Introduction to Numerical Software

Presented to
ATPESC 2025 Participants

Ulrike Meier Yang
Lawrence Livermore National Laboratory

Date 08/05/2025

U.S. DEPARTMENT OF ENERGY | Office of Science

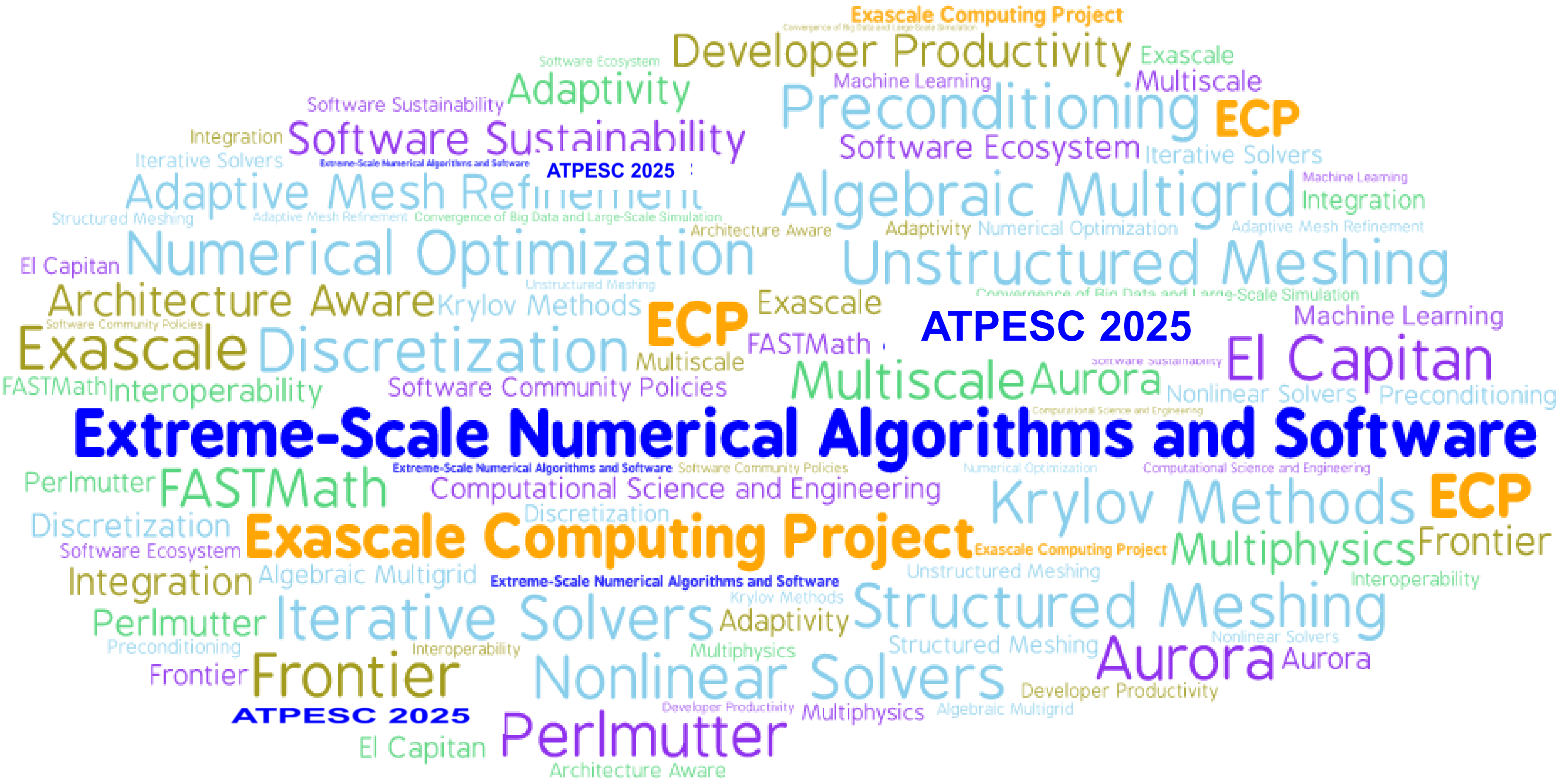
ATPESC Numerical Software Track

ATPESC 2025
EXTREME-SCALE COMPUTING

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

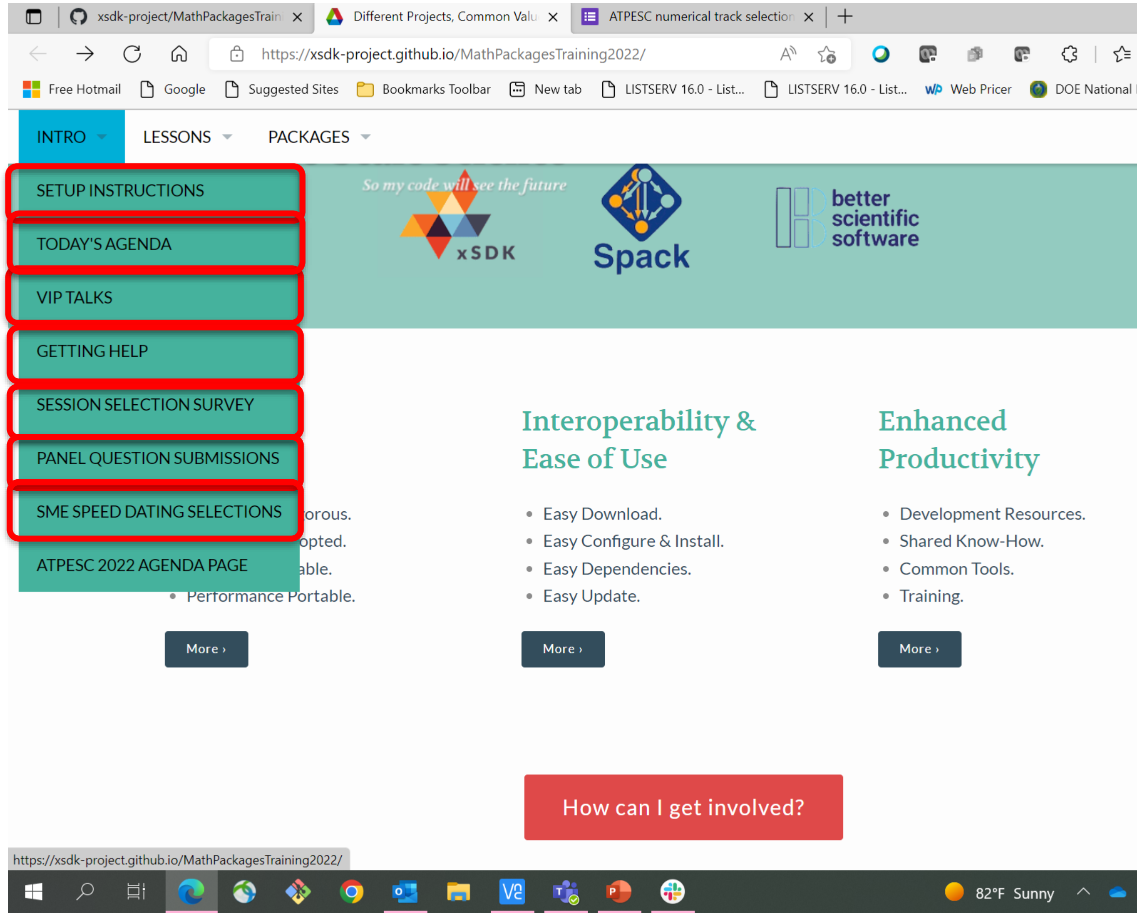


ATPESC 2025
EXTREME-SCALE COMPUTING

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-2025/#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/MathPackagesTraining2025/agenda/

https://xsdk-project.github.io/MathPackagesTraining2025/



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

Agenda

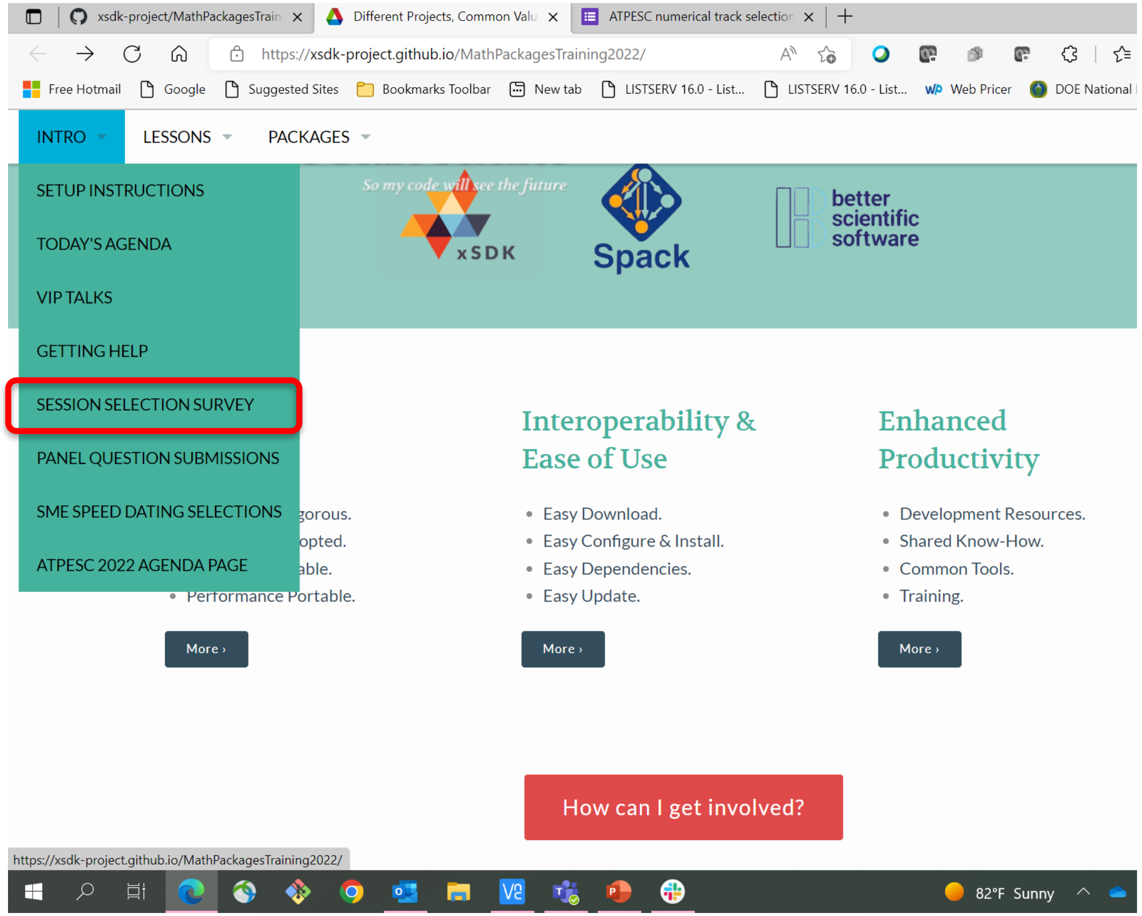
https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

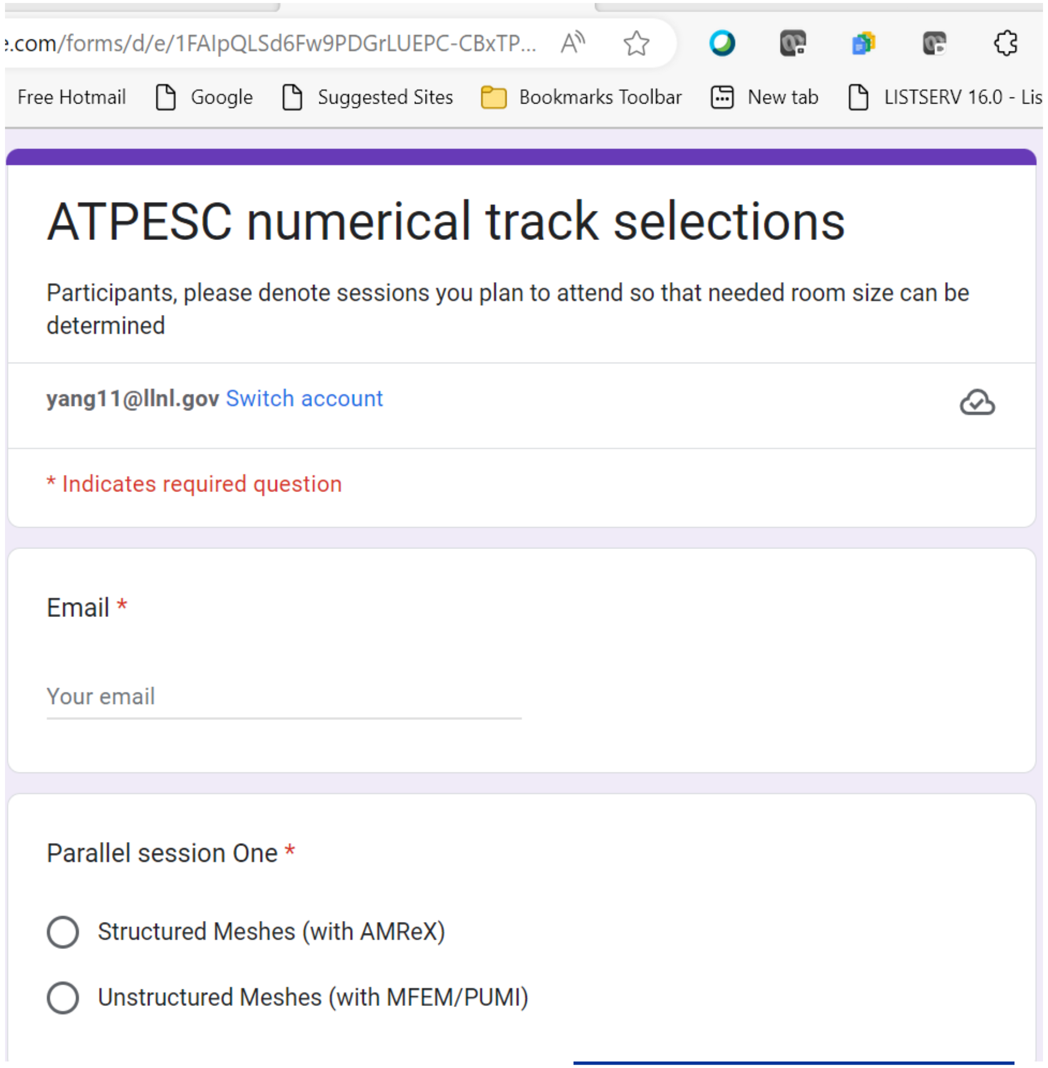
Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

Choose which lecture you want to attend!

Access: ATPESC numerical track selections

https://xsdk-project.github.io/MathPackagesTraining2025/





Agenda

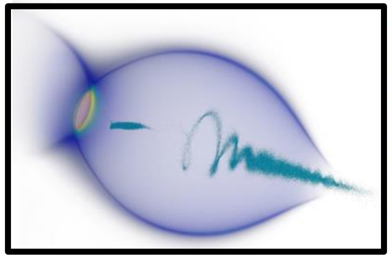
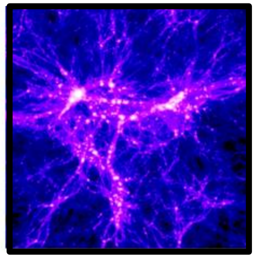
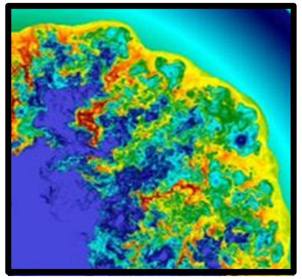
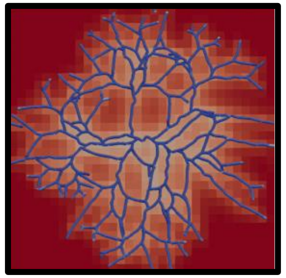
https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	**Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang**	**Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell**
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

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

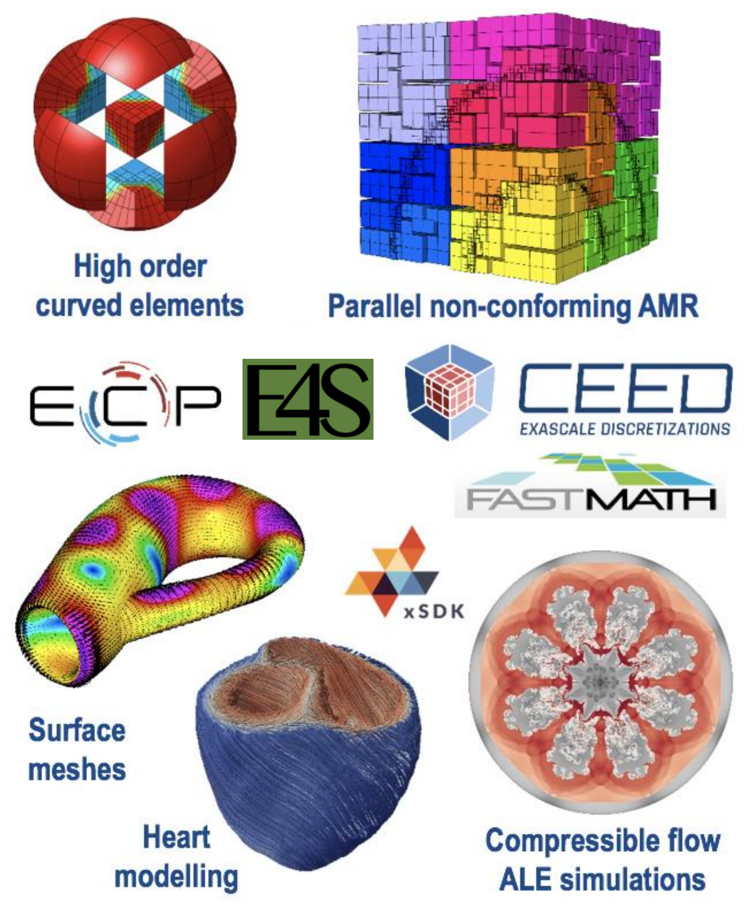
https://www.github.com/AMReX-Codes/amrex

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**
  - BSD with thousands of downloads/year worldwide.
  - Available on GitHub, also via OpenHPC, Spack. Part of ECP's CEED co-design center.



https://mfem.org

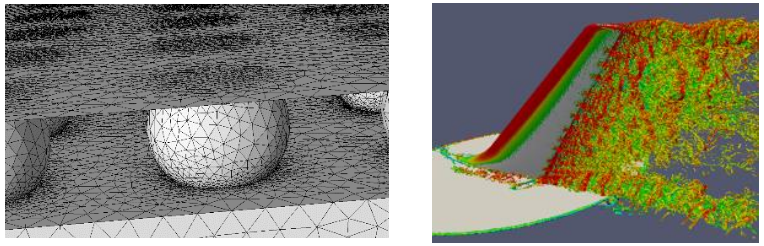
Parallel Unstructured Mesh Infrastructure

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

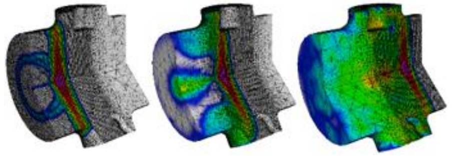
- **Core functionality**
  - Distributed, conformant mesh with entity migration, remote read only copies, fields and their operations
  - Link to the geometry and attributes
  - Mesh adaptation (straight and curved), mesh motion
  - Multi-criteria partition improvement
  - Distributed mesh support for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

PUMI

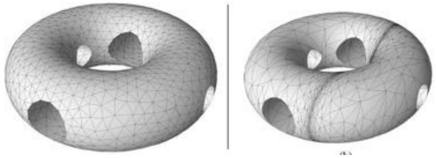
SCOREC Rensselaer



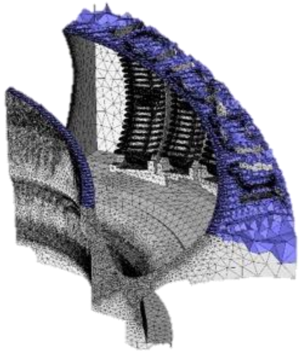
Applications with billions of elements: flip-chip (L), flow control (R)



Mesh adaptation for evolving features



Anisotropic adaptation for curved meshes



RF antenna and plasma surface in vessel.

