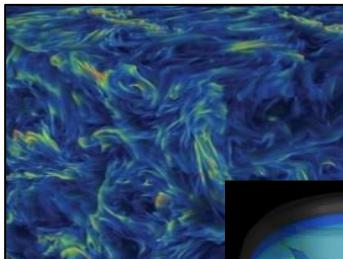
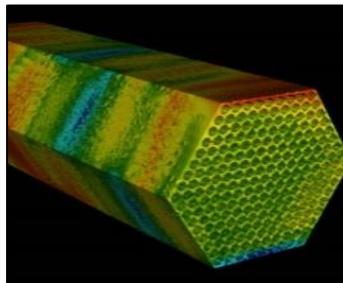
*hypre*  
high performance preconditioners



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

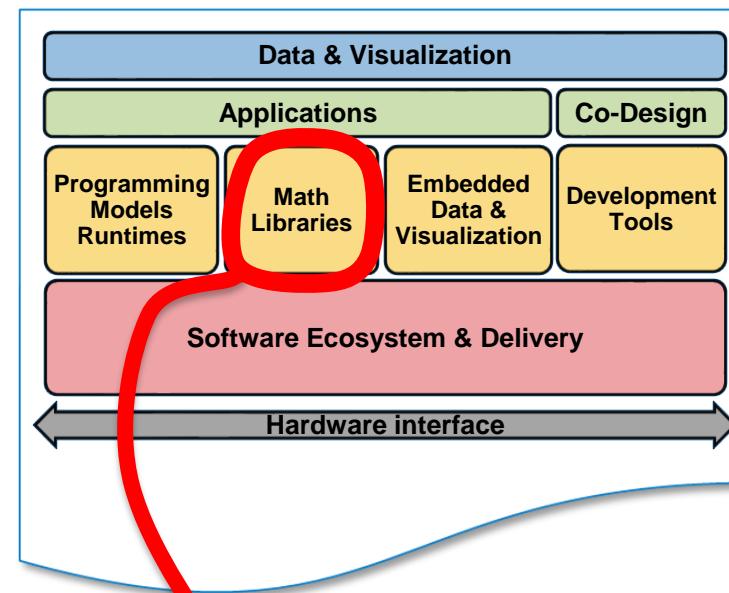
## Application Development

Science and mission applications



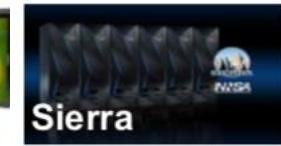
## Software Technology

Scalable software stack



## Hardware and Integration

Relationships: facilities with AD/ST, with vendors



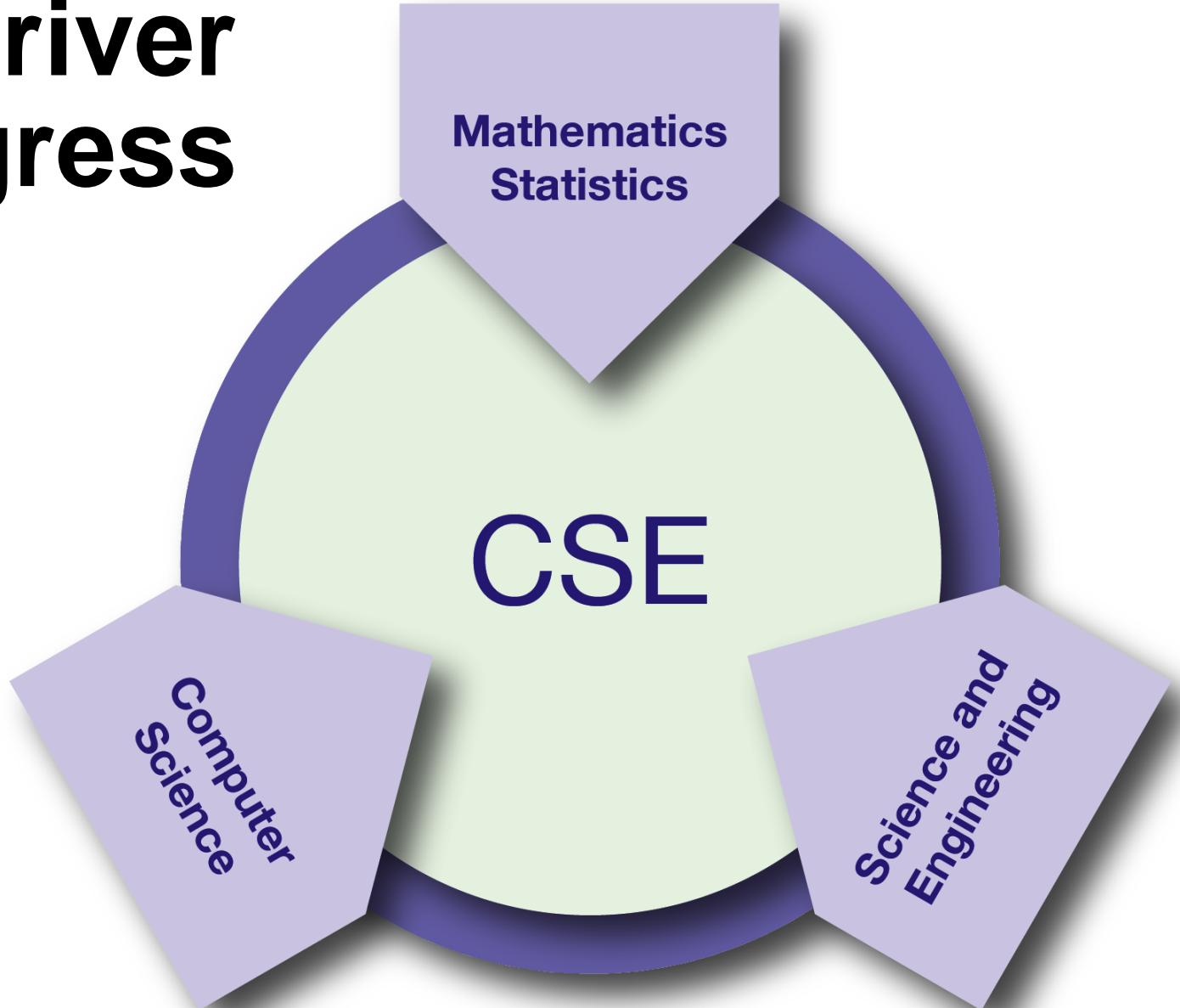
Emphasis for this presentation

# CSE: Essential driver of scientific progress

**CSE = Computational Science & Engineering**

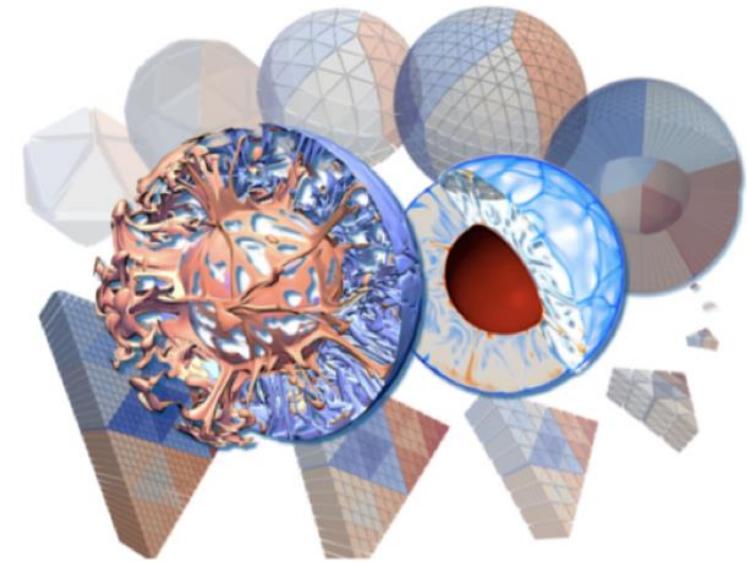
**Development and use of computational methods for scientific discovery**