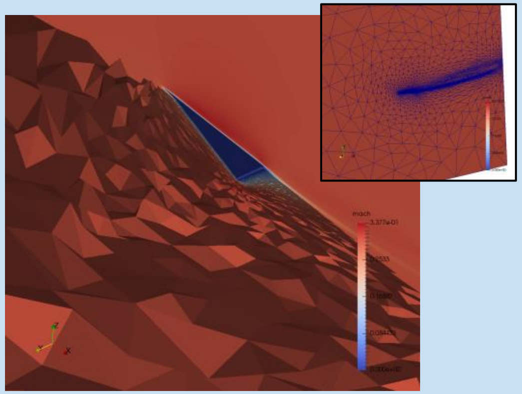
Source Code: github.com/SCOREC/core
User Guide: doi.org/10.5281/zenodo.16412599
Paper: doi.org/10.1145/2814935

Omega_h Parallel Unstructured Mesh Adaptation on GPUs

Parallel adaptation of unstructured meshes on GPUs.
Support the development of unstructured mesh simulation workflows on leadership systems.

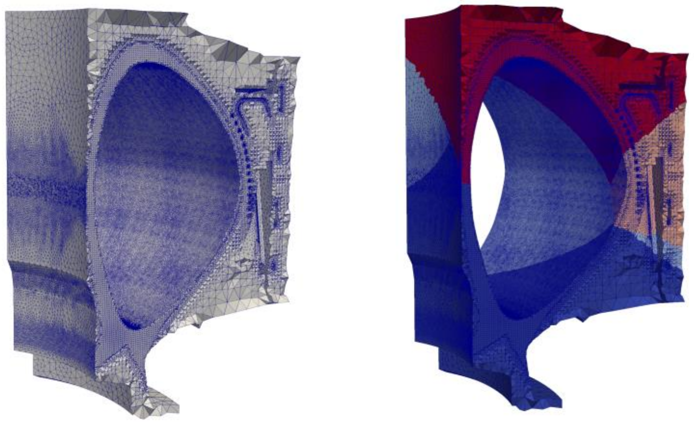
- **Core functionality**
  - Distributed, conformant mesh adaptation (coarsening past initial mesh + refinement)
  - Manycore and GPU parallelism using Kokkos
  - Runs on NVIDIA, AMD, and Intel GPUs
  - Supports complex geometric models via Gmsh and Simmetrix SimModSuite

- **Applications Supported**
  - GITRm: impurity transport
  - XGCm: core+edge fusion plasma physics
  - MFEM: finite element framework specializing in high-order methods
  - PetraM: RF Fusion
  - MALI: land ice melting

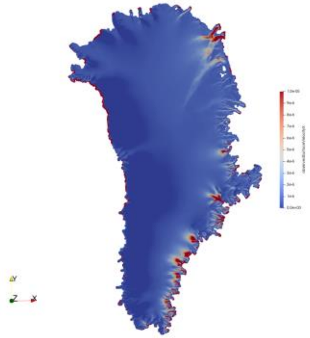


'Crinkle clip' view of wing's top surface after adaptation (main) and wake on symmetry surface (inset).

SCOREC
Rensselaer



Serial and RIB partitioned mesh of RF antenna and vessel model.



Omega_h mesh of the Greenland ice sheet, colored by velocity.

Source Code: github.com/SCOREC/omega_h
Thesis: hdl.handle.net/20.500.13015/1817

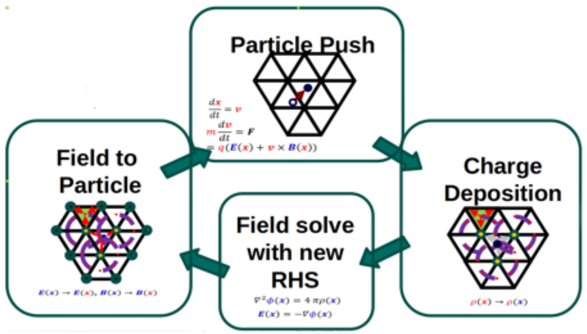
PUMIPic Parallel Unstructured Mesh Infrastructure for Particle-in-Cell

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

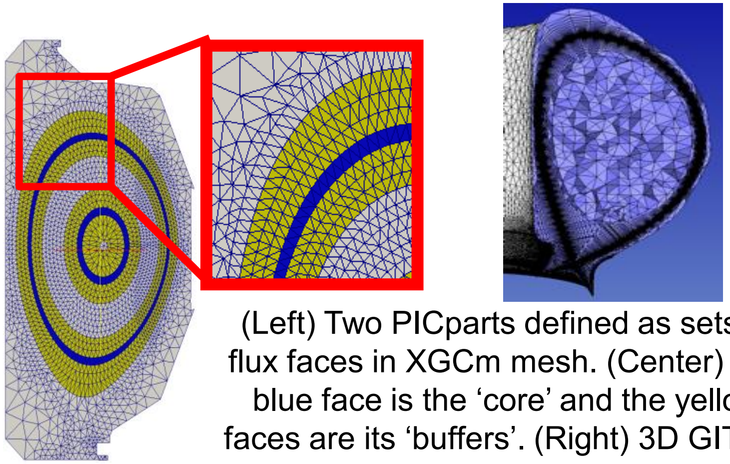
- **Core functionality**
  - Unstructured mesh-based approach
    - Particles accessed through mesh
    - Particle search through mesh adjacencies
    - Effective coupling to PDE solvers
    - Partitioning using bounding flux surfaces, graph, or geometric methods
    - PICpart: owned elements + copied elements from topologically or spatially neighboring processes
    - Stored on GPU using Omega_h library: *github.com/SNLComputation/omega_h*
  - Particles
    - Supports multiple species
    - Particle storage choices: Sell-C-Sigma [Kreutzer 2014], COPA Cabana, and DPS.
    - DPS storage optimized for applications with tens of particles per mesh element.
  - Parallel kernel launch function abstracts underlying particle and mesh storage
  - Supports NVIDIA and AMD GPUs

- **Applications Supported**
  - GITRm: impurity transport
  - XGCm: core+edge fusion plasma physics
  - PolyMPO: GPU polygonal mesh-based material point operations
  - PUMI-Tally: supports Monte Carlo neutral particle transport
  - 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

Rensselaer SCOREC



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) 3D GITRm mesh for impurity transport simulation.

Source Code: github.com/SCOREC/pumi-pic
Paper: doi.org/10.1016/j.jpdc.2021.06.004

Agenda

https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

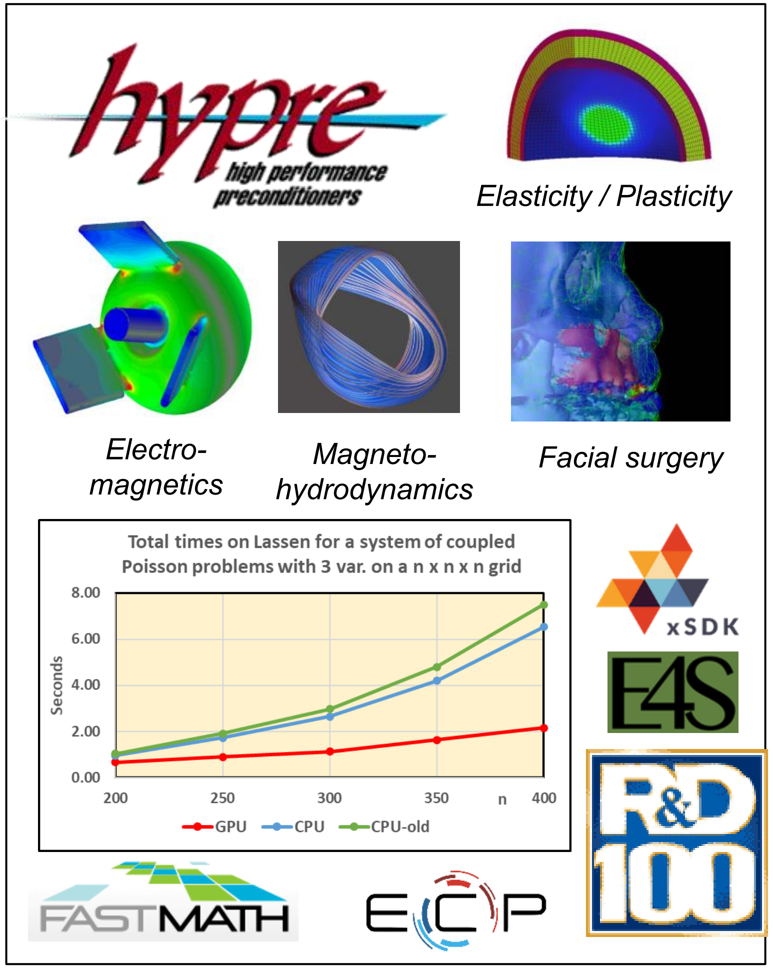
Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	**Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang**	**Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu**
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

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 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



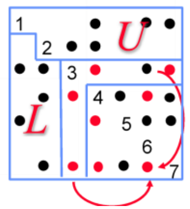
http://www.llnl.gov/CASC/hypre

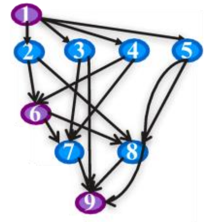
ATPESC 2025
EXTREME-SCALE COMPUTING

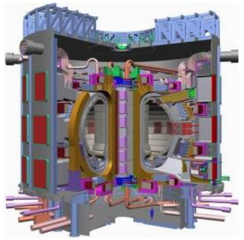
SuperLU

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

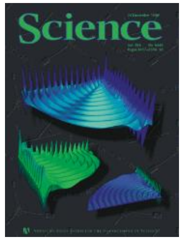
- **Capabilities**
  - Serial (thread-safe), shared-memory (SuperLU_MT, OpenMP or Pthreads), distributed-memory (SuperLU_DIST, hybrid MPI+ OpenM + CUDA/HIP).
    - Written in C, with Fortran interface
  - Sparse LU decomposition (can be nonsymmetric sparsity pattern), triangular solution with multiple right-hand sides
  - Incomplete LU (ILUTP) preconditioner in serial SuperLU
  - Sparsity-preserving ordering: minimum degree or graph partitioning applied to A^TA 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

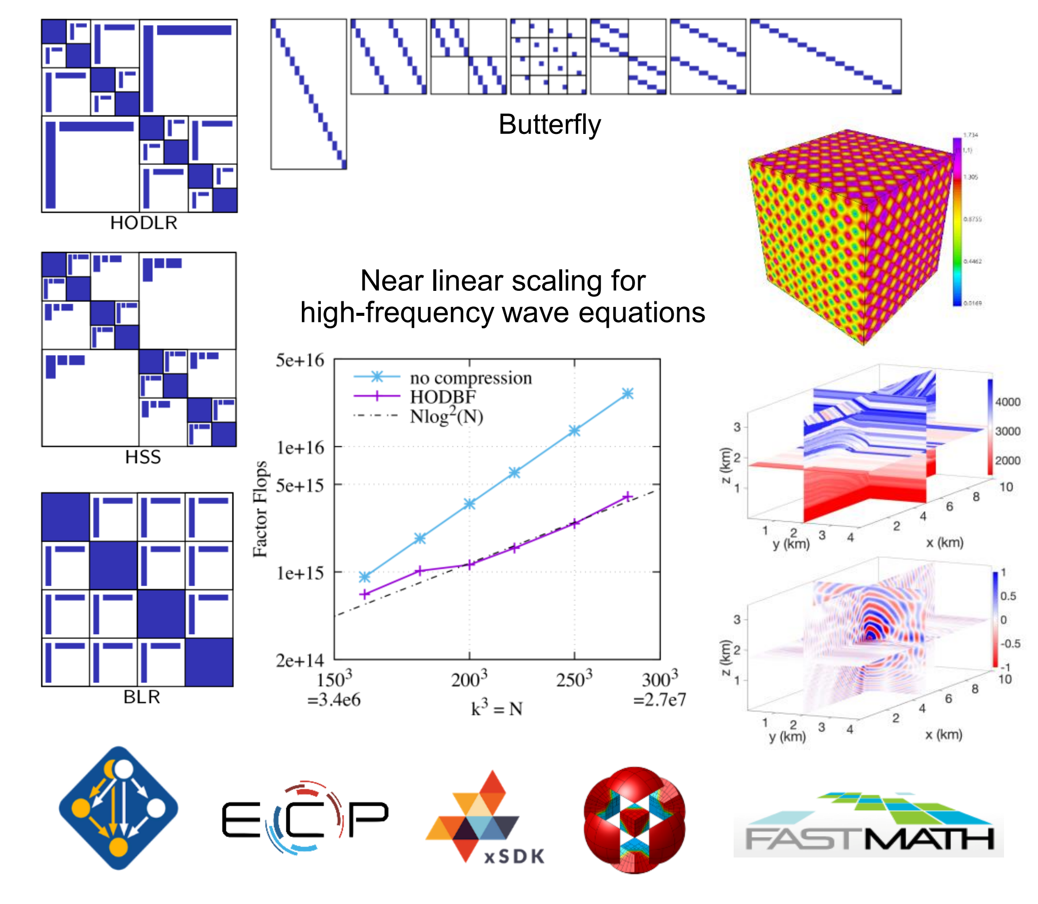
Widely used in commercial software, including AMD (circuit simulation), Boeing (aircraft design), Chevron, ExxonMobile (geology), Cray's LibSci, FEMLAB, HP's MathLib, IMSL, NAG, SciPy, OptimaNumerics, Walt Disney Animation.

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
- **Sparse Direct Solver**
  - Algebraic sparse direct solver
  - GPU: CUDA, HIP/ROCm, DPC++ (in progress)
  - Orderings: (Par)METIS, (PT)Scotch, RCM
- **Preconditioners**
  - Approximate sparse factorization, using hierarchical matrix approximations
  - Scalable and robust, aimed at PDE discretizations, indefinite systems, …
  - Iterative solvers: GMRES, BiCGStab, iterative refinement
- **Software**
  - BSD license
  - Interfaces from PETSc, MFEM, Trilinos, available in Spack



github.com/pghysels/STRUMPACK

Agenda

https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

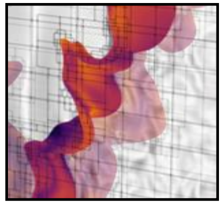
SUNDIALS

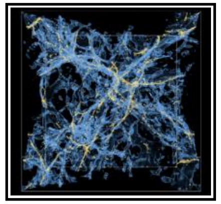
Suite of Nonlinear and Differential /Algebraic Equation Solvers

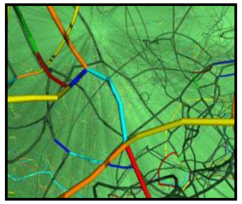
Adaptive time integrators for ODEs and DAEs and efficient nonlinear solvers
Used in a variety of applications. Freely available. Encapsulated solvers & parallelism.

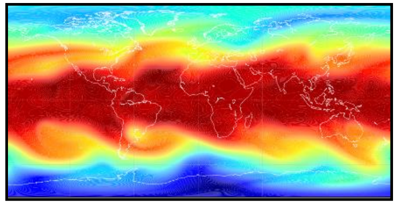
- **ODE and DAE time integrators:**
  - *CVODE:* adaptive order and step BDF (stiff) & Adams (non-stiff) methods for ODEs
  - *ARKODE:* adaptive step implicit, explicit, IMEX, and multirate Runge-Kutta methods for ODEs
  - *IDA*: adaptive order and step BDF methods for DAEs
  - *CVODES* and *IDAS:* provide 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

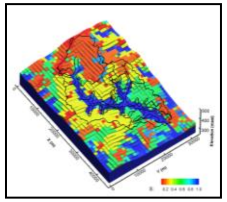
SUNDIALS is used worldwide in applications throughout research and industry


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

PETSc TAO

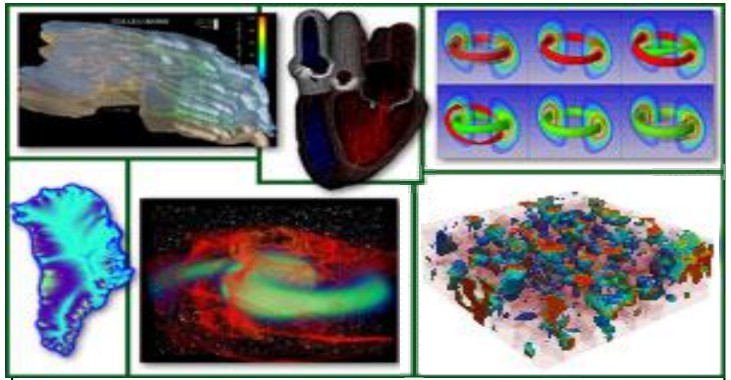
Portable, Extensible Toolkit for Scientific Computation / Toolkit for Advanced

Scalable algebraic solvers for PDEs. Encapsulate parallelism in high-level objects. Active & supported 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 <u>at runtime</u>**
  - 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

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

ECP xSDK FASTMATH R&D 100

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

Agenda

https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

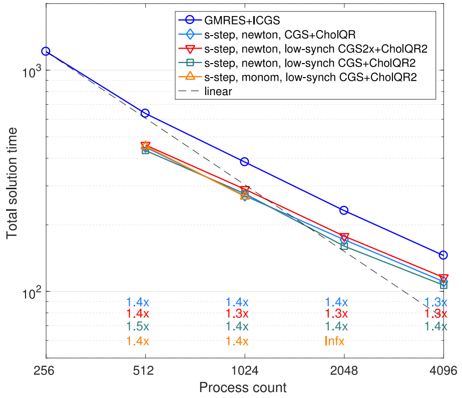
Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	**Optimization (TAO) – Toby Isaac**	**Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper**
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

Trilinos/Belos

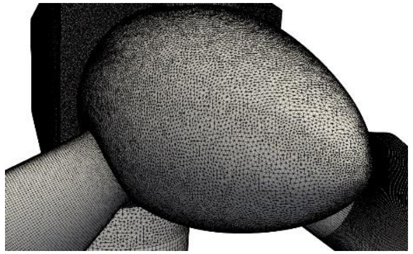
Iterative Krylov-based solvers. Templated C++ allows for generic scalar, ordinal, and compute node types.