- 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

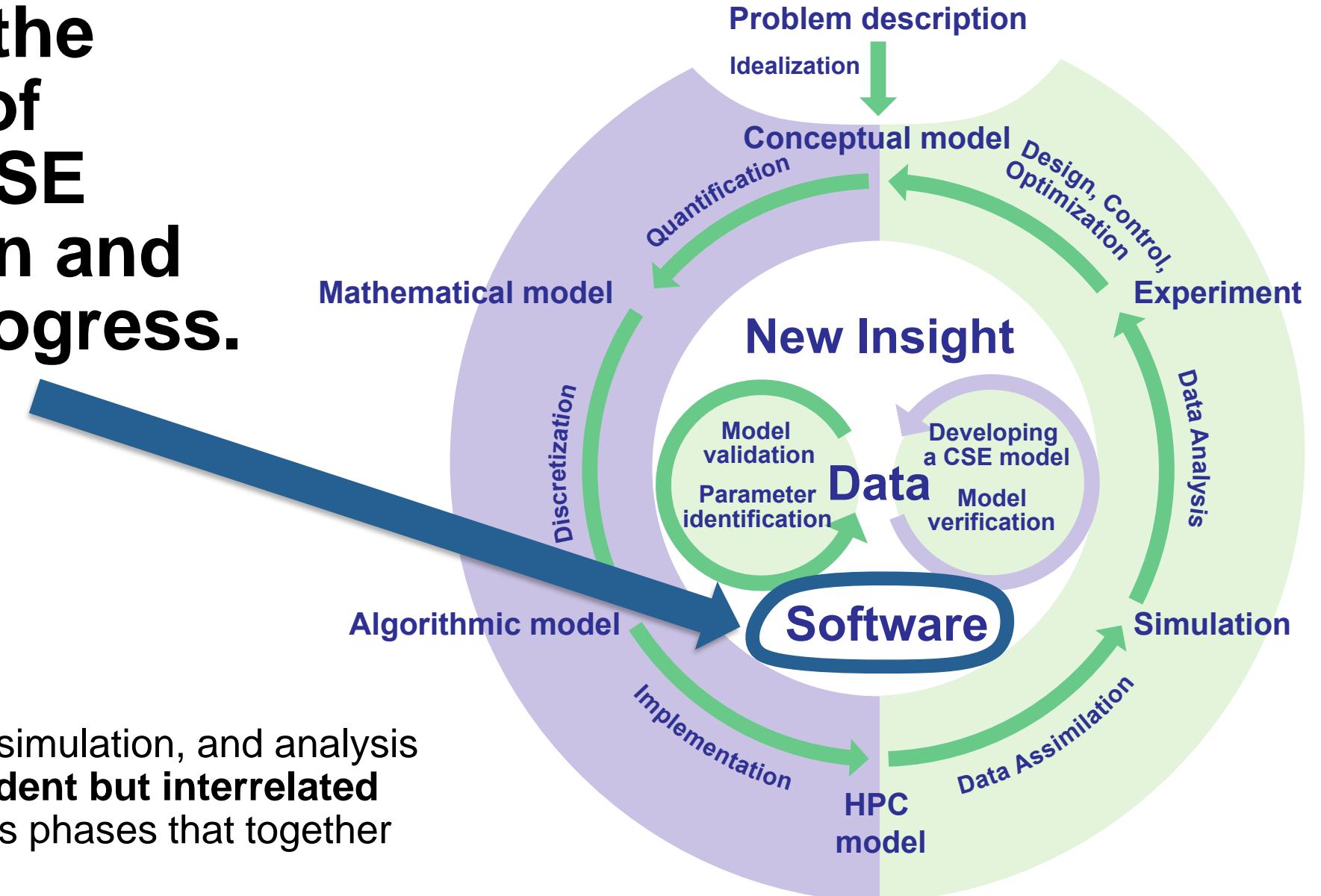
- Mathematical methods and algorithms
- CSE and HPC: Ubiquitous parallelism
- CSE and the data revolution
- CSE software
- CSE education & workforce development



## *Research and Education in Computational Science & Engineering*

U. Rüde, K. Willcox, L.C. McInnes, H. De Sterck, G. Biros, H. Bungartz, J. Coronas, 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.



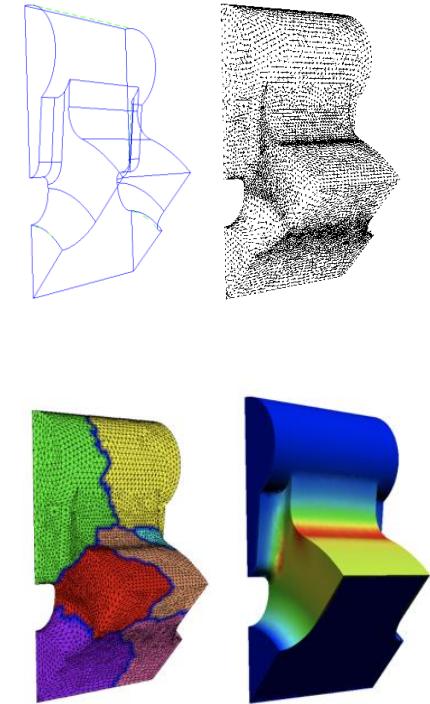
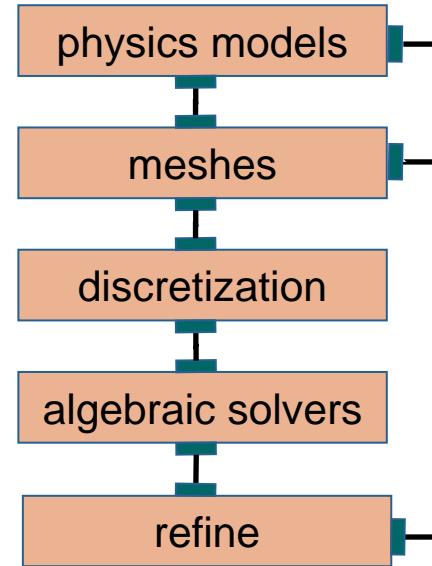
**CSE cycle:** Modeling, simulation, and analysis

- **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

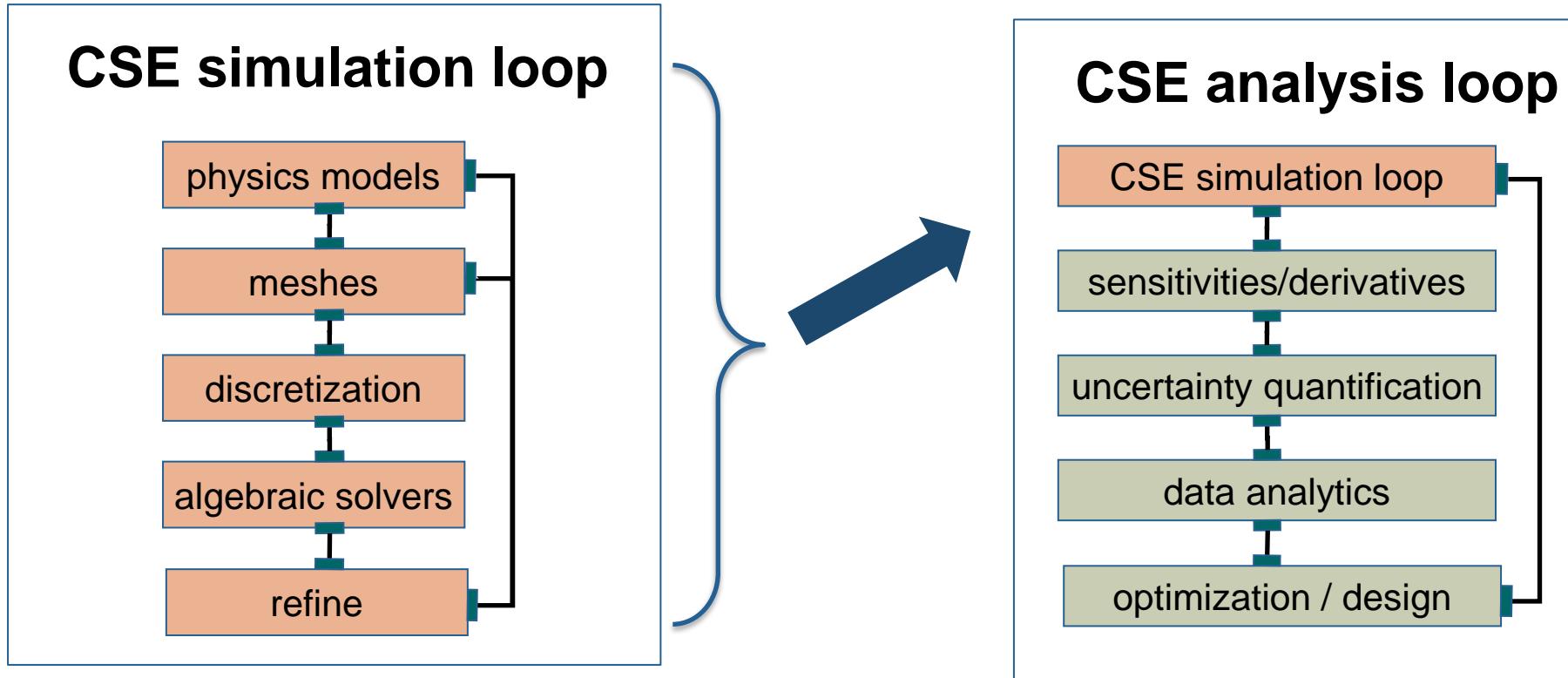
- Develop a mathematical model of the phenomenon of interest
- Approximate the model using a discrete representation
- Solve the discrete representation
- Adapt and refine the mesh or model
- Incorporate different physics, scales

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

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

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

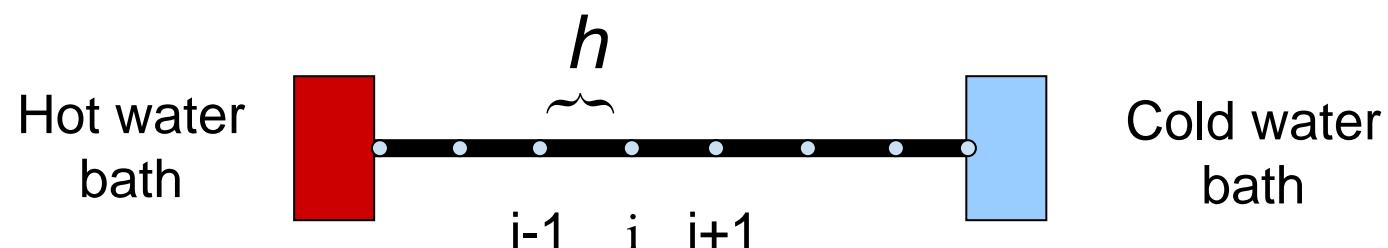


# The first step is to discretize the equations

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

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

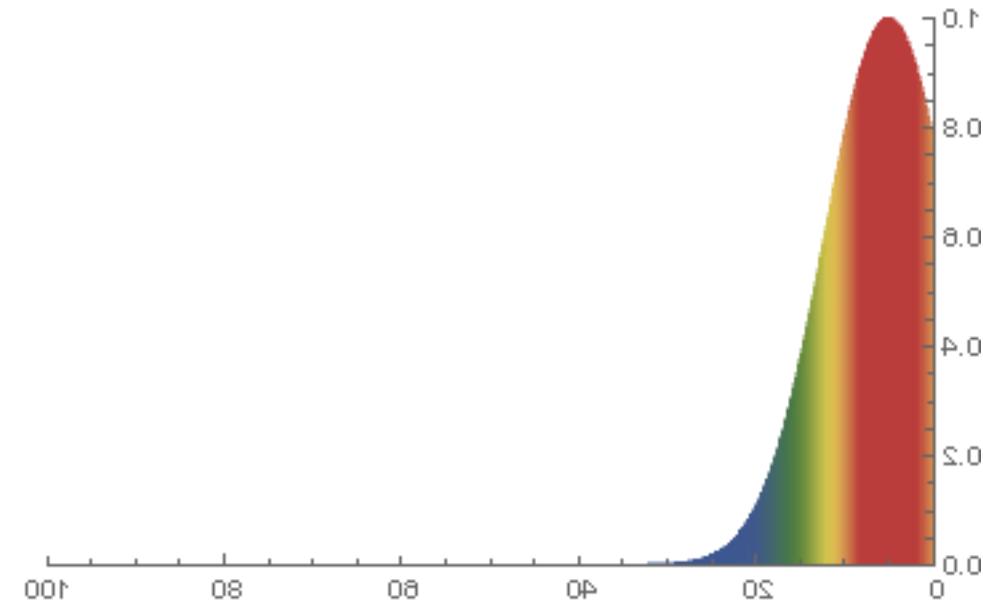
$$T_0 = 180^\circ \quad T_n = 0^\circ$$



# 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  $T_i$

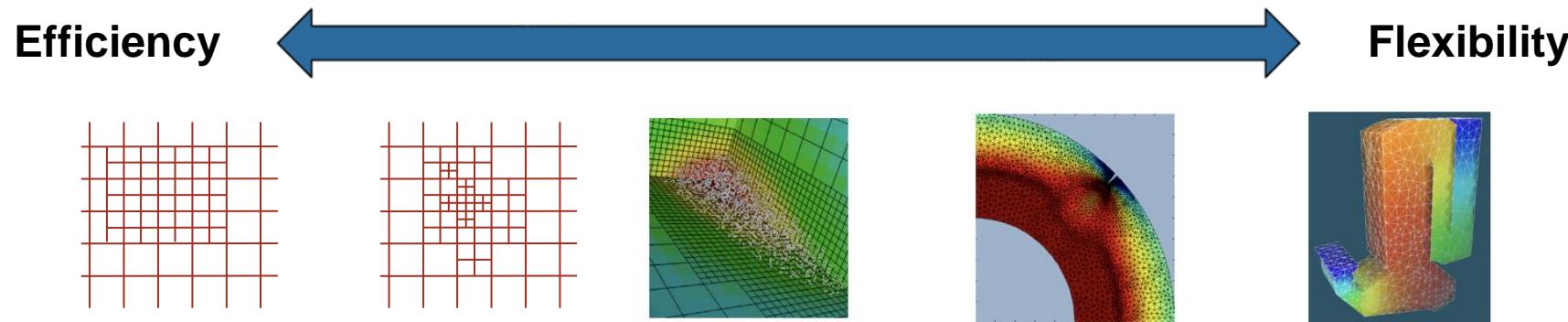
$$\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{n-1} \end{pmatrix} = \begin{pmatrix} 180 h^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$



- Visualize and analyze the results

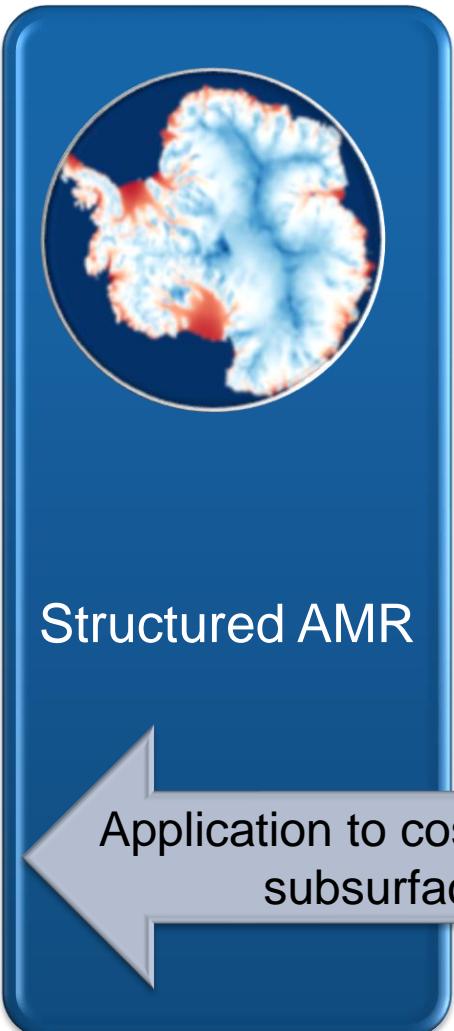
# As problems get more complicated, so do the steps in the process

- Different discretization strategies exist for differing needs

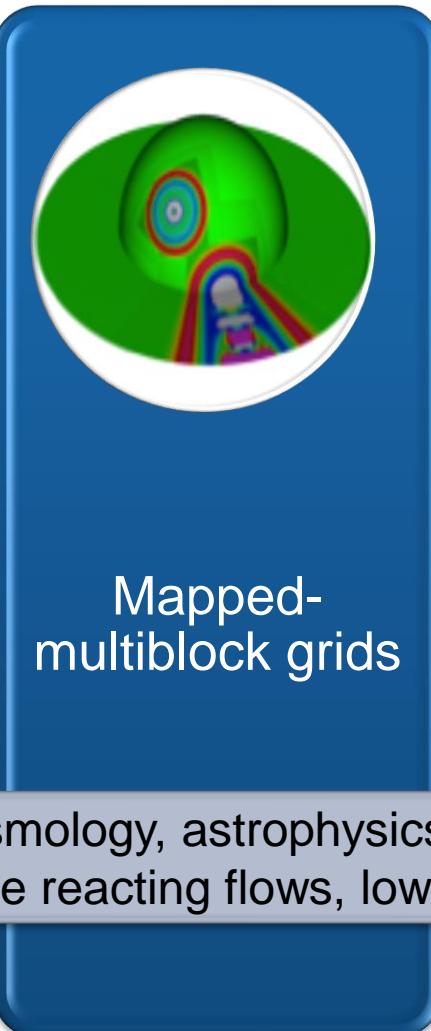


- Most problems are time dependent and nonlinear
  - Need higher algorithmic levels than linear solvers
- Increasingly combining multiple physical processes
  - Interactions require careful handling
- Goal-oriented problem solving requires optimization, uncertainty quantification

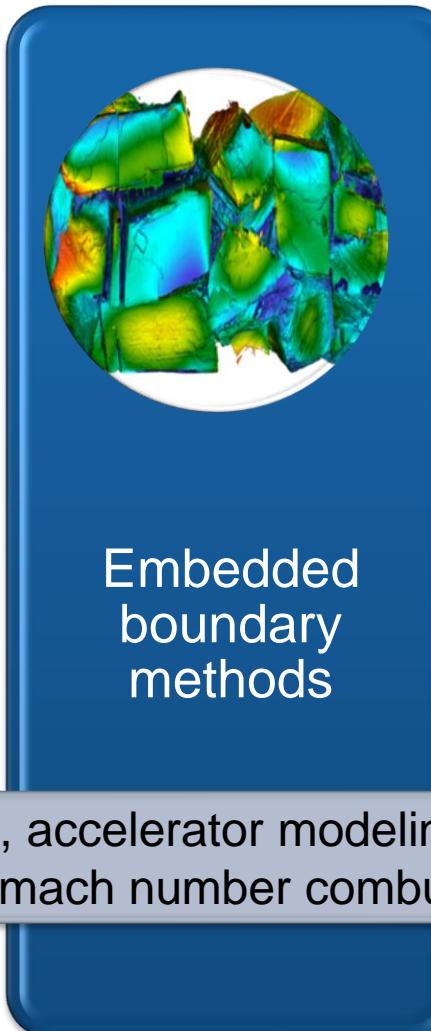
# Structured grid efforts focus on high-order, mapped grids, embedded boundaries, AMR, and particles



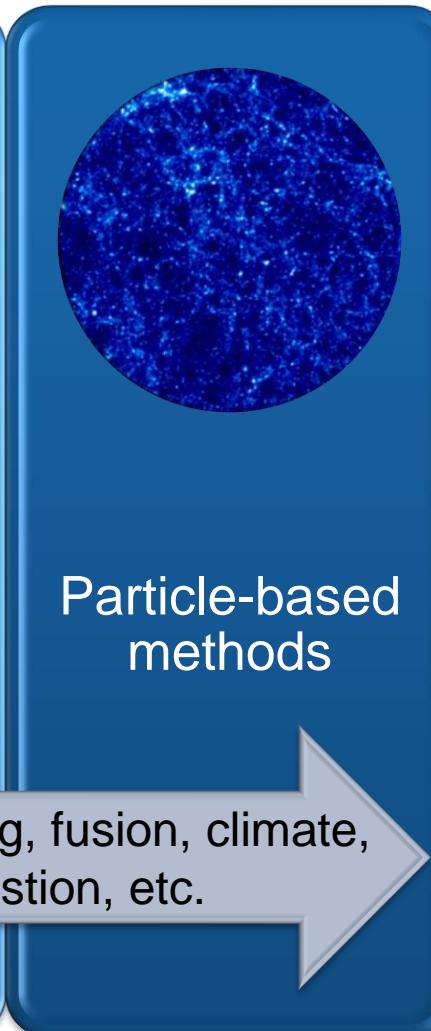
Structured AMR



Mapped-  
multiblock grids



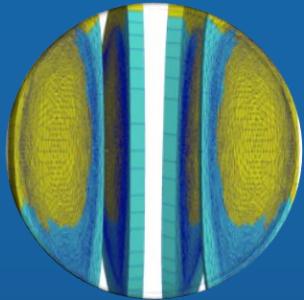
Embedded  
boundary  
methods



Particle-based  
methods

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

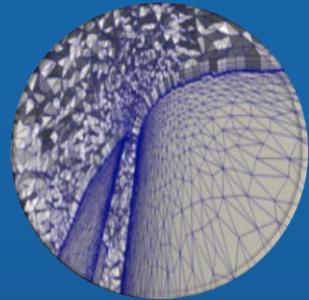
# Unstructured grid capabilities focus on adaptivity, high-order, and the tools needed for extreme scaling