- **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

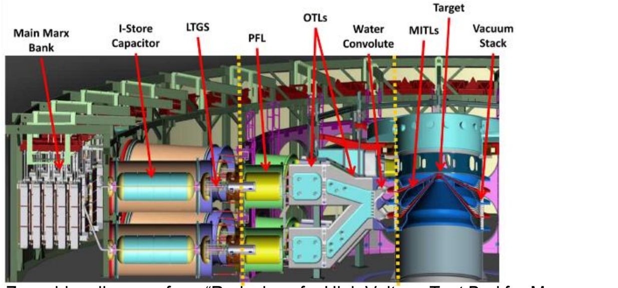
Trilinos/MueLu

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)

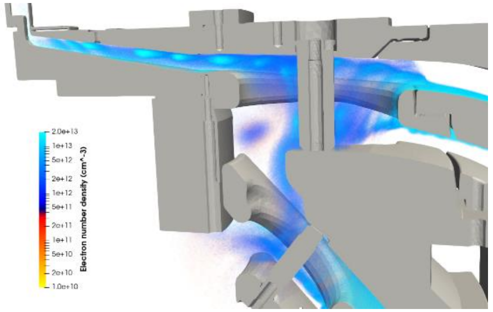
- **Capabilities**
  - — Aggregation-based coarsening
  - — Smoothers: Jacobi, GS, ℓ_1 GS, polynomial, ILU, sparse direct
  - — Load-balancing for good parallel performance
  - — Structured coarsening, geometric multigrid
  - — Setup and solve phases can run on GPUs.
  - — Performance portability via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

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



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

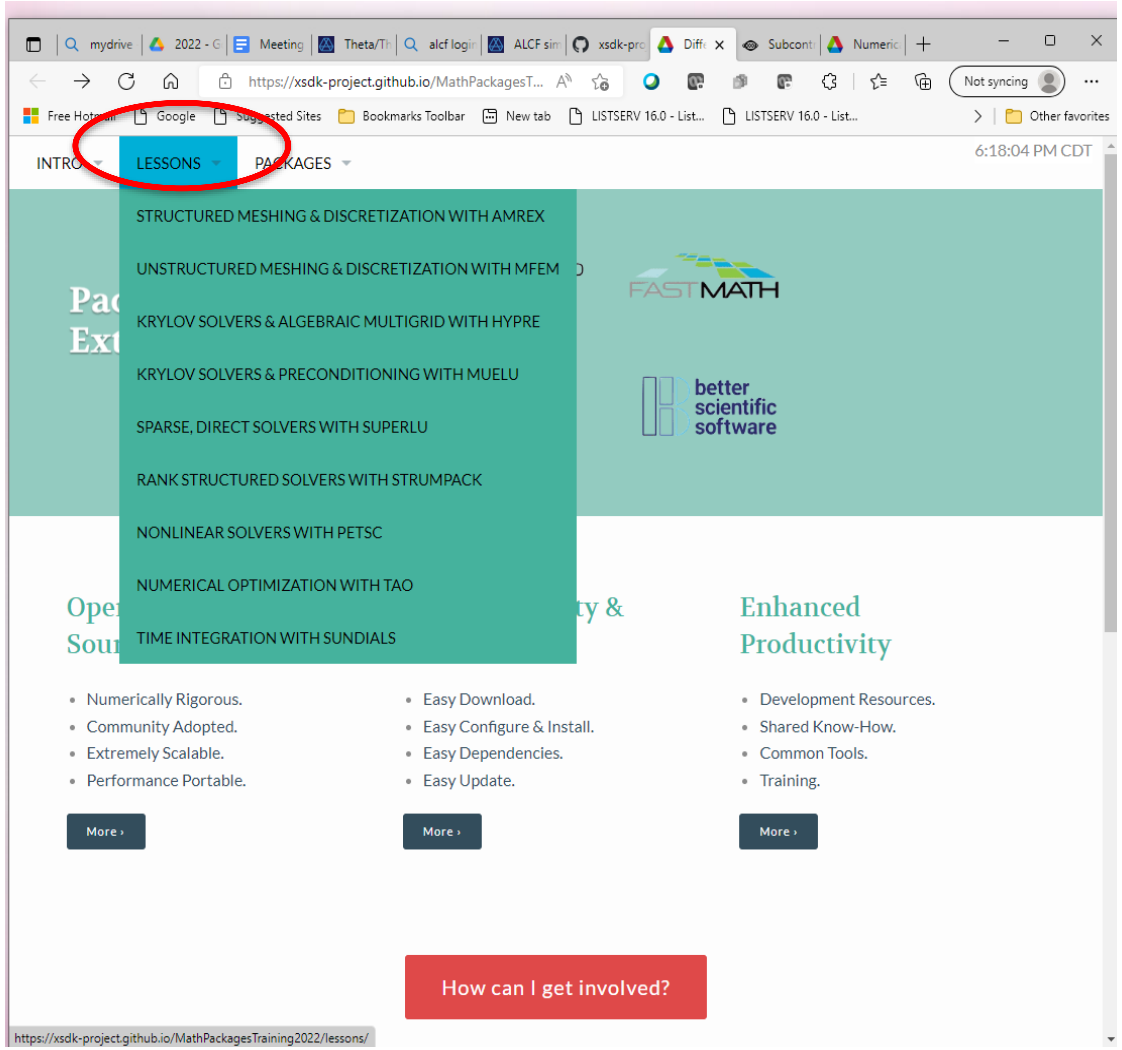
AMG preconditioning for H(curl) systems is key enabling technology in Z machine simulations for determining power from Marx banks to Target.



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

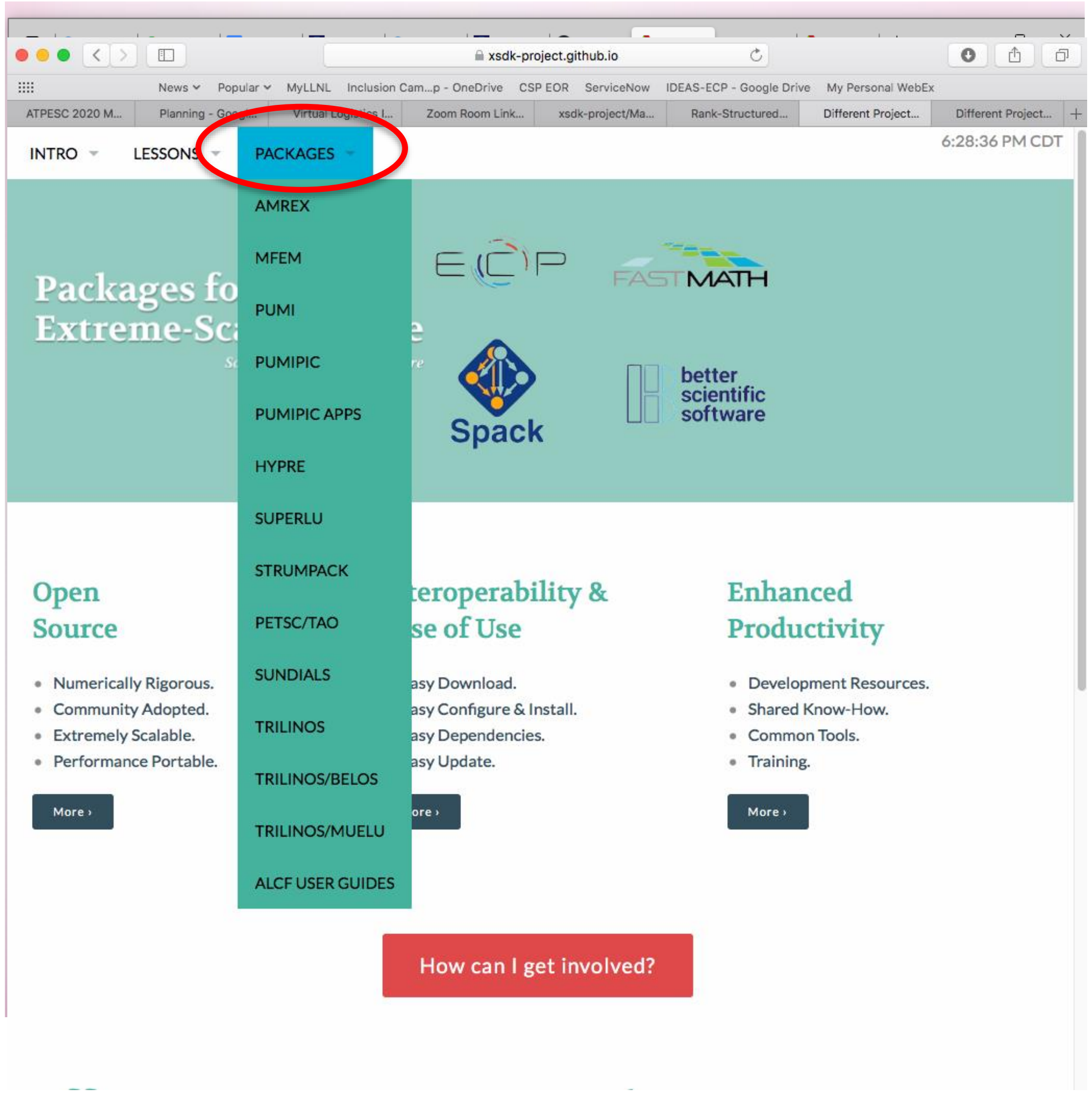
https://trilinos.github.io/muelu.html

https://xsdk-project.github.io/MathPackagesTraining2025/



- Hands-on Lessons

https://xsdk-project.github.io/MathPackagesTraining2025/



- Hands-on Lessons
- Packages

ATPESC 2025
EXTREME-SCALE COMPUTING

Agenda

https://extremecomputingtraining.anl.gov/agenda-2025/#Track-5

Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	**Panel: Extreme-Scale Numerical Algorithms and Software**	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

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 Question Submission Form
- Panelists


Andrew Myers, LBL

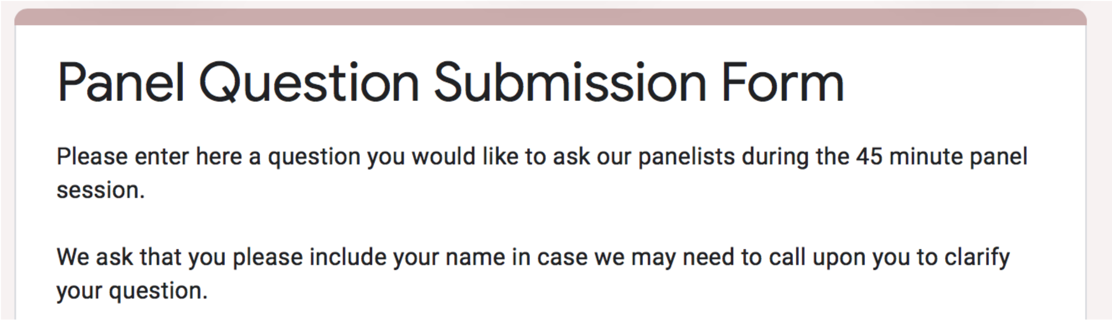

Ulrike Yang, LLNL


Mark Stowell, LLNL


Yang Liu, LBL

- Moderator


Toby Isaac, Nvidia


Panel Question Submission Form

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

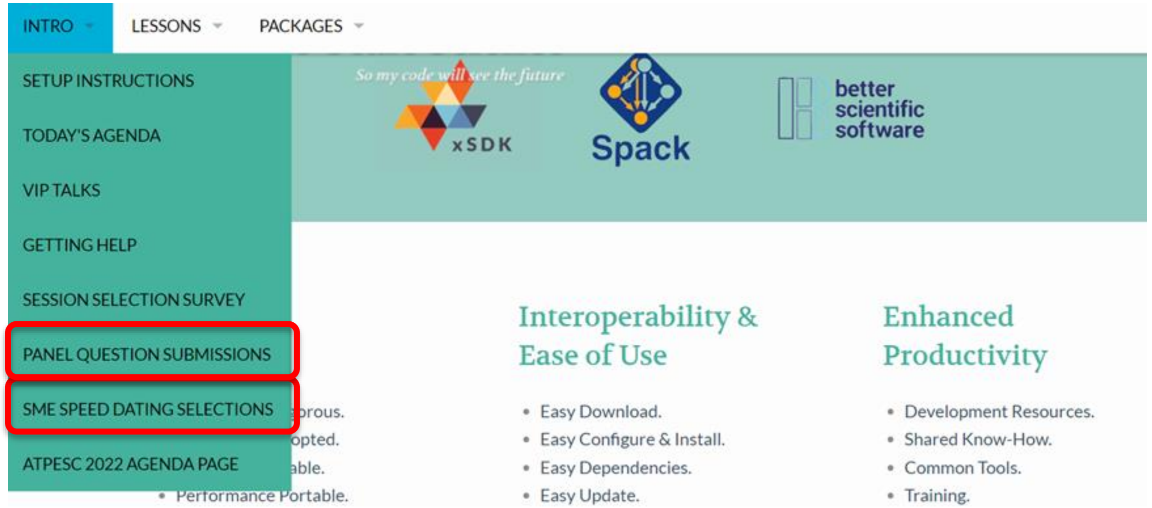
We ask that you please include your name in case we may need to call upon you to clarify your question.

ATPESC 2025
EXTREME-SCALE COMPUTING

Next steps: https://xsdk-project.github.io/MathPackagesTraining2025/agenda

- **Panel: Main Room @ 4:30 pm**
- **SME Speed Dating: @5:15pm**
- **During breaks and lunch**
  - **Provide Panel Questions**
    Due: 4:30 pm
  - **Sign up for discussions with numerical software developers** (optional)
    - Your email address

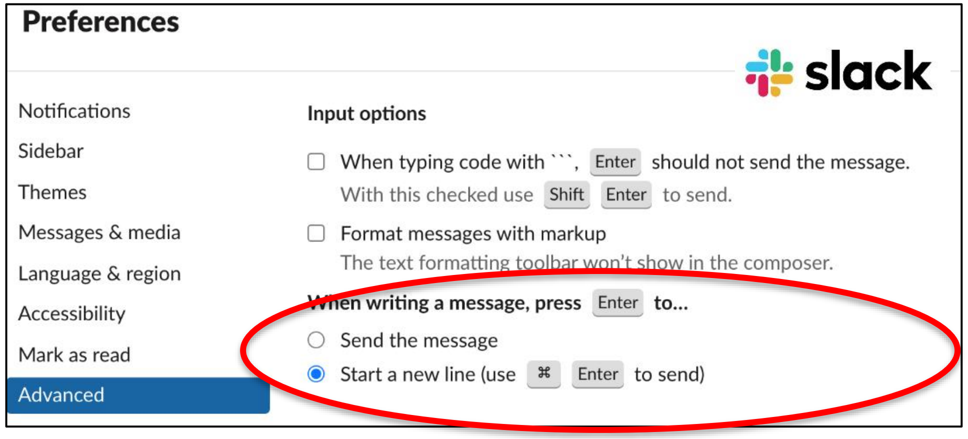
    Due 5:00 pm

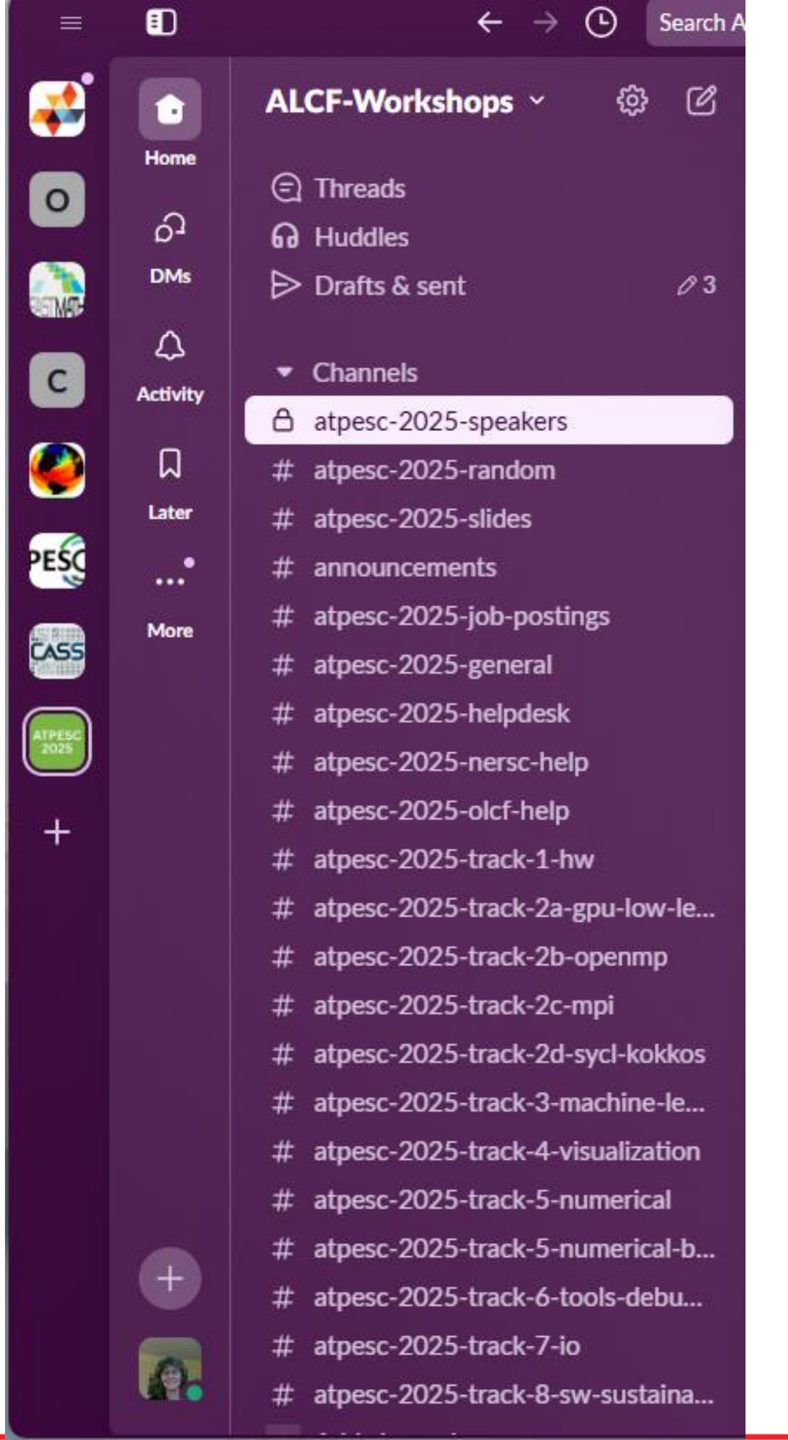


Using Slack

- Recommend using the desktop app, but browser ok too
- **# atpesc-2025-track-5-numerical** channel
- **# atpesc-2025-helpdesk** 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