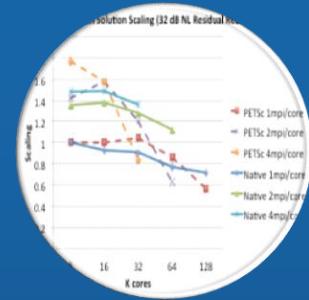
Parallel mesh  
infrastructures



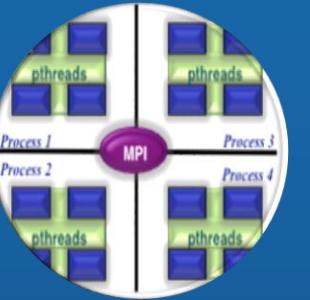
Dynamic load  
balancing



Mesh adaptation  
and quality  
control



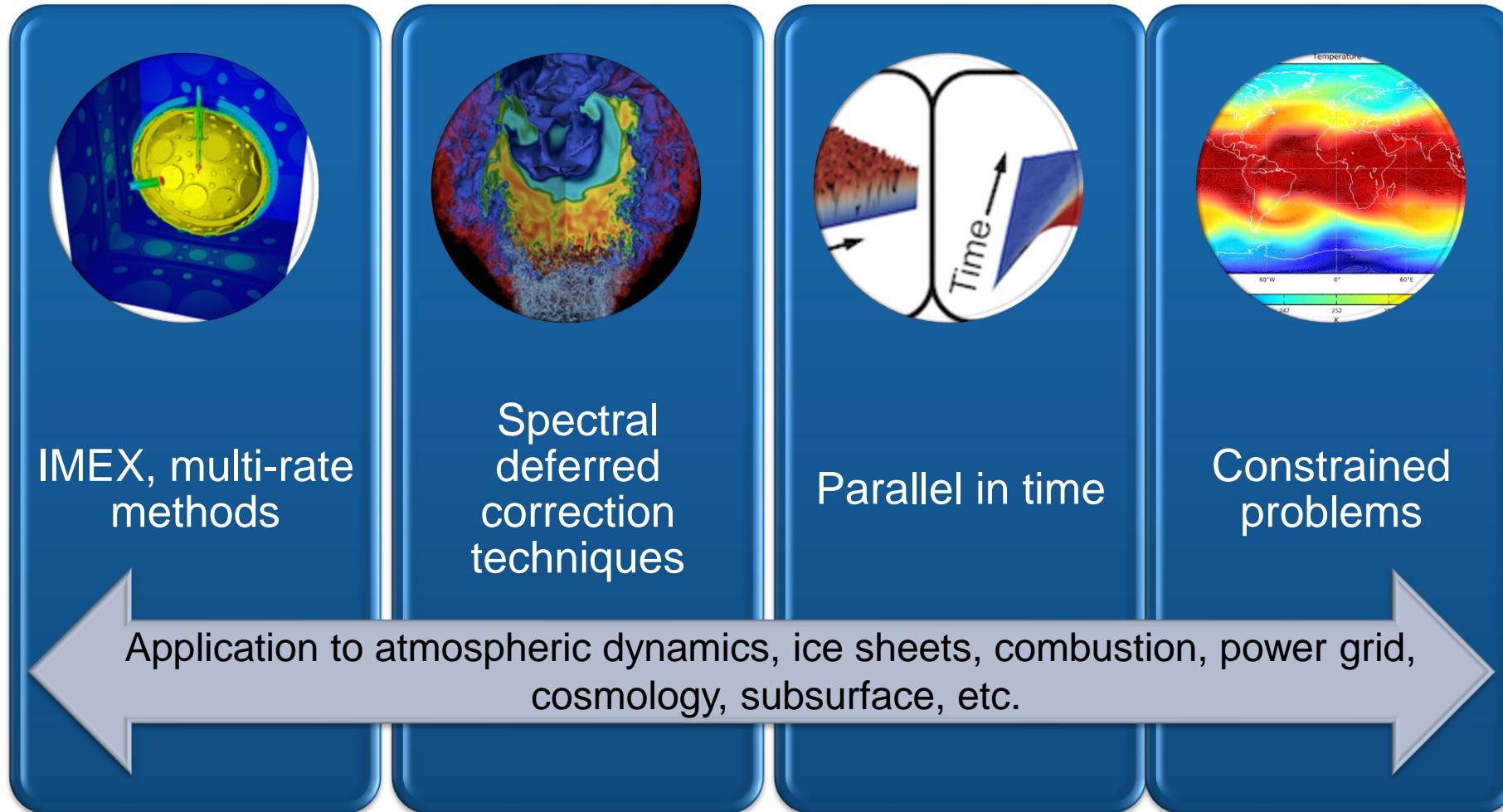
Parallel  
performance on  
unstructured  
meshes



Architecture  
aware  
implementations

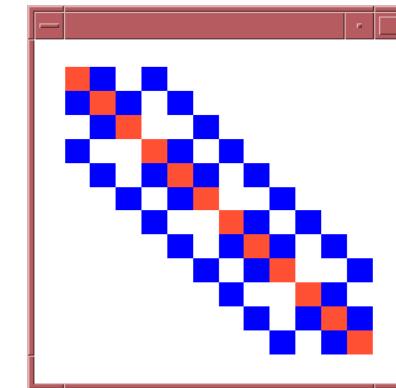
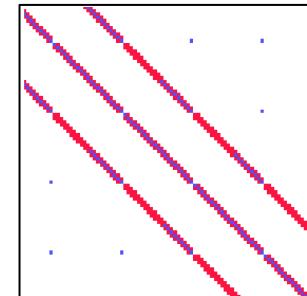
Application to fusion, climate, accelerator modeling, NNSA applications,  
nuclear energy, manufacturing processes, etc.

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

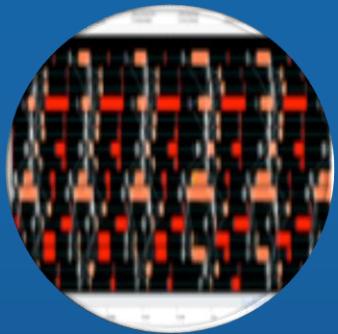


# As problems grow in size, so do corresponding discrete systems

- 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.



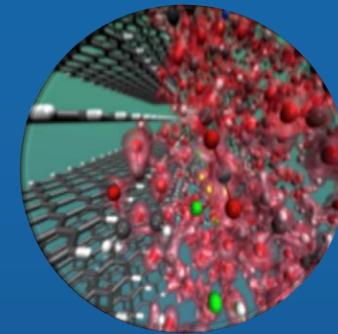
# Research on algebraic systems provides key solution technologies to applications



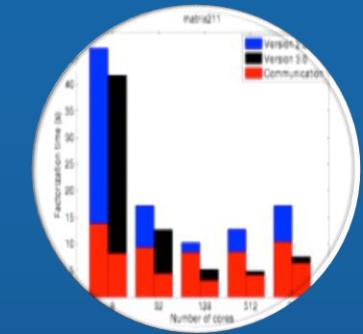
Linear system  
solution using direct  
and iterative solvers



Nonlinear system  
solution using  
acceleration  
techniques and  
globalized Newton  
methods



Eigensolvers using  
iterative techniques  
and optimization



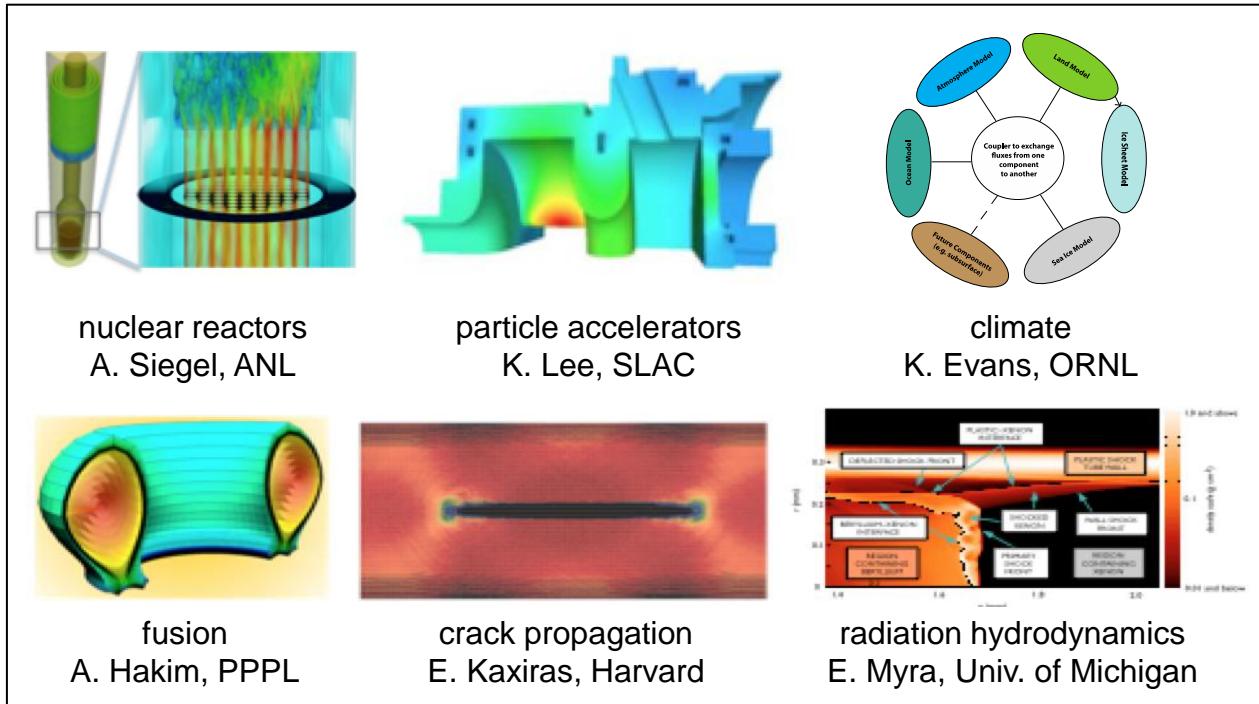
Architecture aware  
implementations

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

# 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



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



The International Journal of High Performance Computing Applications  
27(1) 4-83  
© The Author(s) 2012  
Reprints and permissions:  
[sagepub.co.uk/journalsPermissions.nav](http://sagepub.co.uk/journalsPermissions.nav)  
DOI: [10.1177/1094342012468181](https://doi.org/10.1177/1094342012468181)  
[hpc.sagepub.com](http://hpc.sagepub.com)



## Multiphysics simulations: Challenges and opportunities

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

[doi:10.1177/1094342012468181](https://doi.org/10.1177/1094342012468181)

# DOE HPC Roadmap to Exascale Systems

FY 2012



**Titan**  
ORNL  
Cray/AMD/NVIDIA



**Mira**  
ANL  
IBM BG/Q

FY 2016



**Theta**  
ANL  
Cray/Intel KNL



**Cori**  
LBNL  
Cray/Intel Xeon/KNL

FY 2018



**Summit**  
ORNL  
IBM/NVIDIA

FY 2021

FY 2022

FY 2023

FY 2024



**FRONTIER**  
ORNL  
HPE/AMD



**Aurora**  
ANL  
HPE/Intel



**Perlmutter**  
LBNL  
HPE/AMD/NVIDIA



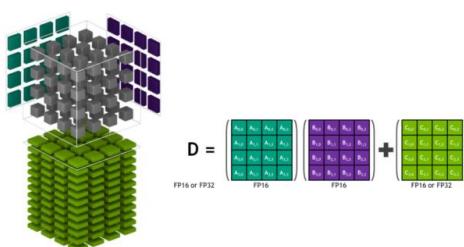
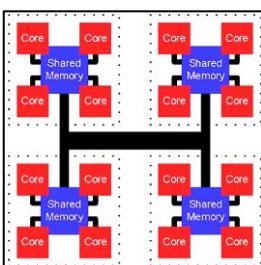
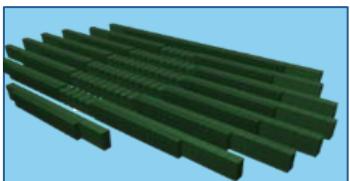
**CROSSROADS**  
LANL/SNL  
HPE/TBD



**EL CAPITAN**  
LLNL  
HPE/AMD

# 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.



$$D = \begin{pmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{pmatrix} + \begin{pmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{pmatrix}$$

# 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\\_\(computing\)](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

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

# 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

### 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 Math Libraries



# Software libraries are not enough

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

**The way you get  
programmer productivity  
is by eliminating lines of  
code you have to write.**

– Steve Jobs, Apple World Wide Developers Conference, Closing Keynote, 1997

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

# Need software ecosystem perspective

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

## Ecosystems are challenging!

**“We often think that when we have completed our study of one we know all about two, because ‘two’ is ‘one and one.’ We forget that we still have to make a study of ‘and.’ ”**



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

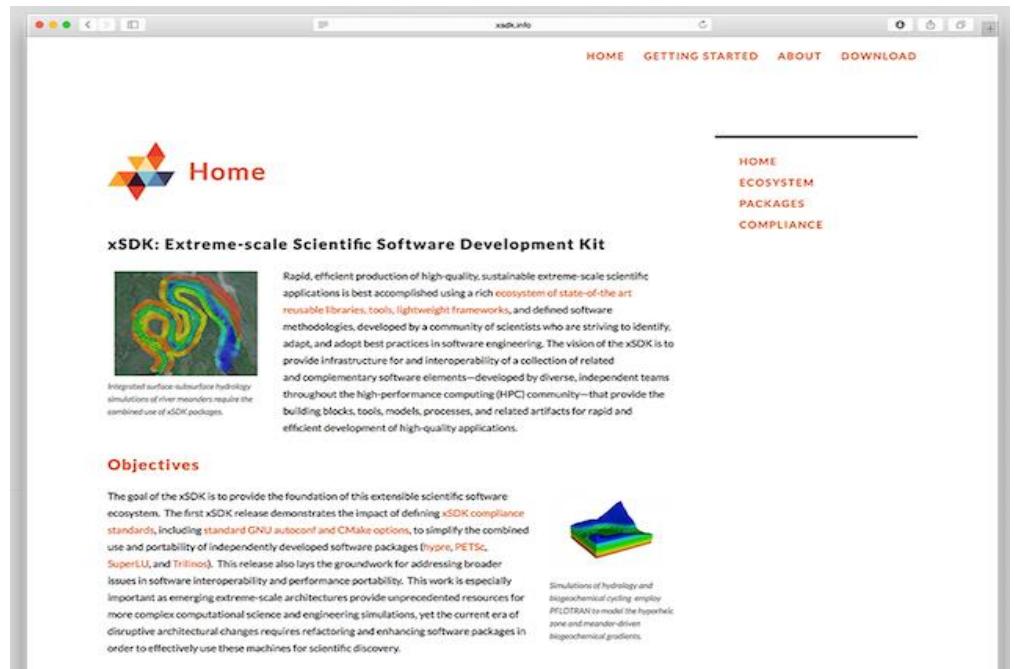


# Building the foundation of a highly effective extreme-scale scientific software ecosystem

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

## Impact:

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



website: [xSDK.info](http://xSDK.info)