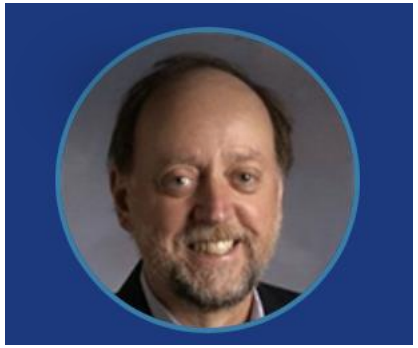
Tip: Consider setting Preferences to customize when to send





VIPs of ATPESC Extreme-Scale Numerical Software Track

- **Paul Fisher, Argonne National Laboratory** [bio]
  - **Scaling Computational Fluid Dynamics to Exascale and Beyond**
    - ATPESC 2025, Monday, July 28, 7:30pm



- **Jack Dongarra, Univ of Tennessee** [bio]
  - **Growing up at Argonne National Laboratory**
    - ATPESC 2025, Monday, August 4, 7:30pm
  - Adaptive Linear Solvers and Eigensolvers, ATPESC 2019 [video]

- **Jim Demmel, UC Berkeley** [bio]
  - **Communication-Avoiding Algorithms for Linear Algebra, Machine Learning and Beyond**
    - ATPESC 2025, Wednesday, August 5, 7:30pm
  - ENLA Seminar, June 2020 [video]

The ATPESC Team 2025

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

Toby Isaac, Nvidia

Andrew Myers, LBL

Richard Mills, ANL

Mark Shephard, RPI

Cameron Smith, RPI

Ulrike Yang, LLNL

Christian Glusa, SNL

Graham Harper, SNL

Sherry Li, LBL

Mark Stowell, LLNL

Satish Balay, ANL

Daniel Osei-Kuffuor, LLNL

David Gardner, LLNL

Weiqun Zhang, LBL

Yang Liu, LBL

ATPESC 2025
EXTREME-SCALE COMPUTING

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?

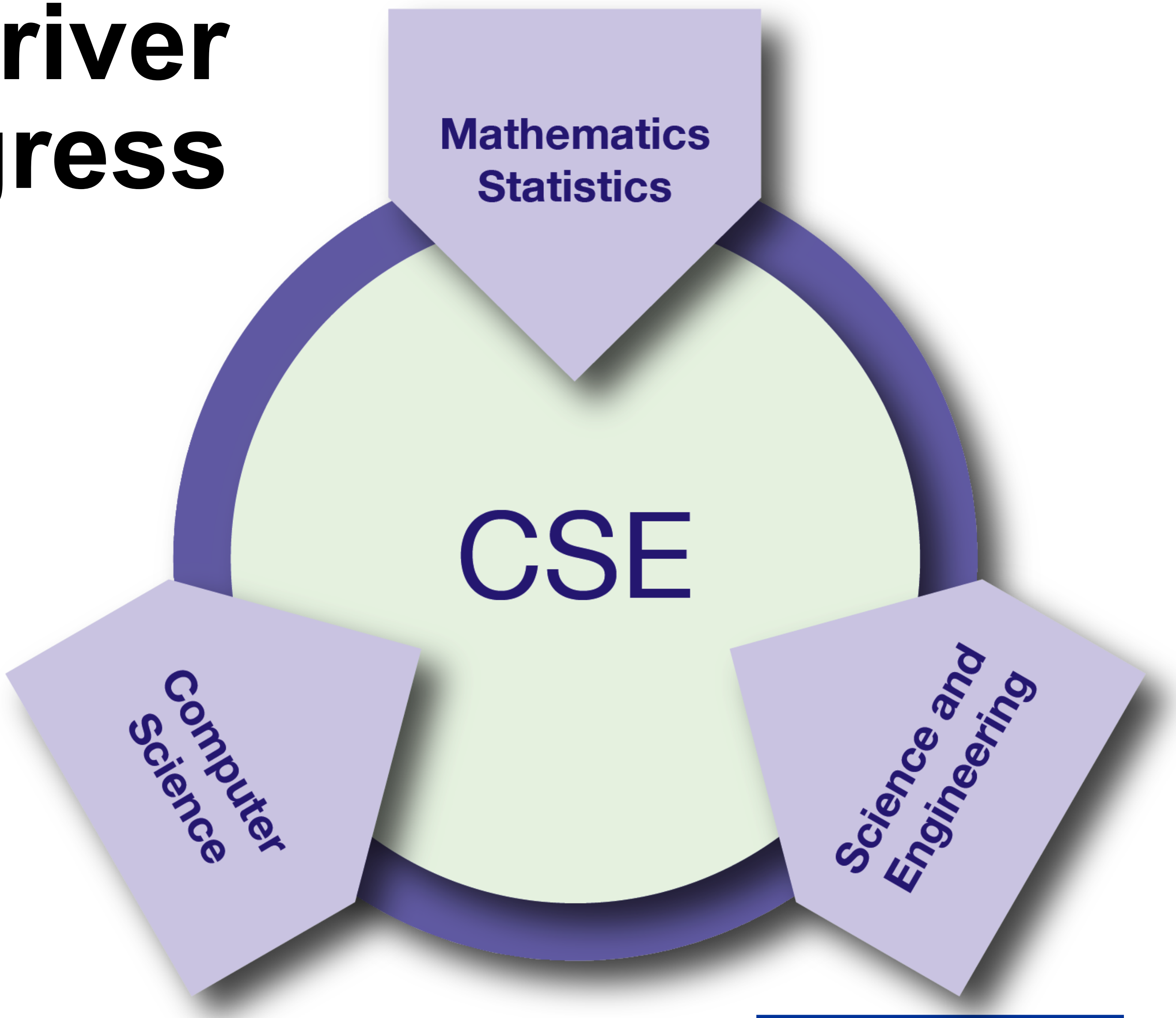
- Libraries enable users to focus on their primary interests
  - Reuse algorithms and data structures developed by experts
  - Customize and extend to exploit application-specific knowledge
  - Cope with complexity and changes over time
- More efficient, robust, reliable, scalable, sustainable scientific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

CSE: Essential driver of scientific progress

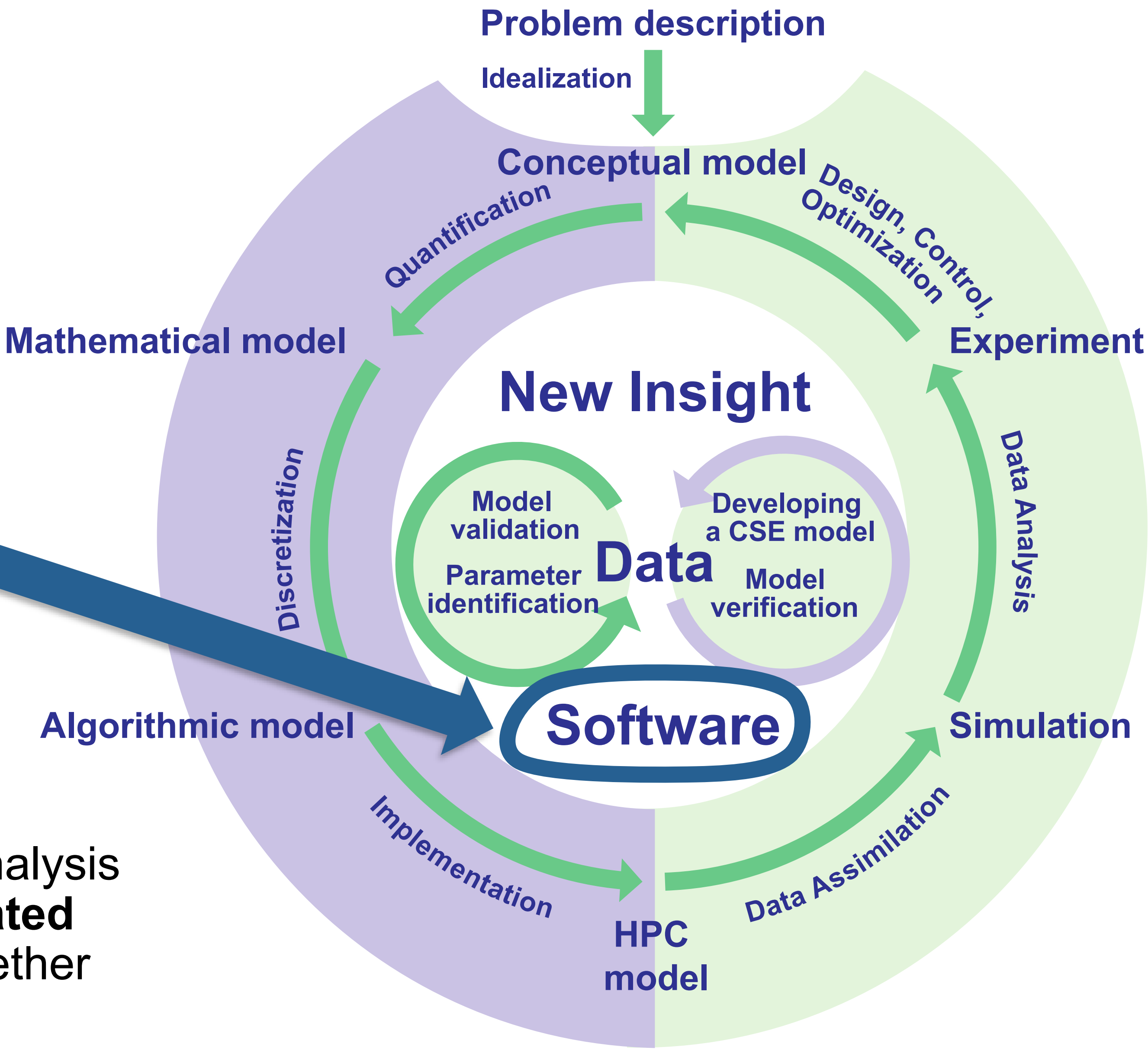
CSE = Computational Science & Engineering

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



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

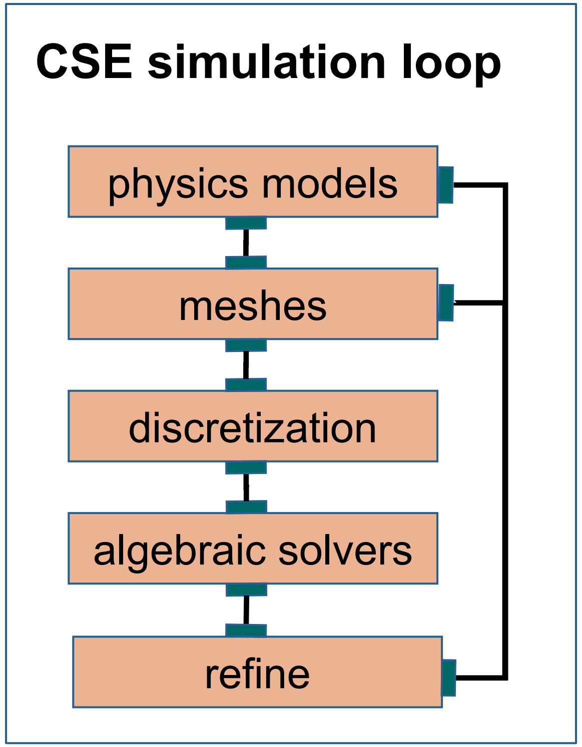


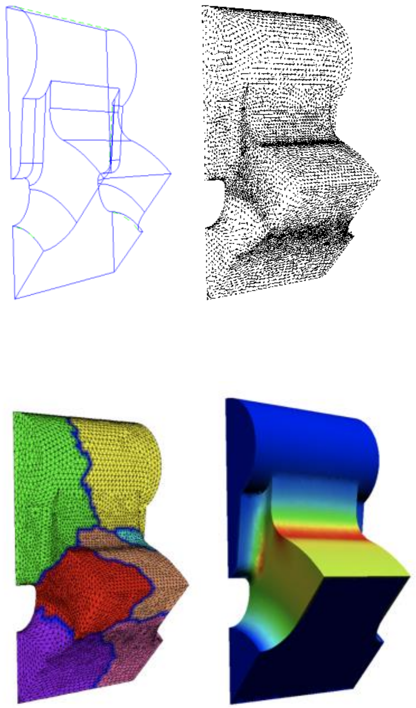
CSE cycle: Modeling, simulation, and analysis

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

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

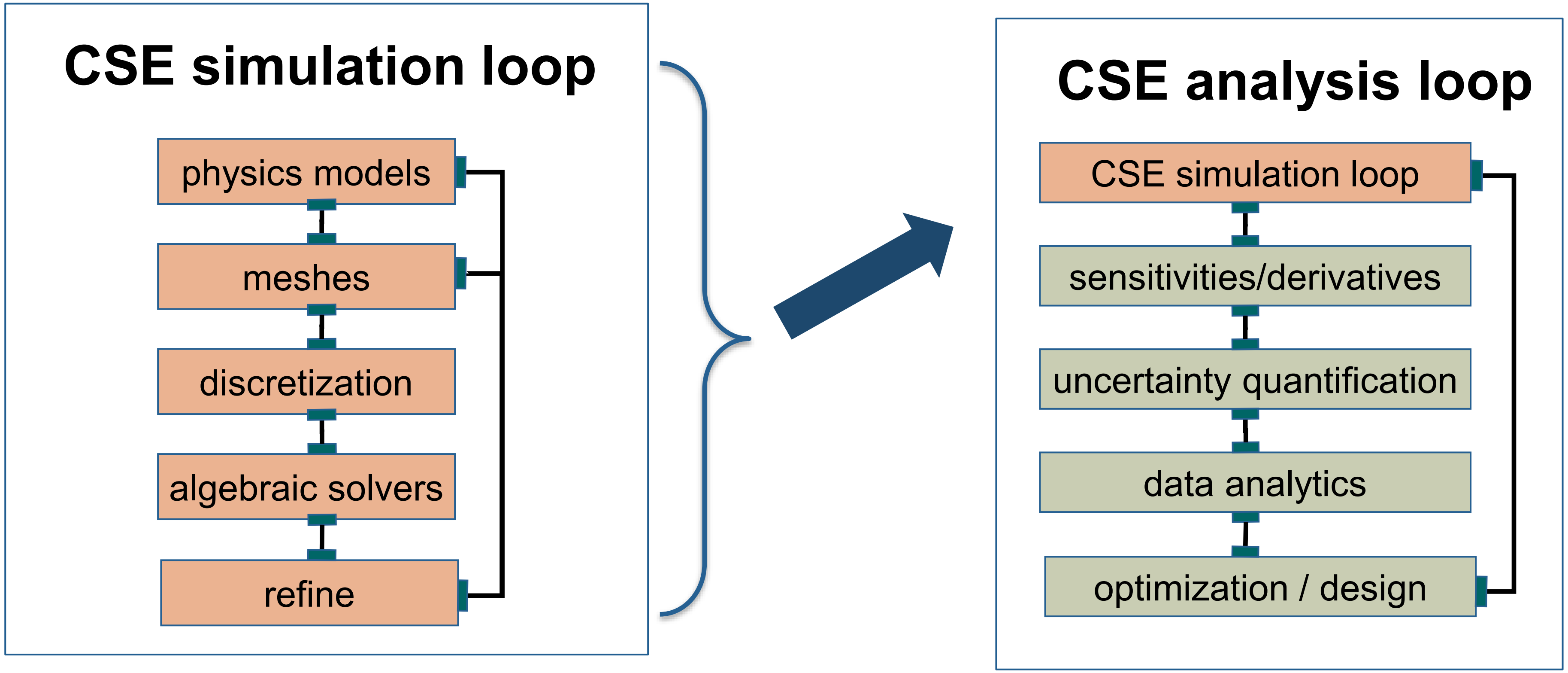
- Develop a mathematical model of the phenomenon of interest
- Approximate the model using a discrete representation
- Solve the discrete representation
- Adapt and refine the mesh or model
- Incorporate different physics, scales