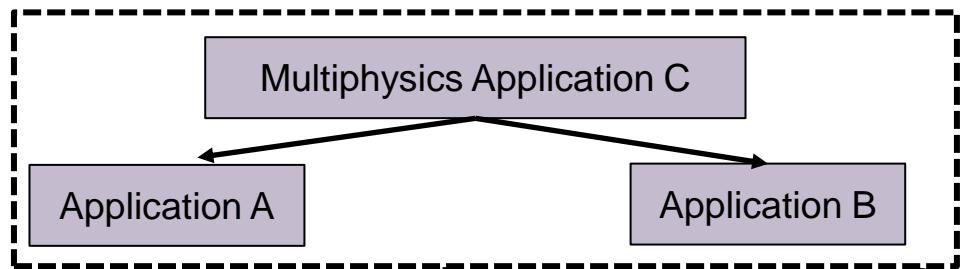


# xSDK Version 0.8.0: November 2022

<https://xsdk.info>

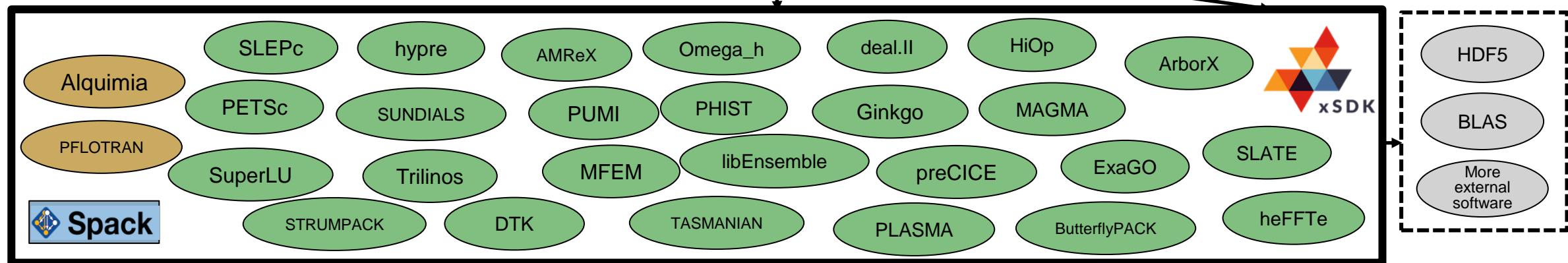
<https://xsdk.info>

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



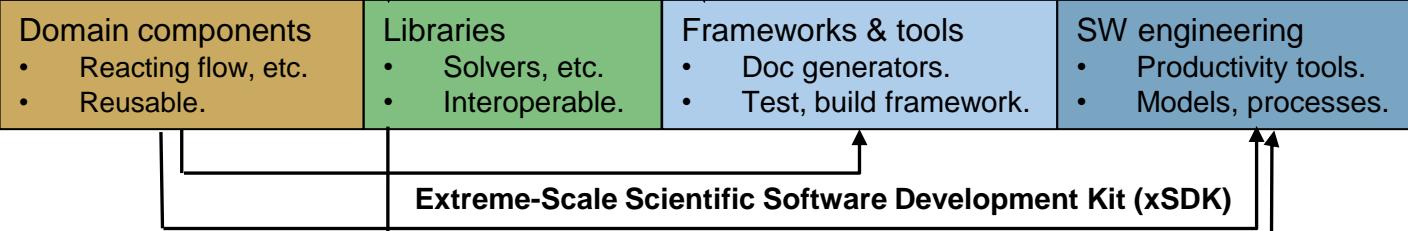
xSDK functionality, Nov 2022

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



## November 2022

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

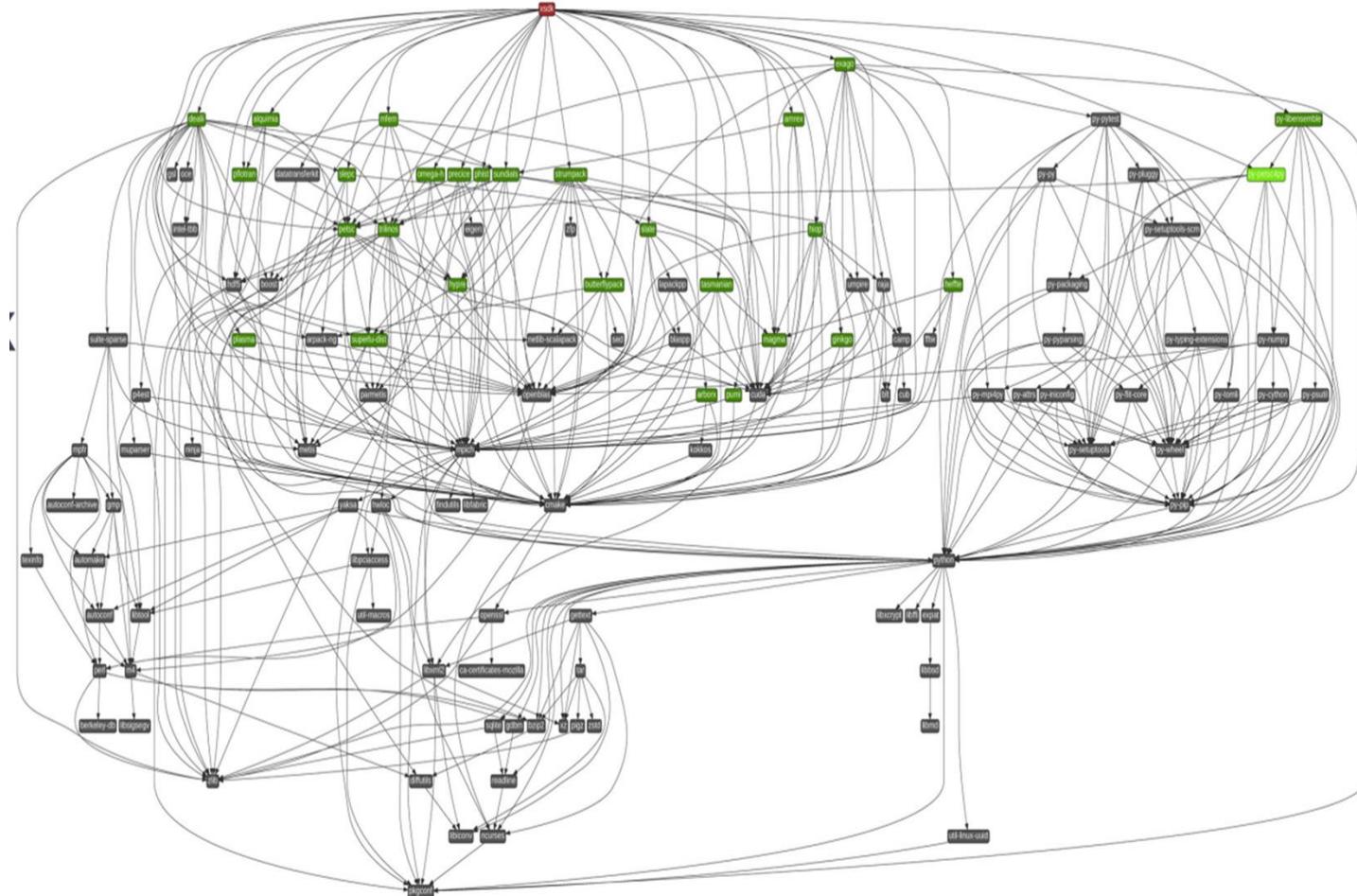


**Impact:** Improved code quality, usability, access, sustainability

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

# 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 (see Track 3: Software Productivity and Sustainability)
- Spack allows the xSDK to be deployed with a single command
  - User can optionally choose compilers, build options, etc.



# xSDK Libraries





# xSDK: <https://xsdk.info>

Building the foundation of an extreme-scale scientific software ecosystem

**xSDK community policies:** Help address challenges in interoperability and sustainability of software developed by diverse groups at different institutions

<https://github.com/xsdk-project/xsdk-community-policies>

**xSDK compatible package: must satisfy the mandatory**

**xSDK policies (M1, ..., M17)**

Topics include configuring, installing, testing, MPI usage, portability, contact and version information, open-source licensing, namespacing, and repository access

Also specify **recommended policies**, which currently are encouraged but not required (R1, ..., R8)

Topics include public repository access, error handling, freeing system resources, and library dependencies, documentation quality

**xSDK member package:**

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

**xSDK policies 1.0.0: Feb 2023**

- Facilitate combined use of independently developed packages

**Impact:**

- Improved code quality, usability, access, sustainability
- Foundation for work on deeper levels of interoperability and performance portability

We encourage feedback and contributions!

# xSDK community policies

<https://github.com/xsdk-project/xsdk-community-policies>

## Mandatory xSDK policies: must be satisfied

- M1.** Support portable installation through Spack  
(includes xSDK Spack variant guidelines)
- 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 publicly available repository.
- 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.** Have a debug build option.
- M17.** Provide sufficient documentation to support use and further development.

<https://x sdk.info/policies>



Version 1.0.0,  
February 2023

## Recommended xSDK policies: currently encouraged, but not required

- R1.** Provide at least one validation test that can be invoked through Spack.
- 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.
- R8.** Provide version comparison preprocessor macros.

**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?

# xSDK community policies

<https://github.com/xsdk-project/xsdk-community-policies>

## Mandatory xSDK policies: must be satisfied

**M1.** Support portable installation through Spack  
(includes xSDK Spack variant guidelines)

**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 publicly available repository.

**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.** Have a debug build option.

**M17.** Provide sufficient documentation to support use and further development.

<https://x sdk.info/policies>



Version 1.0.0,  
February 2023

## Recommended xSDK policies: currently encouraged, but not required

**R1.** Provide at least one validation test that can be invoked through Spack.

**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.

**R8.** Provide version comparison preprocessor macros.

**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?

# Interoperability is challenging, particularly for deeper levels!

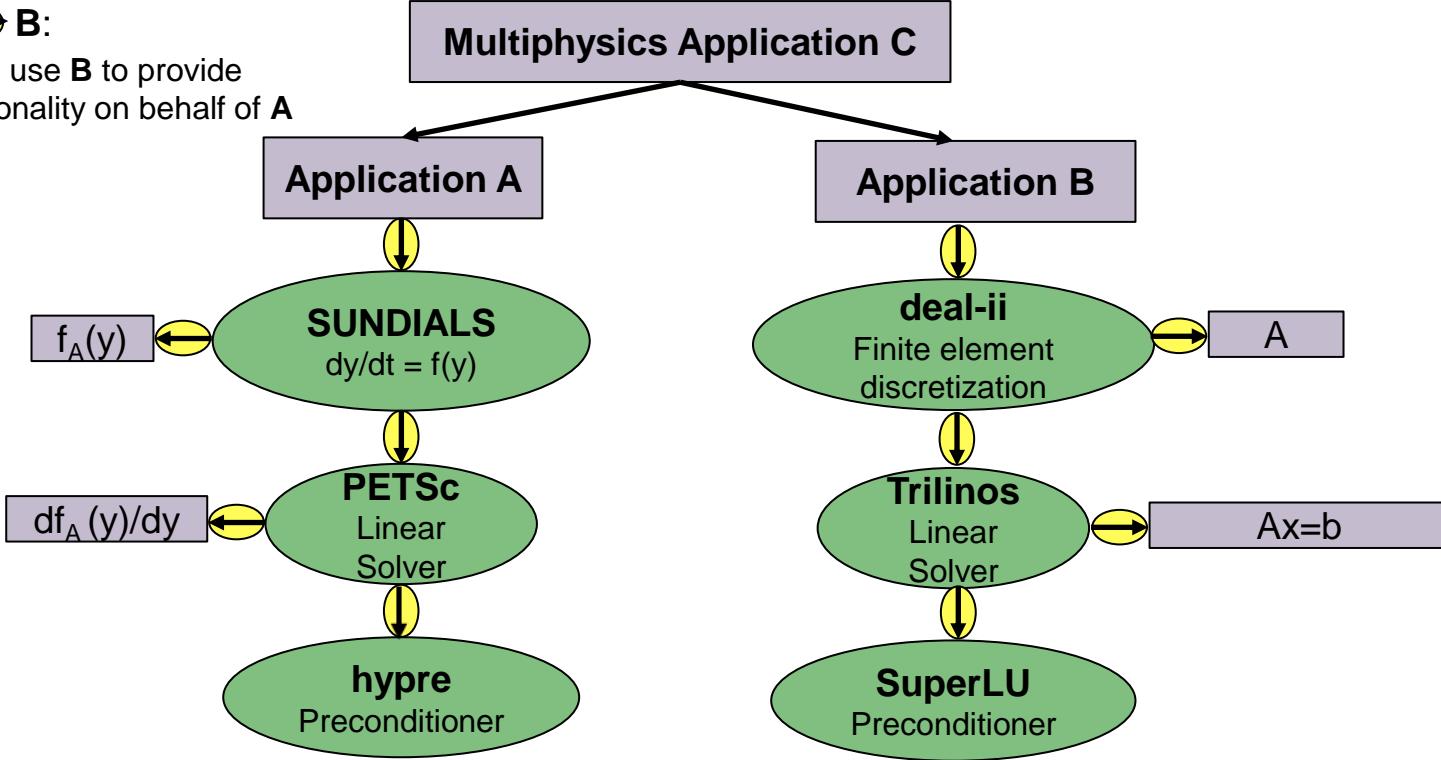
## Levels of package interoperability:

- **Interoperability level 1**
  - Both packages can be used (side by side) in an application
- **Interoperability level 2**
  - The libraries can exchange data (or control data) with each other
- **Interoperability level 3**
  - Each library can call the other library to perform unique computations

Notation:

$A \rightarrow B$ :