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

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



Beyond interpretive simulations …

working toward predictive science

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

First consider a very simple example

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

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

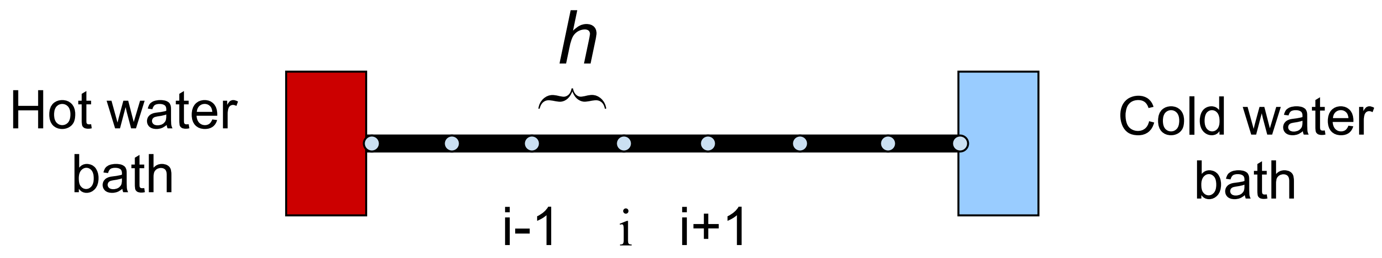
$$T(0) = 180^o \quad T(1) = 0^o$$

The first step is to discretize the equations

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

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

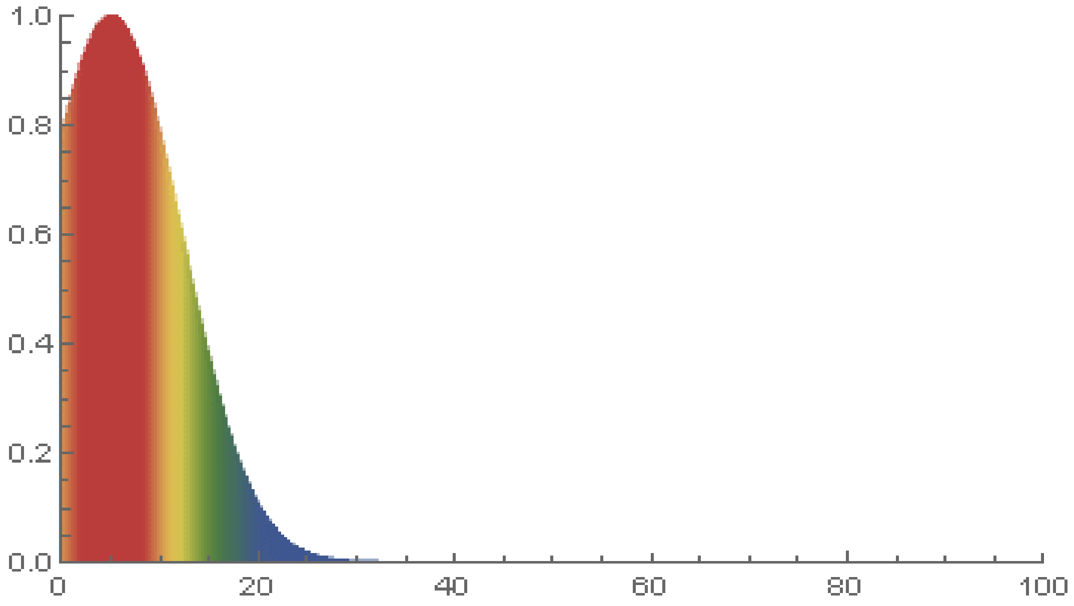
$$T_0 = 180^o \qquad T_n = 0^o$$



Then you can solve for the unknowns T_i

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

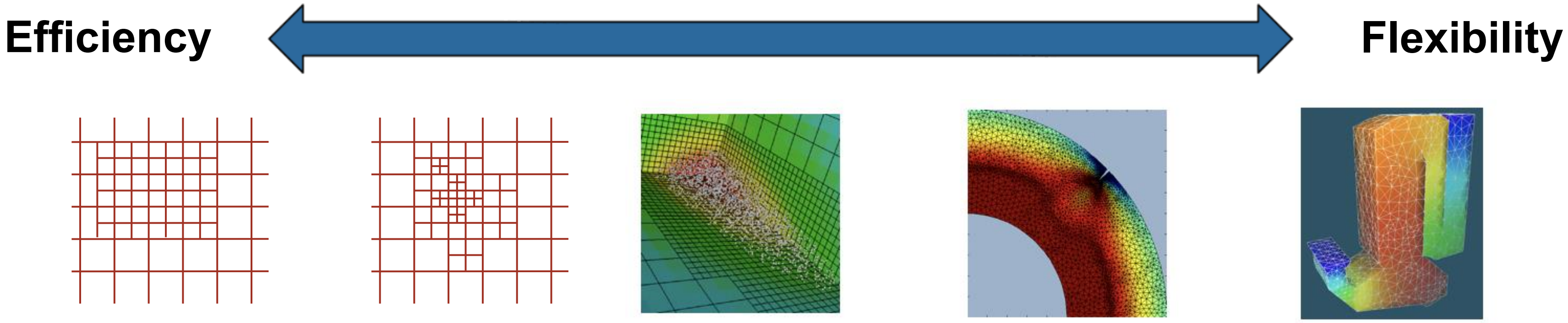
$$\begin{pmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 \ldots & 0 \\ 0 & -1 & 2 & -1 \;\; 0 \ldots & 0 \\ & & \ldots & & \\ 0 & \ldots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ . \\ T_{n-1} \end{pmatrix} = \begin{pmatrix} 180\, h^2 \\ 0 \\ 0 \\ . \\ 0 \end{pmatrix}$$



- Visualize and analyze the results

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

- Different discretization strategies exist for differing needs



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

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

https://scidac5-fastmath.lbl.gov

EXASCALE COMPUTING PROJECT

https://exascaleproject.org

- FASTMath SciDAC Institute
- Exascale Computing Project (ECP)
- 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 8:
Software Sustainability (Aug 8)

Community efforts: Join us!

https://xsdk.info

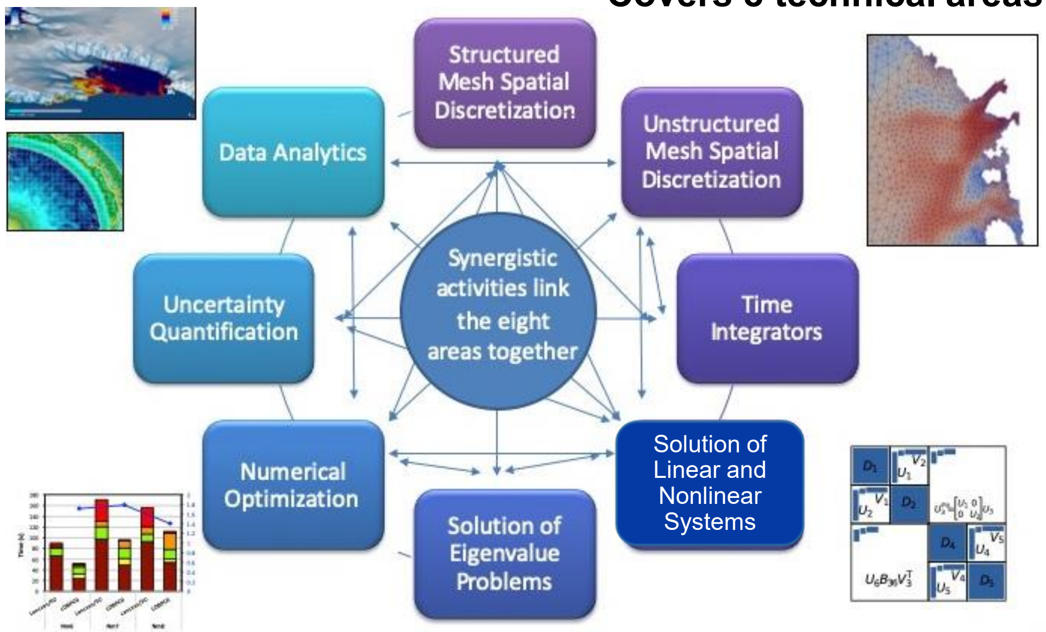
https://bssw.io

https://e4s.io

FASTMath: Frameworks, Algorithms & Scalable Technologies for Mathematics

https://scidac5-fastmath.lbl.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

100's of person years of experience building math software

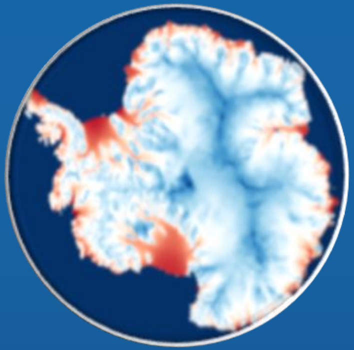
SuperLU, AMReX, sundials, StrumPACK, DAKOTA, mfem, PUMi, hypre, libEnsemble, Albany, TRILINOS, PETSc, TAO

50+ researchers from 6 DOE labs and 6 universities

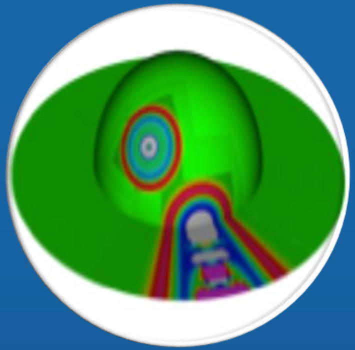
Berkeley Lab, Argonne National Laboratory, Oak Ridge National Laboratory, Sandia National Laboratories, NREL, USC, MIT, Rensselaer, SMU

Carol Woodward, FASTMath Institute Director (cswoodward@llnl.gov)

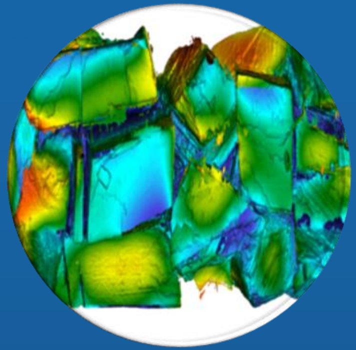
Structured grid efforts focus on high-order, mapped grids, embedded boundaries, AMR, and particles



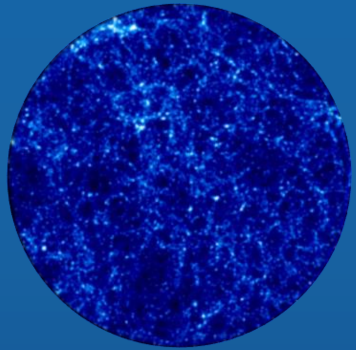
Structured AMR



Mapped-multiblock grids



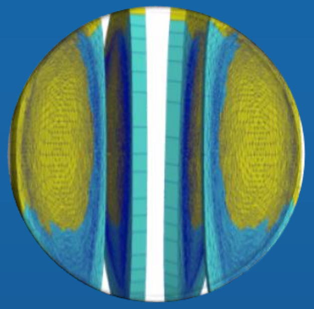
Embedded boundary methods



Particle-based methods

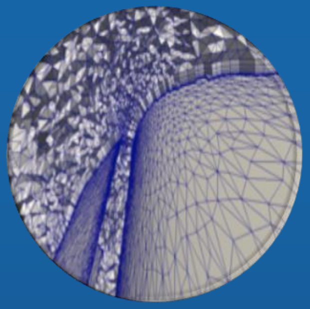
Application to cosmology, astrophysics, accelerator modeling, fusion, climate, subsurface reacting flows, low mach number combustion, etc.

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

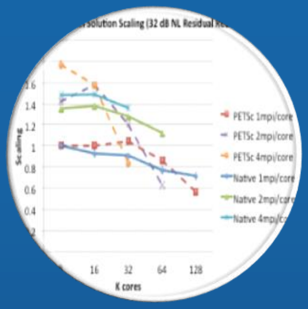


Parallel mesh infrastructures

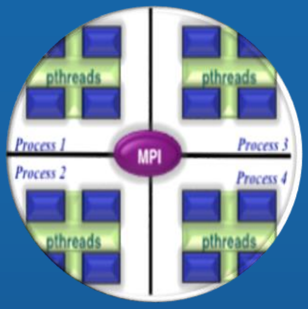
Dynamic load balancing



Mesh adaptation and quality control



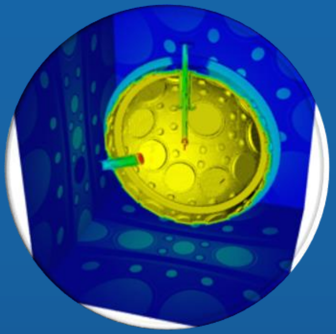
Parallel performance on unstructured meshes



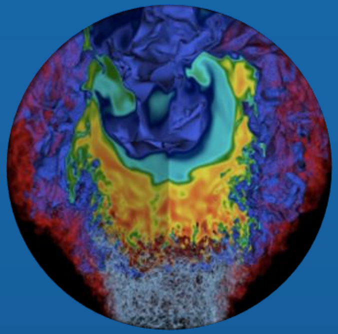
Architecture aware implementations

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

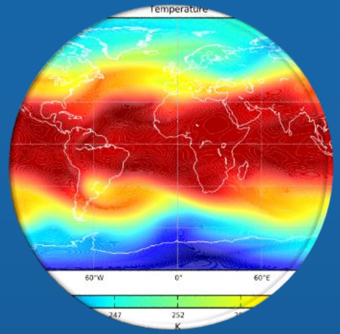


IMEX, multi-rate methods



Spectral deferred correction techniques

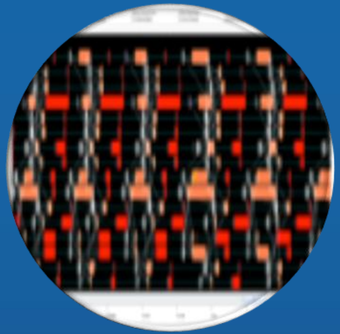
Parallel in time



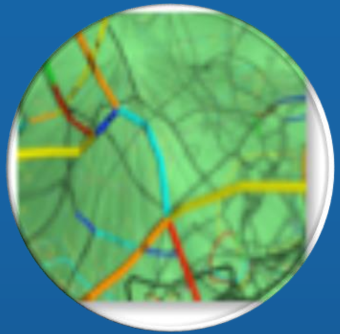
Constrained problems

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

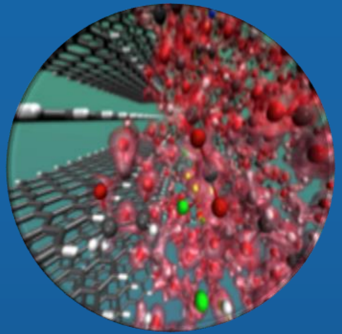
Research on algebraic systems provides key solution technologies to applications



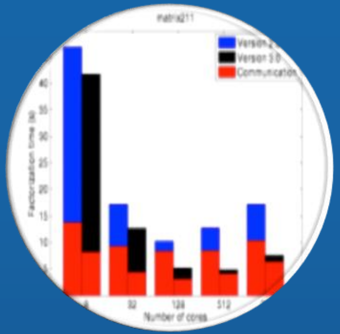
Linear system solution using direct and iterative solvers



Nonlinear system solution using acceleration techniques and globalized Newton methods



Eigensolvers using iterative techniques and optimization



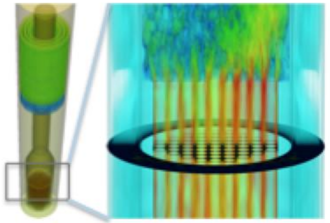
Architecture aware implementations

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

Multiphysics: A primary motivator for exascale

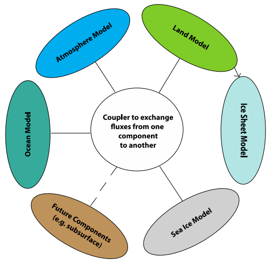
Multiphysics: greater than 1 component governed by its own principle(s) for evolution or equilibrium

– Also: broad class of coarsely partitioned problems possess similarities

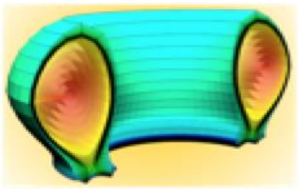


nuclear reactors
A. Siegel, ANL

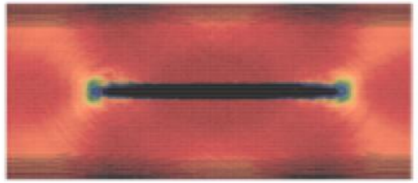
particle accelerators
K. Lee, SLAC



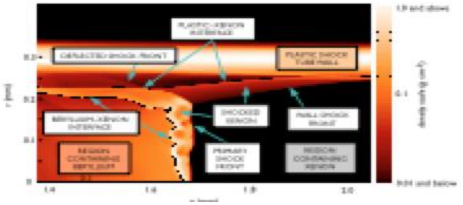
climate
K. Evans, ORNL



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

The Exascale Computing Project was designed to help launch the exascale era

ECP funded teams spanned national labs, universities, and industry

$1.8B over 7 years in a formal Department of Energy 413.3B project

Funded by DOE's Office of Science and NNSA programs

81 **Total research, development, and deployment projects**

- 6 Co-Design Centers
- 25 Application Development
- 35 Software Technology
- 17 Hardware and Integration

15	**57**	**36**
US DOE Labs, 6 of which are Core Partners	University Partners	Industry Organizations

- A **unique collaboration** brought together some of the brightest application, software, and computational experts from coast to coast
- **Best practices and lessons learned** for how to program GPUs – moving the nation forward
- **1000+ researchers** trained and ready for accelerator-based computing
- **1000+ students** introduced to HPC and Exascale computing through ECP's outreach, training, and workforce development initiatives

Technical work was largely complete as of Dec 31, 2023

Project leadership team is now working to close out the formal DOE 413.3B project

Next 18 slides courtesy of Lori Diachin

The Exascale Computing Project was started in 2016 and tasked to meet four DOE mission needs in high performance computing

Deliver a long-term, **sustainable software ecosystem** that can be used and maintained for years to come

Promote the **health of the US HPC industry**

Ensure that exascale systems can be used to deliver **mission-critical applications**

Maintain **international leadership in HPC**

To meet mission needs, the ECP was organized into three technical focus areas partnered with formal project management experts

Performant mission and science applications at scale

Aggressive RD&D project	Mission apps; integrated S/W stack	Deployment to DOE HPC Facilities	Hardware technology advances

Application Development (AD)

Develop and enhance the predictive capability of applications critical to DOE

24 applications
National security, energy, Earth systems, economic security, materials, data

6 co-design centers
ML, graph analytics, mesh refinement, PDE discretization, particles, online data analytics


Andrew Siegel, AD Director
Erik Draeger, AD Deputy Director

Software Technology (ST)

Deliver expanded and vertically integrated software stack to achieve full potential of exascale computing

70 unique software products spanning programming models and runtimes, math libraries, data and visualization, development tools


Mike Heroux, ST Director
Lois Curfman McInnes, ST Deputy Director

Hardware and Integration (HI)

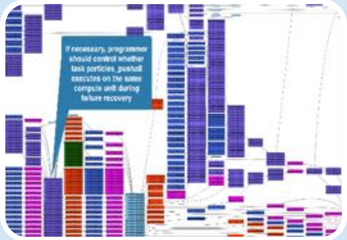
Integrated delivery of ECP products on targeted systems at leading DOE HPC facilities

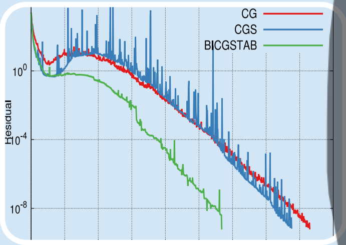
6 US HPC vendors
focused on exascale node and system design; application integration and software deployment to Facilities