A can use B to provide functionality on behalf of A

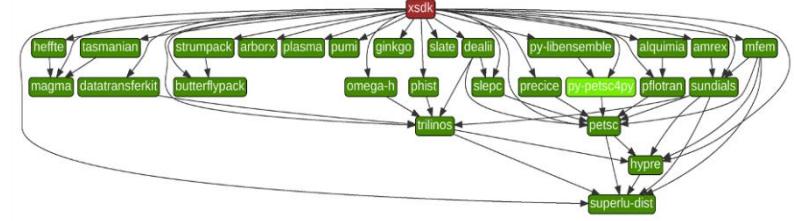


**xSDK4ECP:** Focus on inter-package functionality, denoted by

- Coordinating use of on-node resources
- Integrated execution (control inversion, adaptive execution strategies)

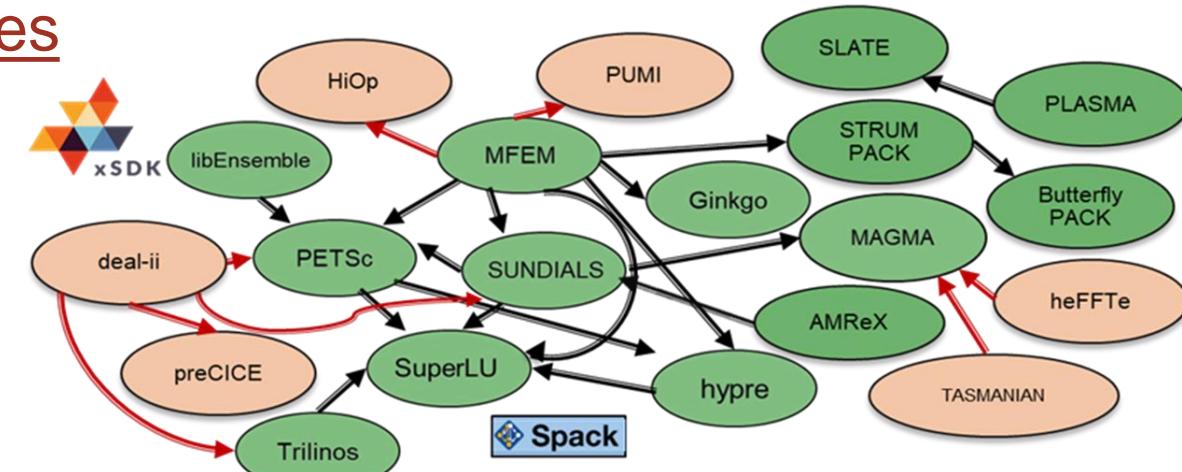
# Technical Challenges

- Staying up to date while facing continual changes
  - xSDK release schedule not aligned with individual xSDK library and Spack release schedule
  - Lower dependencies can cause additional problems
- Testing difficulties
  - CI failures need to be investigated to understand what is broken and who should fix it
  - Often there is more than one package causing the issue, but finding the issues is a sequential process,  
i.e., the first issue needs to be fixed before the next one is discovered
  - The responsible package developers need to be contacted
  - Consistent oversight requires more people to respond to CI failures
- Designed test plan
  - Improve xSDK-examples test suite and integrate it with the xSDK testing process
  - Evaluate and extend current xSDK CI testing through the definition and use of hierarchical test layers, addition of new platforms and increased oversight of test results



# Multi-library example codes demonstrating interoperability

- Suite of example codes has been made available in a github repository and included in the xSDK documentation. :  
<https://github.com/xsdk-project/xsdk-examples>
- The example codes are a demonstration of interoperability between xSDK libraries and provide training for xSDK library users interested in using these capabilities.
- Difficulty in building via `spack install xsdk-examples`, since new interoperabilities generally not enabled in spack and/or xSDK yet. Provide simple build via `cmake`.
- Test suite important piece of xSDK testing strategy plan



# xSDK community policies

<https://github.com/xsdk-project/xsdk-community-policies>

## Mandatory xSDK policies: must be satisfied

**M1.** Support portable installation through Spack  
(includes xSDK Spack variant guidelines)

**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 publicly available repository.

**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.** Have a debug build option.

**M17.** Provide sufficient documentation to support use and further development.

<https://x sdk.info/policies>



Version 1.0.0,  
February 2023

## Recommended xSDK policies: currently encouraged, but not required

**R1.** Provide at least one validation test that can be invoked through Spack.

**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.

**R8.** Provide version comparison preprocessor macros.

**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?

# What is performance portability?

- Discussion at 2016 DOE Center of Excellence meeting, Phoenix, AZ, USA  
(<https://performanceportability.org/perfport/definition/>)
  - Attendees included scientists, engineers for DOE's Office of Science and National Nuclear Security Agency, as well as vendors (Intel, NVIDIA, IBM, etc)
- "For the purposes of this meeting, it is the ability to run an application with acceptable performance across KNL and GPU-based systems with a single version of source code." (Rob Neely)
- "An application is performance portable if it achieves a consistent level of performance (e.g., defined by execution time or other figure of merit (not percentage of peak flops across platforms)) relative to the best-known implementation on each platform." (John Pennycook, Intel)
- "Hard portability = no code changes and no tuning. Software portability = simple code mods with no algorithmic changes. Non-portable = algorithmic changes" (Adrian Pope, Vitali Morozov)
- "(Performance portability means) the same source code will run productively on a variety of different architectures" (Larkin)
- "Code is performance portable when the application team says its performance portable!" (Richards)

# What is performance portability? (continued)

- Conclusion: There is currently no universally accepted definition of performance portability

An application is **performance portable** if it achieves a **consistent ratio** of the actual time to solution to either the best-known or the theoretical best time to solution **on each platform with minimal platform specific code required.**

BSSw, Anshu Dubey:

An application has portable performance if in addition to running on diverse platforms it exhibits **similar accuracy, stability, and reliability across these platforms for a given configuration.** Moreover, the time to solution should reflect **efficient utilization of available computational resources on each platform.**

# Portability Strategies of xSDK Libraries

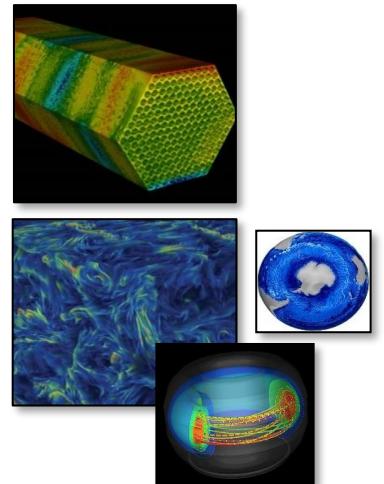
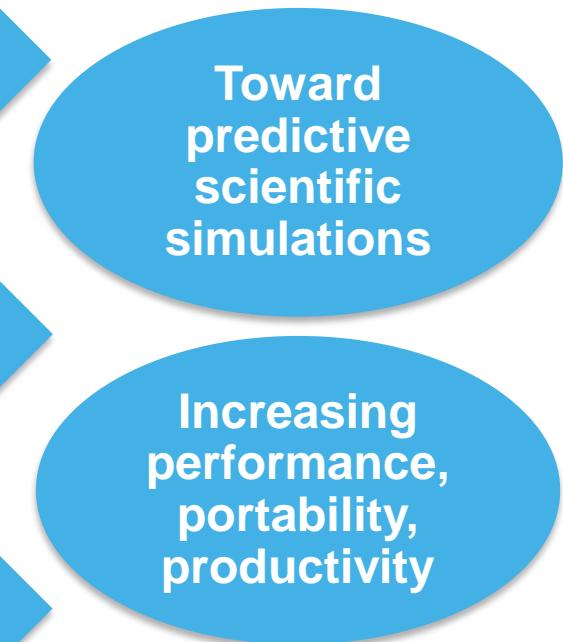
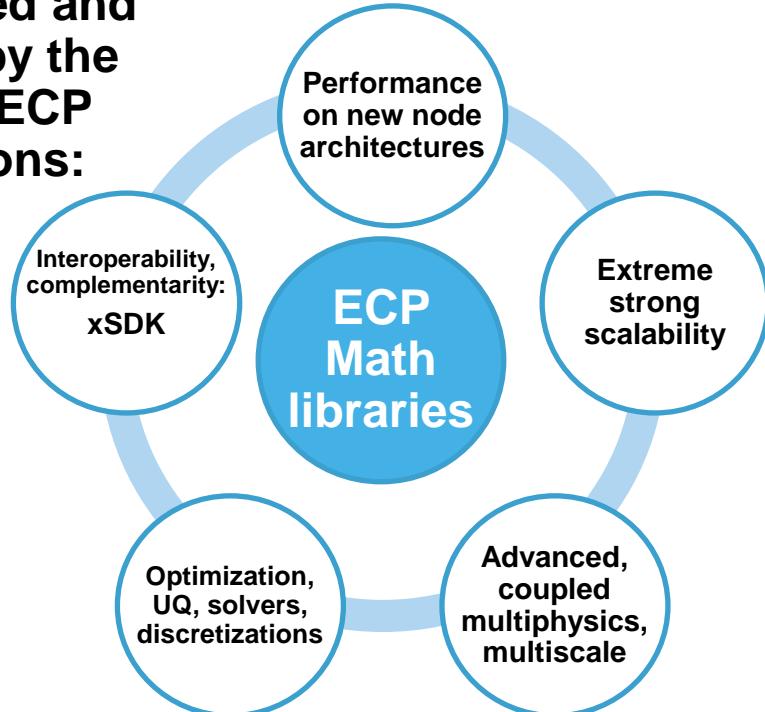
- Use of portable programming models that provide abstractions
- Use of abstraction to limit code that interacts with devices
- Use of fast kernel libraries designed for individual architectures
- Write own CUDA kernels, and use vendor provided tools to port kernels
- Develop new algorithms more suitable for GPUs  
(most challenging, but possibly best results!)



cuBLAS, cuSPARSE  
rocBLAS, rocSPARSE  
MKL

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

As motivated and validated by the needs of ECP applications:



Timeline:

xSDK  
release 1

xSDK  
release 2

· · · ·

xSDK  
release n

# Extreme-scale Scientific Software Stack (E4S)

<https://e4s.io>



A screenshot of the E4S website homepage. The header features the "E4S" logo in white on a dark green bar. Below it is a navigation menu with links: HOME, EVENTS, ABOUT, PRODUCT DOCPORTAL, COMMUNITY POLICIES, CONTACT, JOIN, FAQ, and DOCUMENTATION. A prominent orange "DOWNLOAD" button is located below the menu. The main content area has a dark green background with a diagonal black stripe. The text "E4S Project" is centered above the title "The Extreme-scale Scientific Software Stack".