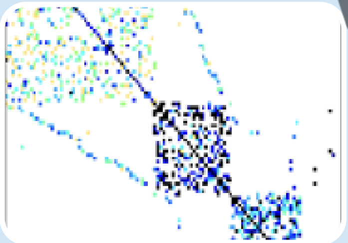

Richard Gerber, HI Director
Susan Coghlan, HI Deputy Director

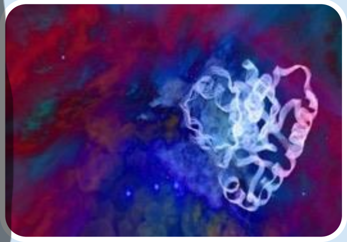
ECP invested in six software technology areas

Emphasis for this presentation









E4S

NNSA
National Nuclear Security Administration

Programming Models & Runtimes

- Enhance and get ready for exascale the MPI and OpenMP programming models (hybrid programming models, deep memory copies)
- Develop performance portability tools (e.g., Kokkos and Raja)
- Support alternate models for potential benefits and risk mitigation: PGAS (UPC++/GASNet), task-based models (Legion, PaRSEC)
- Libraries for deep memory hierarchy and power management

Development Tools

- Continued, multifaceted capabilities in portable, open-source LLVM compiler ecosystem to support expected ECP architectures, including support for F18
- Performance analysis tools that accommodate new architectures, programming models, e.g., PAPI, Tau

Math Libraries

- Linear algebra, iterative linear solvers, direct linear solvers, integrators and nonlinear solvers, optimization, FFTs, etc
- Performance on new node architectures; extreme strong scalability
- Advanced algorithms for multi-physics, multiscale simulation and outer-loop analysis
- Increasing quality, interoperability, complementarity of math libraries

Data and Visualization

- I/O via the HDF5 API
- Insightful, memory-efficient in-situ visualization and analysis
- Data reduction via scientific data compression
- Checkpoint restart

Software Ecosystem

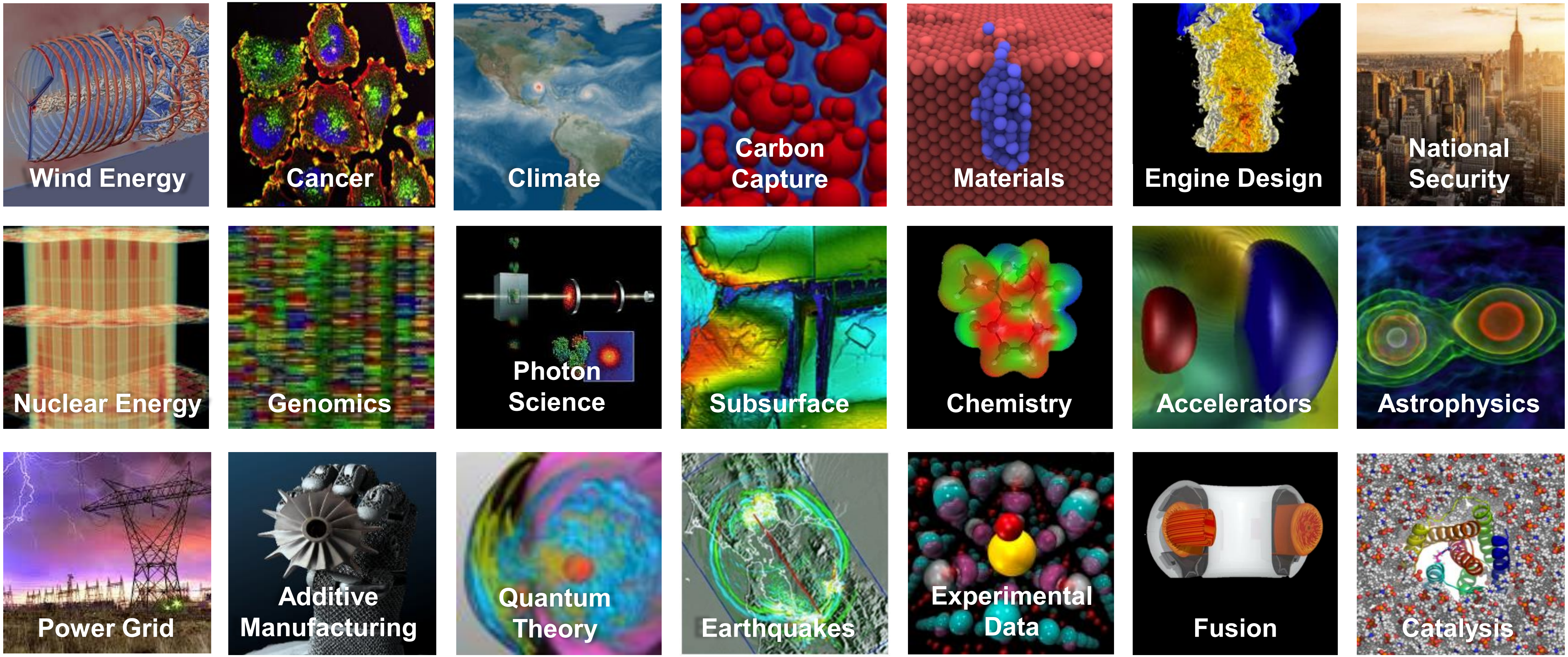
- Develop features in Spack necessary to support ST products in E4S, and the AD projects that adopt it
- Develop Spack stacks for reproducible turnkey software deployment
- Optimization and interoperability of containers for HPC
- Regular E4S releases of the ST software stack and SDKs with regular integration of new ST products

NNSA ST

- Open source NNSA Software projects
- Projects that have both mission role and open science role
- Major technical areas: New programming abstractions, math libraries, data and viz libraries
- Cover most ST technology areas
- Subject to the same planning, reporting and review processes

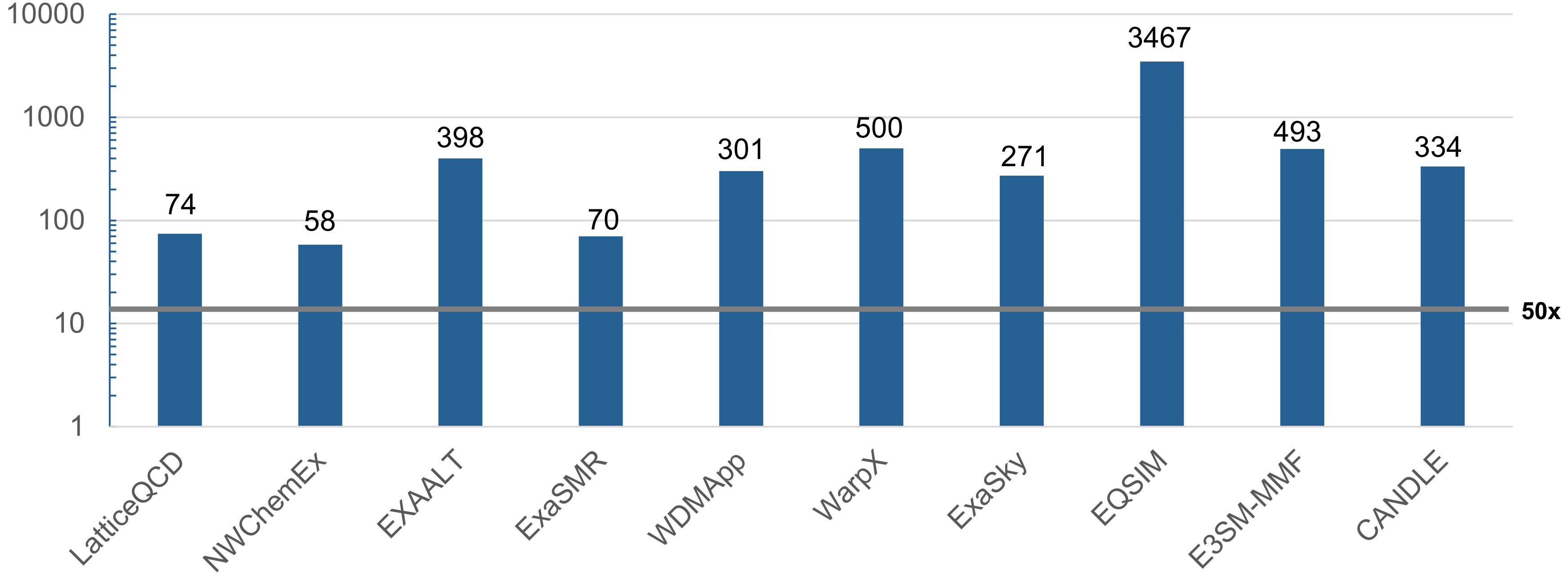
Area Leads: Rajeev Thakur | Jeff Vetter | Sherry Li | Jim Ahrens | Todd Munson | Kathryn Mohror

ECP invested in a broad range of critical application areas; some with limited HPC experience at project inception

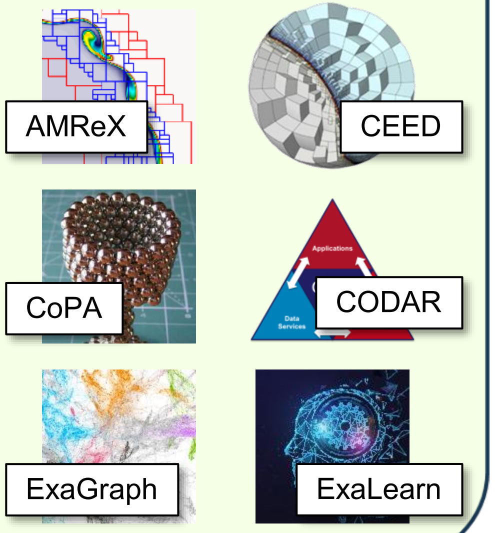


ECP application results exceeded expectations

10 out of 11 projects surpassed an ambitious 50x performance target

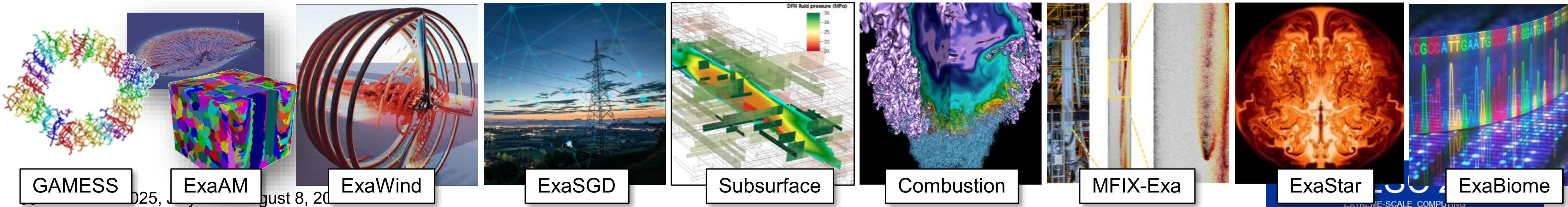


Co-design played a critical role



9 out of 10 new HPC science projects completed exascale capability demonstrations

3 out of 4 NNSA applications demonstrated exascale readiness

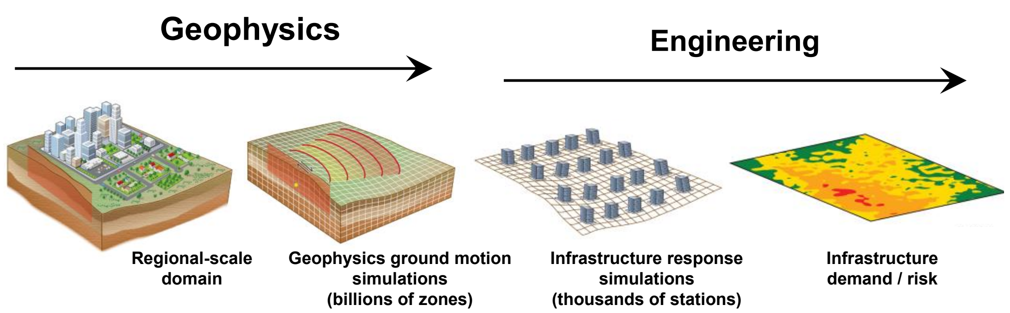


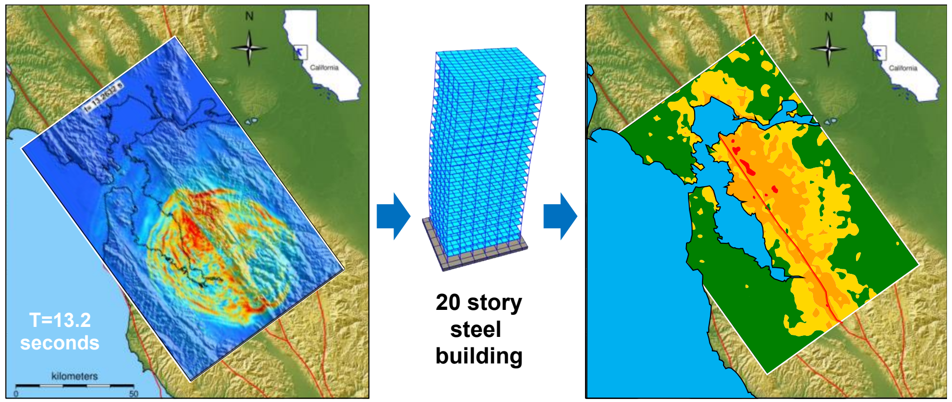
slide credit: A. Siegel and E. Draeger

One example EQSIM: exascale-capable code redesign had massive impact

EQSIM PI: Dave McCallen, LBNL

- **Starting point**: CPU-only code written in C by physicists. Code had complex nested loops (fourth-order finite difference stencil) that the compiler struggled to optimize, find SIMD.
- **ECP accomplishments**
  - Algorithmic improvements using curvilinear mesh refinement improved scientific work-rate by a factor of 2.85
  - Rewrote code in C++ with RAJA, with ZFP data compression to save sufficient data to maintain adequate precision in stored data
  - Infrastructure simulations now include strong coupling with OpenSees soil/building modeling and using in soil-structure interaction models; help gain insight into areas of maximum risk.



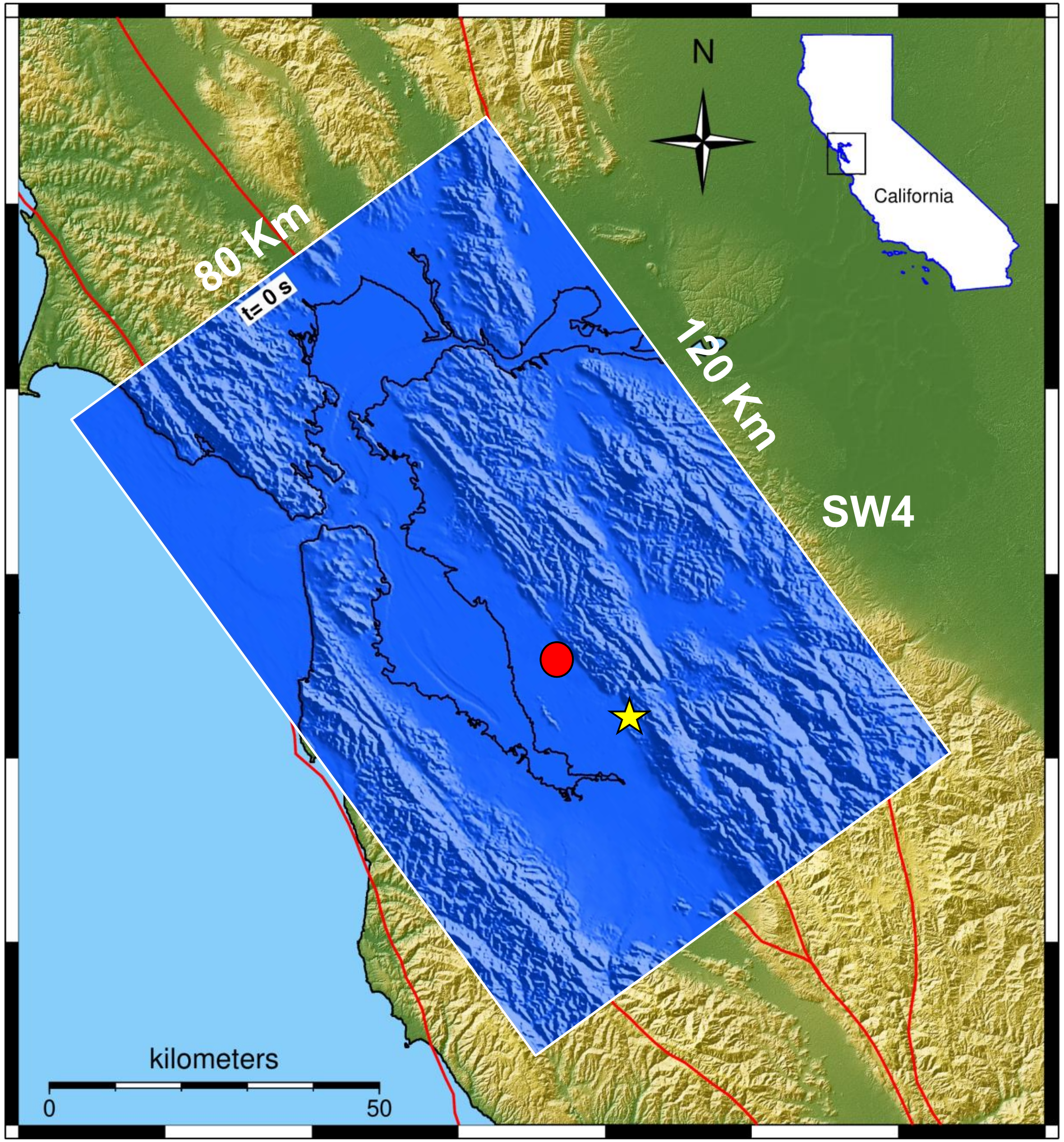


Achieved a **3500X improvement in computational performance** compared to initial baseline on Cori (~30 PF KNL system); Simulation of regional-scale ground motions at frequencies of engineering interest (5-10 Hz) now within reach.