- E4S is a community effort to provide open-source software packages for developing, deploying, and running scientific applications on HPC platforms.
- E4S provides containers and turn-key, from-source builds of 100+ popular HPC, EDA (e.g., Xyce), and AI/ML packages (e.g., TensorFlow, PyTorch)
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products
- E4S Spack build cache has over 100,000 binaries.
- Platforms: x86\_64, ppc64le, and aarch64. GPUs runtimes: NVIDIA (CUDA), AMD (ROCm), and Intel (OneAPI)
- E4S DocPortal provide a single online location for *accurate* product descriptions for software products.
- E4S helps applications reduce the burden to install dependencies
- Quarterly releases: E4S 23.05 released on May 31, 2023: [https://e4s.io/talks/E4S\\_23.05.pdf](https://e4s.io/talks/E4S_23.05.pdf)

# E4S Summary



## What E4S is not

A closed system taking contributions only from DOE software development teams.

A monolithic, take-it-or-leave-it software behemoth.

A commercial product.

A simple packaging of existing software.

## What E4S is

Extensible, open architecture software ecosystem accepting contributions from US and international teams.  
Framework for collaborative open-source product integration.

A full collection if compatible software capabilities **and**  
A manifest of a la carte selectable software capabilities.

---

Vehicle for delivering high-quality reusable software products in collaboration with others.

---

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

- [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: 2021 highlights](#), Rinku Gupta
- And many more ...

The screenshot shows the homepage of the Better Scientific Software (BSSw) website. At the top right, there is a large red link to the website: <https://bssw.io>. The main header features the BSSw logo (a stylized blue 'B' and 'S') and the text "better scientific software". Below the header, there is a navigation bar with links for "Information For", "Contribute to BSSw", "Receive Our Email Digest", and "Contact BSSw". The main content area has a dark blue background with white text. It features the title "Better Scientific Software (BSSw)" and a brief description: "Software—the foundation of discovery in computational science & engineering—faces increasing complexity in computational models and computer architectures. BSSw provides a central hub for the community to address pressing challenges in software productivity, quality, and sustainability." There are also several buttons for "GET ORIENTED", "Communities Overview", "Intro to CSE", and "Intro to HPC". At the bottom left, there is a callout box with the text "See also Track 7: Software Productivity & Sustainability (Aug 11)".

# HandsOn Lessons

- Structured meshing & discretization
- Unstructured meshing & discretization
- Krylov solvers & preconditioners
- Sparse direct solvers
- Nonlinear solvers
- Time integration
- Numerical optimization



## ATPESC 2023 Hands On Lessons

<a href="#">Meshing and Discretization with AMReX</a>	A Block Structured Adaptive Mesh Refinement Framework
<a href="#">Unstructured Meshing &amp; Discretization with MFEM</a>	Finite Elements and Convergence
<a href="#">Krylov Solvers and Algebraic Multigrid with hypre</a>	Demonstrate utility of multigrid
<a href="#">Iterative Solvers &amp; Algebraic Multigrid (with Trilinos, Belos &amp; MueLu)</a>	Introduction to Krylov Solvers and Preconditioning, with emphasis on Multigrid
<a href="#">Sparse, Direct Solvers with SuperLU</a>	Role and Use of Direct Solvers in Ill-Conditioned Problems
<a href="#">Rank Structured Solvers with STRUMPACK</a>	Using STRUMPACK for dense and sparse linear systems
<a href="#">Nonlinear Solvers with PETSc</a>	Introduction to Nonlinear Solvers: Newton, Krylov

And more ...

Github pages site:

<https://xSDK-project.github.io/MathPackagesTraining2023/lessons/>

# If you haven't yet done so, please choose which session you plan to attend!

Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Ann Almgren, Andrew Myers	Unstructured Discretization (MFEM/PUMI) – Tzanio Kolev, Mark Shephard, Cameron Smith
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Sarah Osborn, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Pieter Ghysels
12:30 – 1:30	Lunch, SME Selections, Panel Questions	
1:30 – 2:45	Nonlinear Solvers (PETSc) – Richard Mills	Time Integration (SUNDIALS) – David Gardner
2:45 – 3:15	Break, SME Selections, <b>Panel Questions Due</b>	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:30	Wrap-up (Ann Almgren) / Panel: Extreme-Scale Numerical Algorithms and Software	
5:30 – 6:30	Unstructured Time: <b>SME Selections Due</b> , Informal Discussion, Continue Hands-on	
6:30 – 7:30	Dinner	
7:30 – 9:30	Optional Activity: SME Speed-dating	

# Choose which lecture you want to attend!

Access form here

The screenshot shows a Google Forms survey titled "ATPESC numerical track selections". The survey instructions ask participants to denote sessions they plan to attend. It includes a header with participant information (yang11@llnl.gov) and a note about required questions. The first question, "Parallel session One \*", has two options: "Structured Meshes (with AMReX)" and "Unstructured Meshes (with MFEM/PUMI)".

ATPESC numerical track selections

Participants, please denote sessions you plan to attend so that needed room size can be determined

yang11@llnl.gov [Switch account](#)

\* Indicates required question

Email \*

Your email

Parallel session One \*

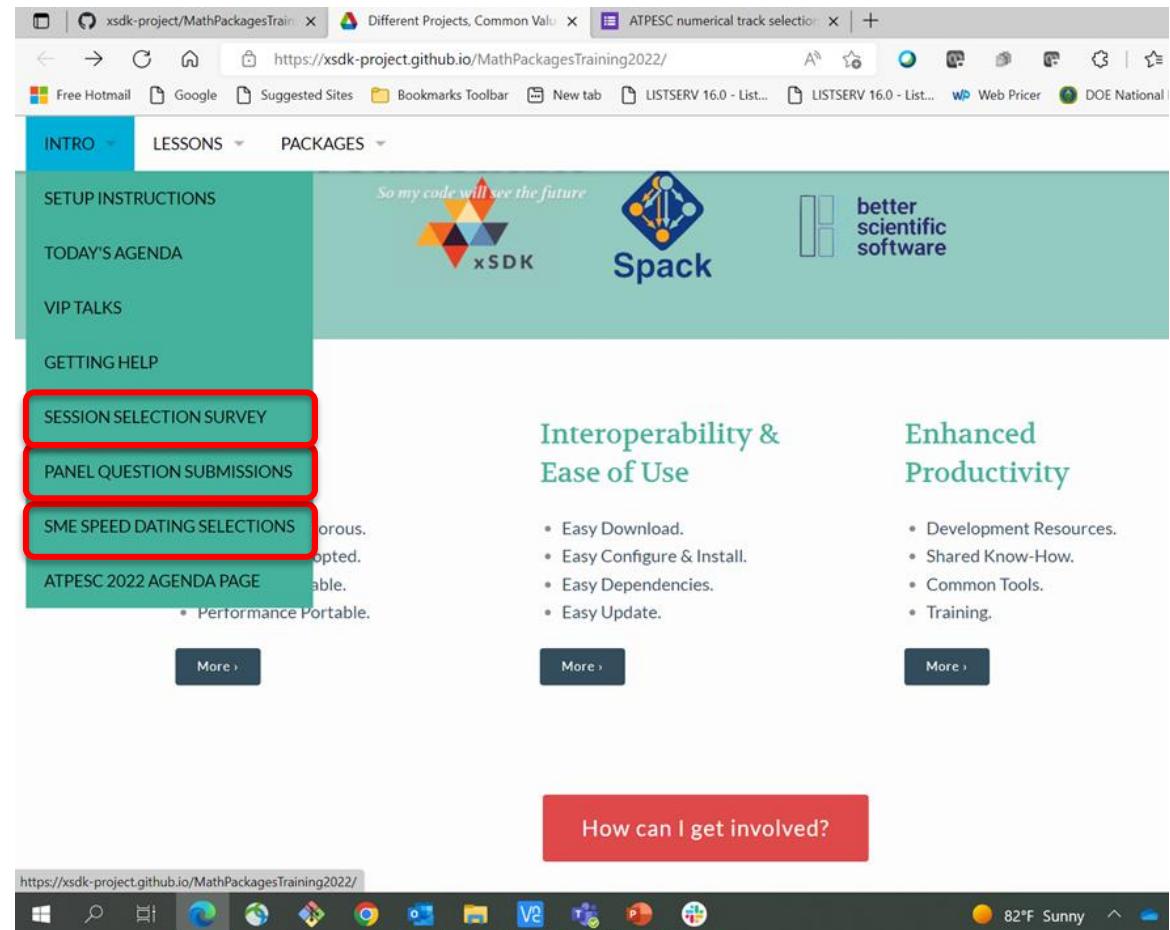
Structured Meshes (with AMReX)

Unstructured Meshes (with MFEM/PUMI)

# Next steps

- If you haven't done so
  - Choose which session you will attend!

- During breaks and lunch
  - Submit questions for panelists (optional)
  - Sign up for discussions with numerical software developers (optional)
    - Your email address
    - Complete by 3:30 pm CDT



# 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!**



# CASC

Center for Applied  
Scientific Computing



# Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. LLNL-PRES-852746

This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC) program, and by the Exascale Computing Project, a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

#### **Disclaimer**

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.