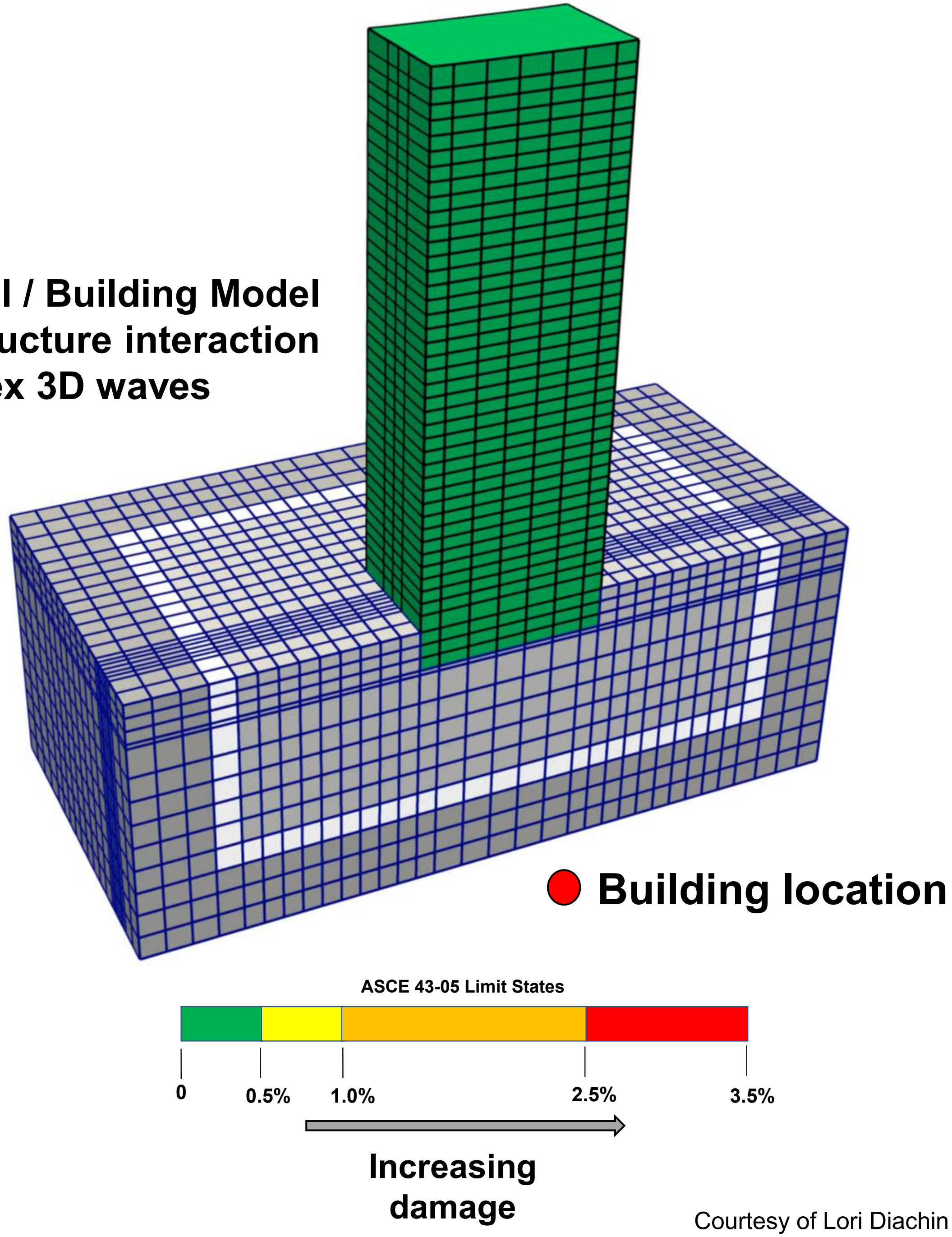
BERKELEY LAB

Regional Geophysics Model (M7 Hayward fault earthquake)

Fmax 10 Hz Vsmin 140 m/s
391 Billion grid points



Local Soil / Building Model
- Soil-structure interaction
- Complex 3D waves

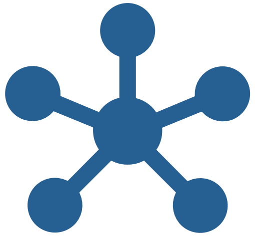


Courtesy of Lori Diachin

There where several key ingredients required for a successful ECP Application Development Project

Algorithmic innovation

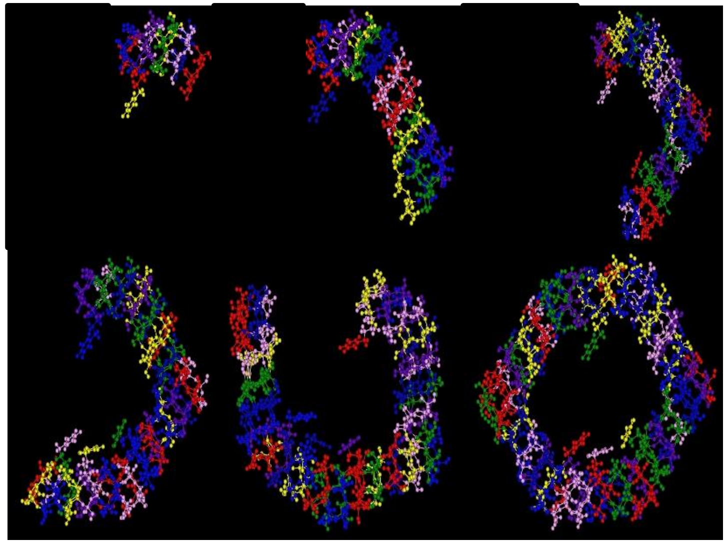
Porting

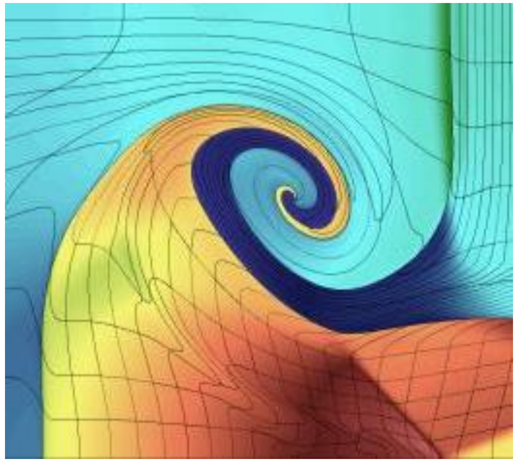


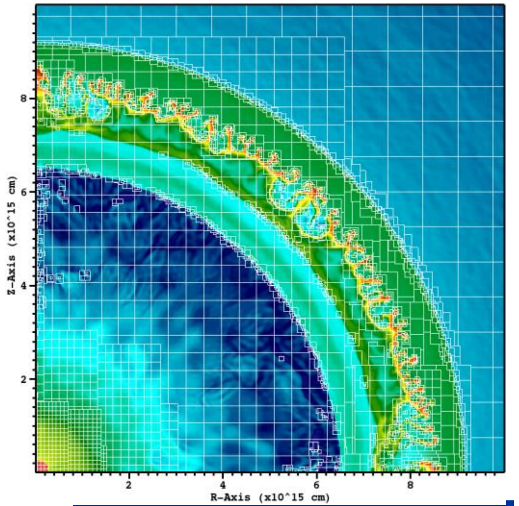
Integration

Algorithmic innovation: domain-driven adaptations critical for making efficient use of exascale systems

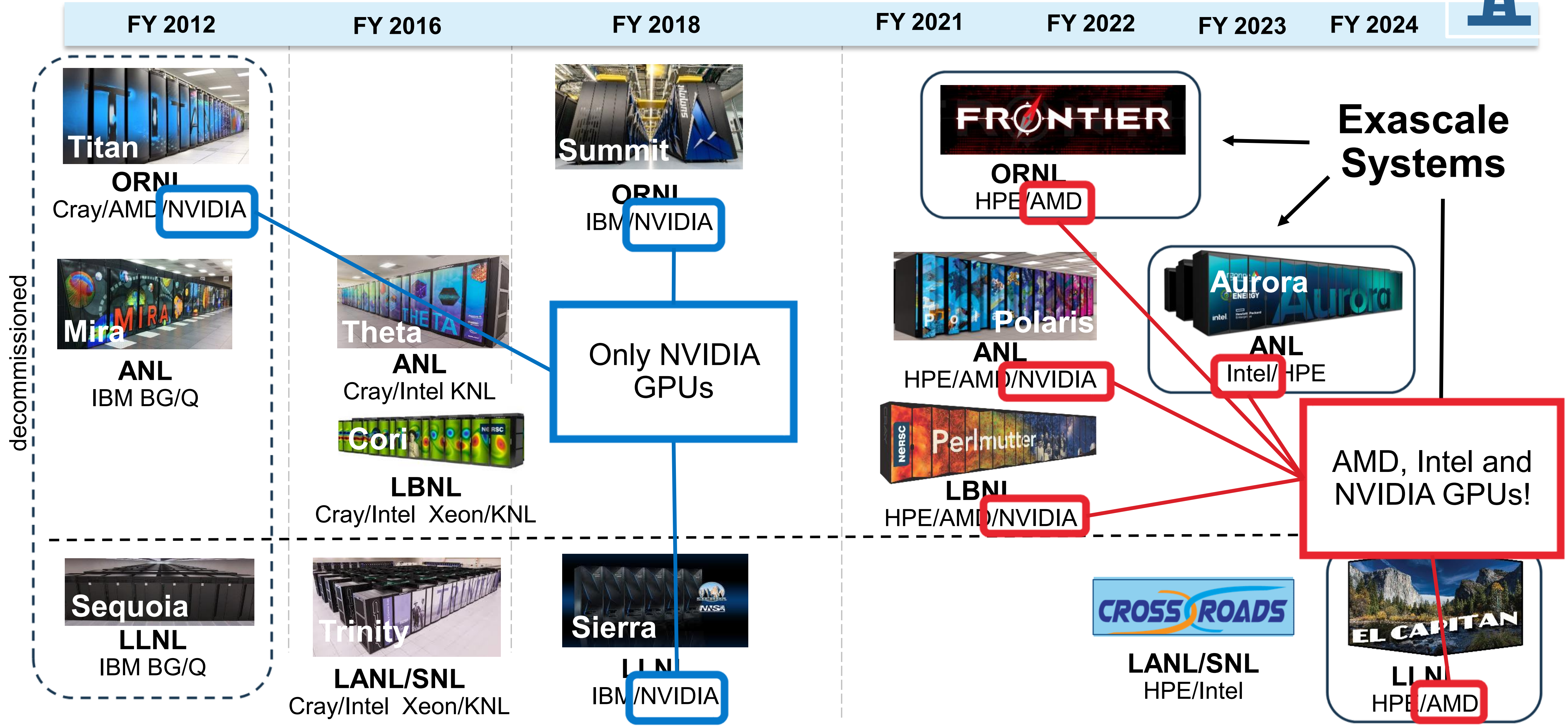
- Inherent strong scaling challenges on GPU-based systems
  - Ensembles vs. time averaging
  - Fluid dynamics, seismology, molecular dynamics, time-stepping
- Increase dimensions of (fine-grained) parallelism to feed GPUs
  - Ray tracing, Markov Chain Monte Carlo, fragmentation methods
- Localized physics models to maximize "free flops"
  - MMF, electron subcycling, enhanced subgrid models, high-order discretizations
- Alternatives to sparse linear systems
  - Higher order methods, Monte Carlo
- Reduced branching
  - Event-based models





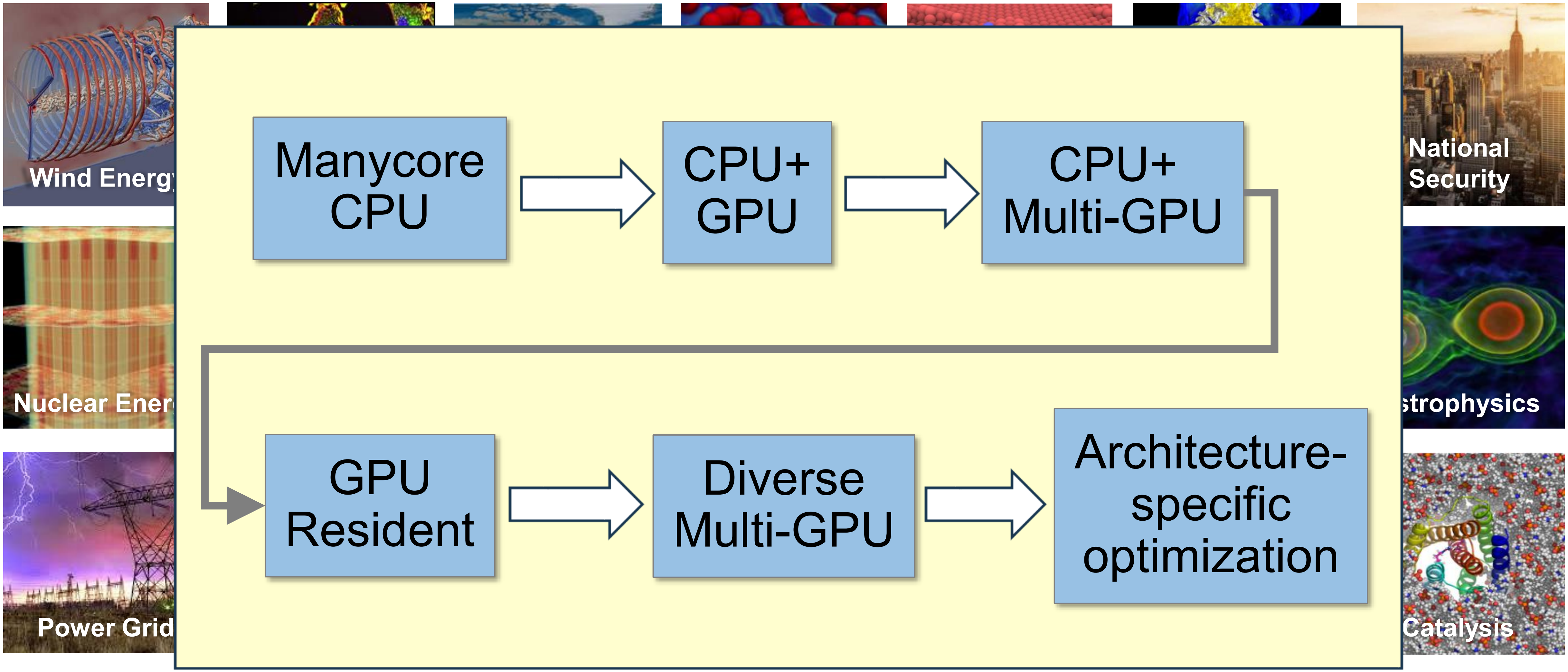


HPC systems have come a long way since ECP's inception



ATPESC 2025
EXTREME-SCALE COMPUTING

Exascale applications were designed to be flexible and adaptive



ECP applications teams used several different programming models to achieve performance portability

GPU-specific kernels

- Isolate the computationally-intensive parts of the code into CUDA/HIP/SYCL kernels.
- Refactoring the code to work well with the GPU is the majority of effort.

Loop pragma models

- Offload loops to GPU with OpenMP or OpenACC.
- Most common portability strategy for Fortran codes.

C++ abstractions

- Fully abstract loop execution and data management using advanced C++ features.
- Kokkos and RAJA developed by NNSA in response to increasing hardware diversity.

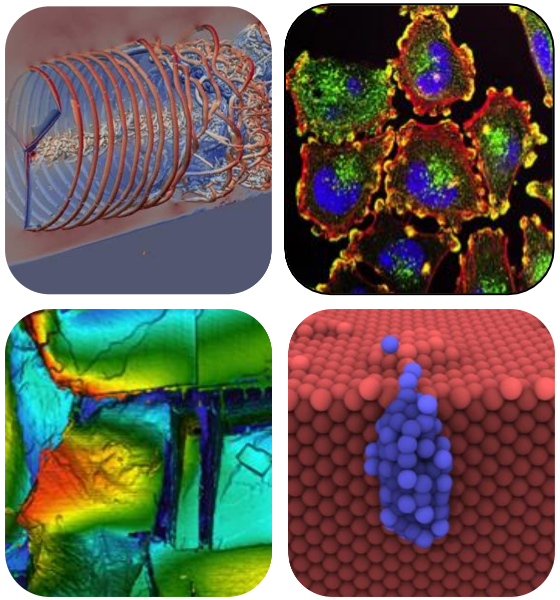
Co-design frameworks

- Design application with a specific motif to use common software components
- Depend on co-design code (e.g. CEED, AMReX) to implement key functions on GPU.

The success of exascale applications relied on significant software infrastructure

Programming abstractions

Memory / heterogeneous data management



In situ visualization and analysis

Solvers / preconditioners

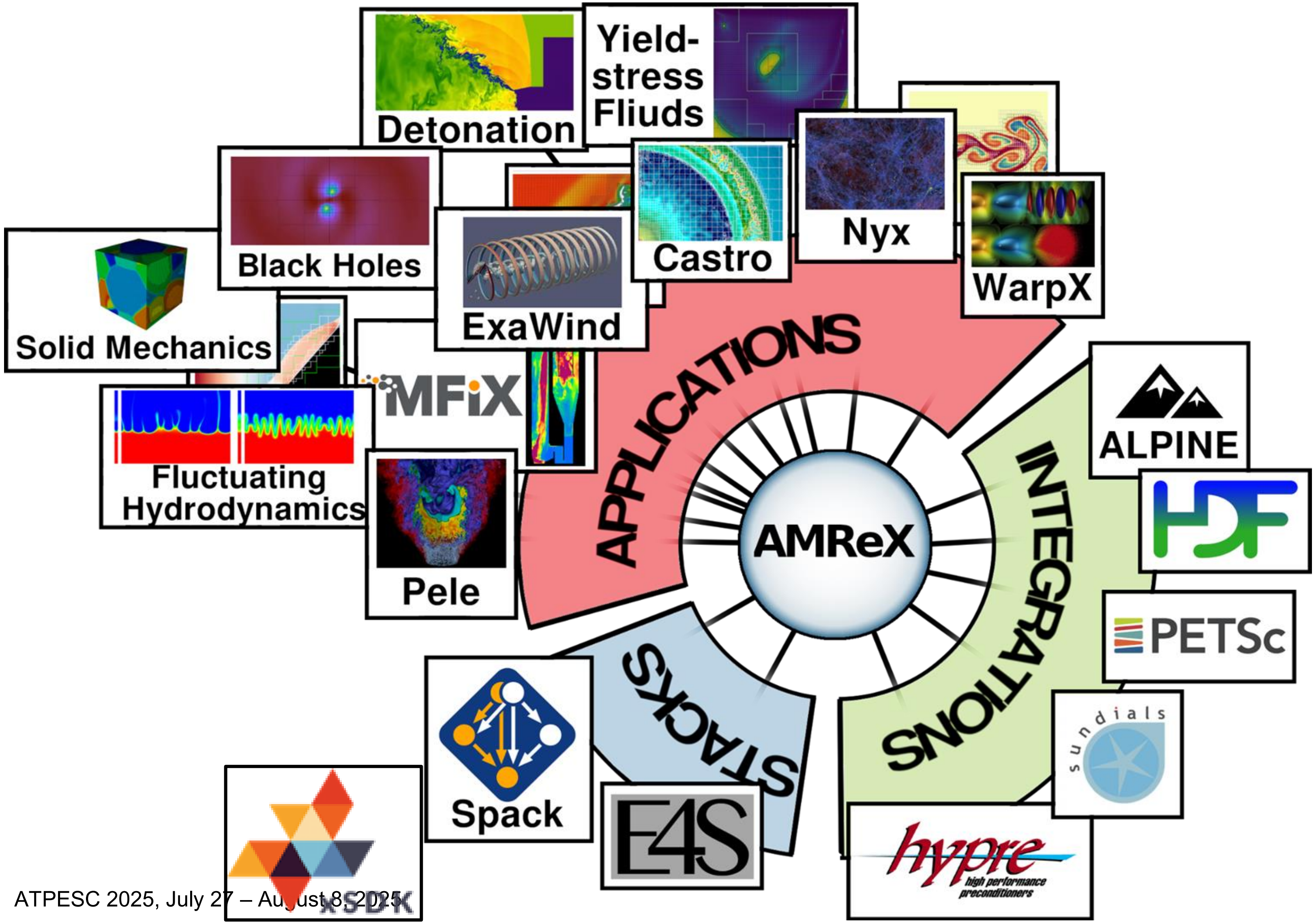
Numerical infrastructure / discretizations

Build and deployment

Performance analysis and monitoring

slide credit: A. Siegel and E. Draeger

Integration of multiple technologies allow users of higher-level frameworks, e.g., AMReX, access many different technologies



Software Integrations:

- SUNDIALS – Chemical reactions, time integrators
- *hypre*, PETSc – Linear solvers on mesh data
- In-situ: Ascent, Sensei
- Offline visualization: VisIt, Paraview, yt
- IO: HDF5, ADIOS

Software Stacks:

- Spack
  - Smoke test – CUDA, AMD HIP
- xSDK
- E4S

ECP software technologies were broadly deployed through the Extreme Scale Scientific Software Stack (E4S)

- ECP Software Technology lead: Mike Heroux (SNL)
- E4S: HPC software ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source**, **containers**, **cloud**, **binary caches**
- Not a commercial product – an open resource for all
- Supported by DOE and commercial entities (Paratools)
- Growing functionality: November 2023: E4S 23.11 – 100+ full release products

Levels of Integration | Product | Source and Delivery

ECP ST Open Product Integration Architecture

- Build all SDKs
- Build complete stack
- Containerize binaries

E4S

Source: ECP E4S teams; Non-ECP products (all dependencies)
Delivery: spack install e4s; containers; CI testing

- Group similar products
- Make interoperable
- Assure policy compliant
- Include external products

SDKs

Source: SDK teams; Non-ECP teams (policy compliant, spackified)
Delivery: Apps directly; spack install sdk; future: vendors and facilities

- Standard workflow
- Existed before ECP

ST Products

Source: ECP teams; Non-ECP teams; Standards
Delivery: Apps directly; spack; vendor & facility stacks

ECP ST Individual Products

Community Policies Commitment to SW quality	DocPortal Single portal to all E4S product info
Portfolio testing Especially leadership platforms	Curated collection The end of dependency hell
Quarterly releases	Build caches 10X build time improvement
Turnkey stack A new user experience	https://e4s.io
Post-ECP Strategy LSSw, ASCR Task Force	

https://e4s.io

E4S lead: Sameer Shende (U Oregon)

Spack https://spack.io

Spack lead: Todd Gamblin (LLNL)

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

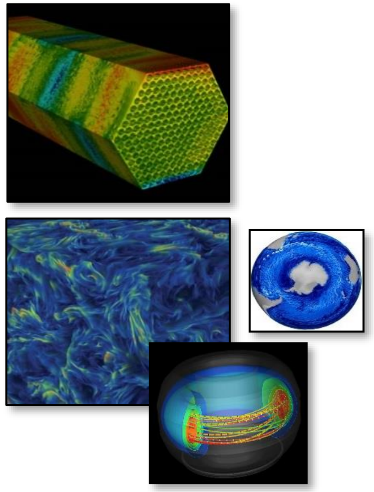
Next-generation algorithms

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

xSDK Version 1.1.0: November 2024

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.

Multiphysics Application C

Application A

Application B

xSDK functionality, Nov 2024

Tested on Linux work stations

SLEPc, hypre, AMReX, Omega_h, deal.II, heFFTe (Highly Efficient FFTs for Exascale), Trilinos, PETSc, TAO, DTK, PHIST, PUMi, sundials, MAGMA, mfem, ArborX, HiOp, xSDK, SuperLU, StrumPACK, preCICE, Ginkgo, libEnsemble, Spack, PLASMA, ButterflyPACK, ECP SLATE ICL, TASMANIAN

HDF5

BLAS

More external software

November 2024

- 25 math libraries
- 17 mandatory xSDK community policies
- Spack xSDK installer

Domain components
- Reacting flow, etc.
- Reusable.

Libraries
- Solvers, etc.
- Interoperable.

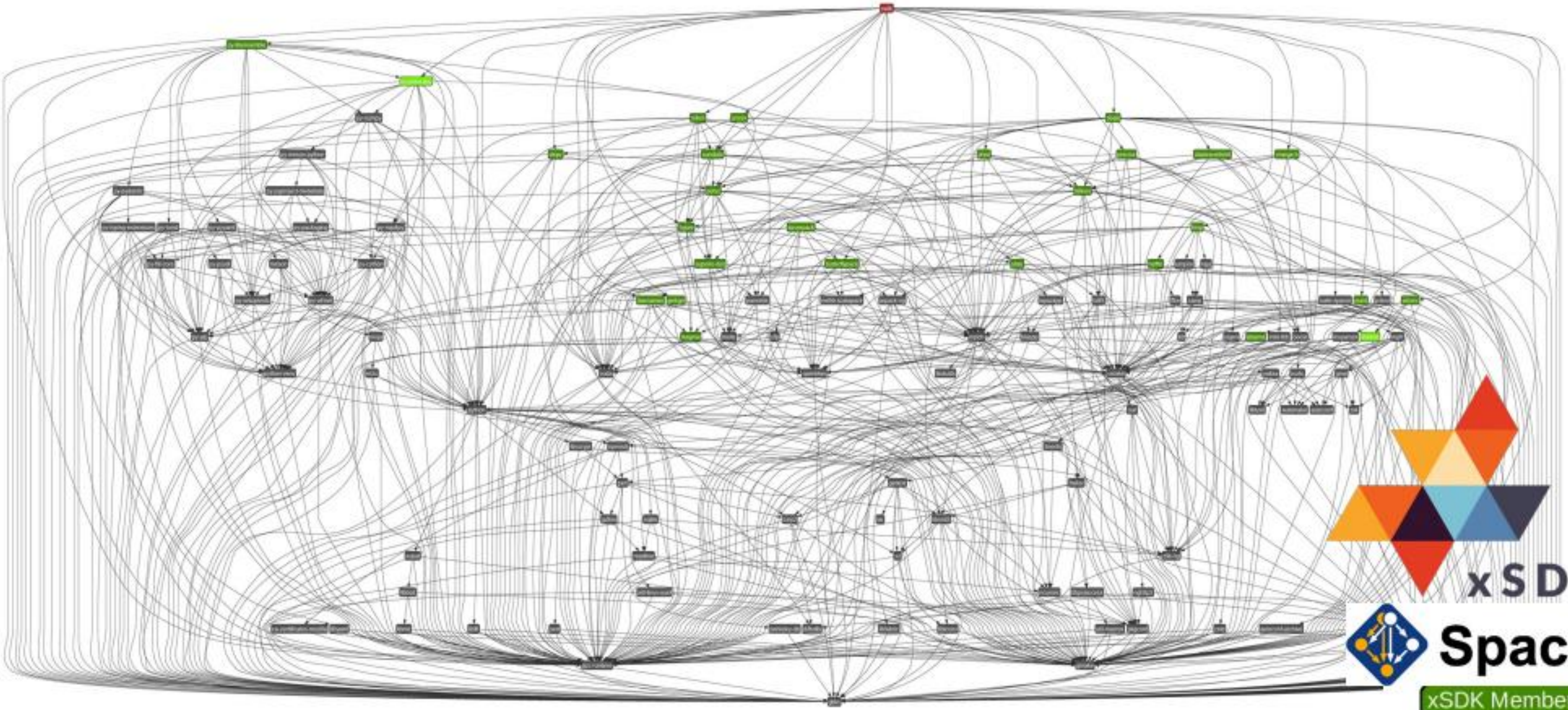
Frameworks & tools
- Doc generators.
- Test, build framework.

SW engineering
- Productivity tools.
- Models, processes.

Extreme-Scale Scientific Software Development Kit (xSDK)

Impact: Improved code quality, usability, access, sustainability

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



xSDK

Spack

xSDK Member

Dependency

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!

The Exascale Computing Project has delivered on its mission needs

Deliver a long-term, **sustainable software ecosystem** that can be used and maintained for years to come

- E4S deployed at HPC facilities around the US and the world
- 76 HPC products available for computing at all scales
- Performance portability tools developed and widely used

Promote the **health of the US HPC industry**

- Six vendors funded under PathForward; outcomes realized in exascale systems
- Accelerator-based computing lowers cost of energy across the board
- The ECP Industry and Agency Council stimulates consumption of HPC resources

Ensure that exascale systems can be used to deliver **mission-critical applications**

- ECP applications demonstrate outstanding performance and capabilities at exascale
- Previously unattainable results in real-world challenge problems
- ECP lessons learned pave the way for many additional applications to leverage accelerator-based computing

Maintain **international leadership in HPC**

- El Capitan is the world's first exascale machine, Frontier the world's second exascale system – in part due to ECP/ECI investments
- Aurora is the world's third exascale system
- 1000+ researchers trained in GPU computing

HandsOn Lessons

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

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

And more …

Github pages site:

https://xsdk-project.github.io/MathPackagesTraining2025/lessons/

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

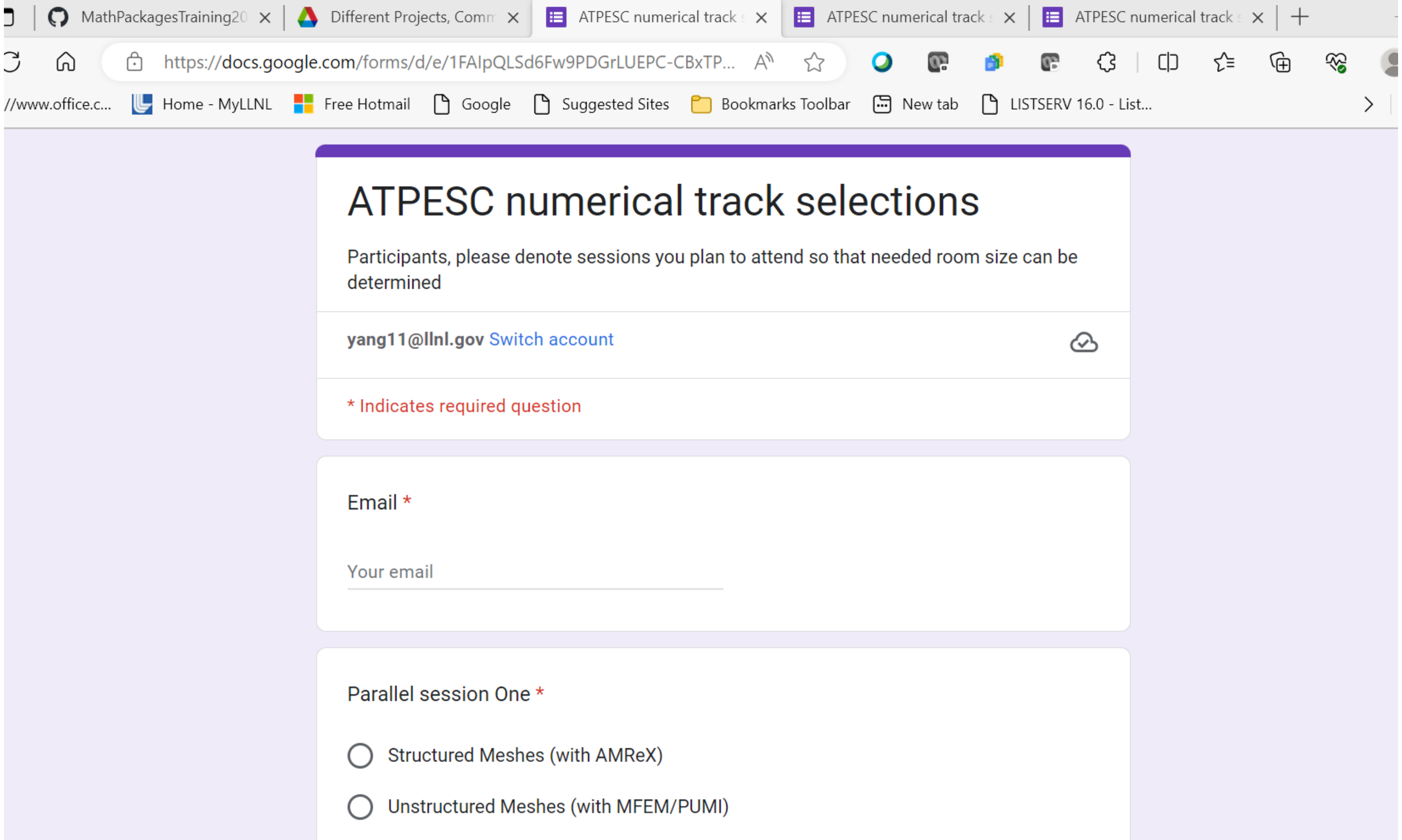
Time	Room?	Room?
8:30 – 9:30	Introduction to Numerical Software – Ulrike Yang	
9:30 – 10:45	Structured Discretization (AMReX) – Andrew Myers, Weiqun Zhang	Unstructured Discretization (MFEM/PUMI) – Mark Shephard, Cameron Smith, Mark Stowell
10:45 – 11:15	Break, Subject Matter Expert (SME) Selections, Panel Questions	
11:15 – 12:30	Iterative Solvers & Algebraic Multigrid (hypre) – Daniel Osei-Kuffuor, Ulrike Yang	Direct Solvers (SuperLU, STRUMPACK) – Sherry Li, Yang Liu
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**, **Panel Questions Due**	
3:15 – 4:30	Optimization (TAO) – Toby Isaac	Iterative Solvers & Algebraic Multigrid (Trilinos/ Belos/MueLU) – Christian Glusa, Graham Harper
4:30 – 5:15	Panel: Extreme-Scale Numerical Algorithms and Software	
5:15 – 6:30	Optional Activity: SME Speed-dating	
6:30 – 7:30	Dinner	
7:30 – 9:00	After-Dinner Talk: Jim Demmel	

Room Choice

- Please raise your hand if you want to attend
  - Structured meshing
  - Unstructured meshing

Choose which lecture you want to attend!

[ATPESC numerical track selections]

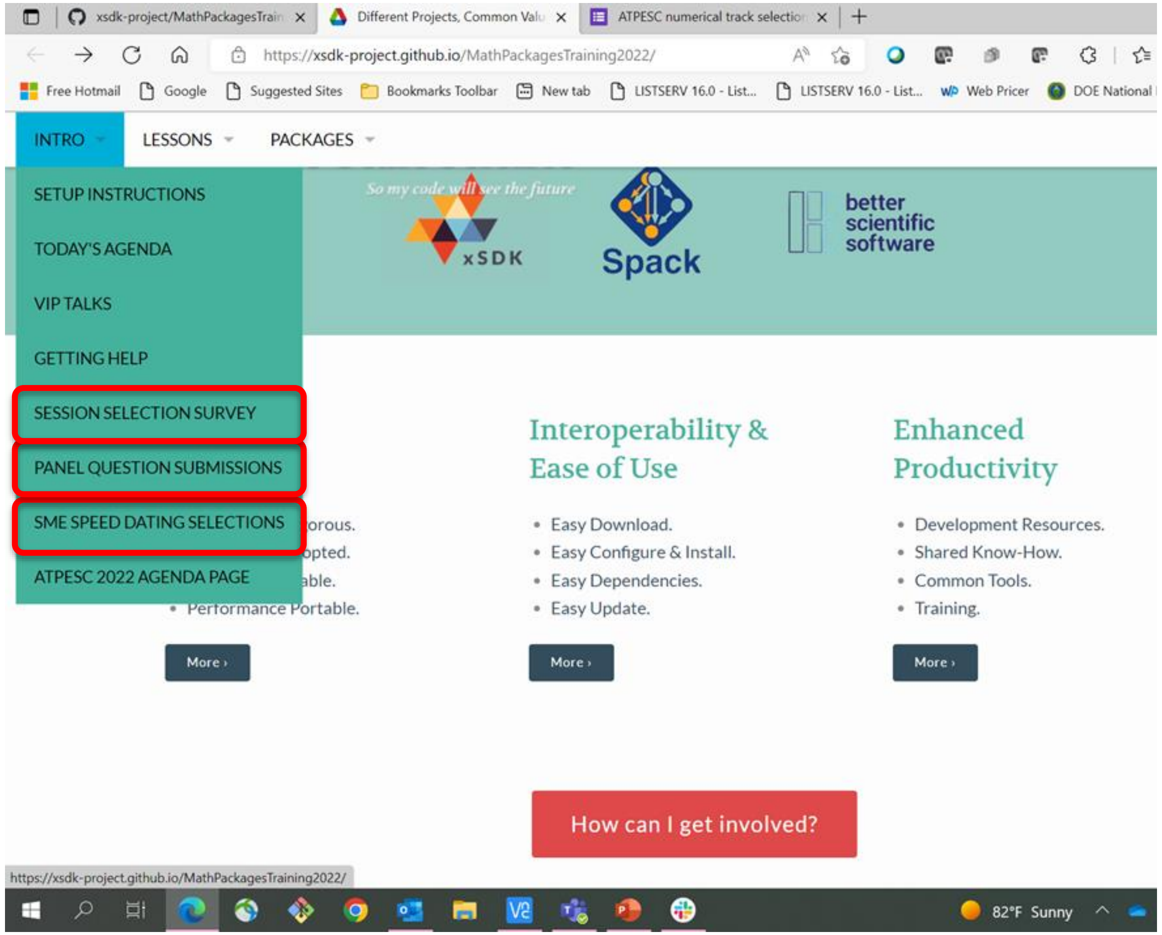


ATPESC 2025
EXTREME-SCALE COMPUTING

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 4:30 pm CDT



ATPESC 2025
EXTREME-SCALE COMPUTING

Thank you to all ATPESC staff

Special thanks to Ray Loy
and
Kathy Gorgan, Yasaman Ghadar,
Haritha Siddabathuni Som,
Johnny Marquez and …

For their outstanding work in running the 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-2009458
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, the FASTMath Institute, 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.