The State of MFEM

MFEM Community Workshop
October 25, 2022

Tzanio Kolev
LLNL

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

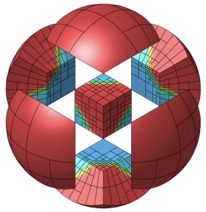
mfem

Lawrence Livermore National Laboratory

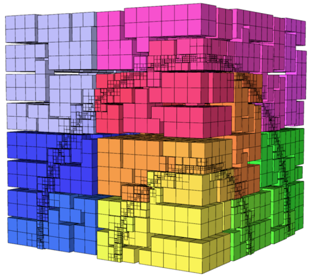
MFEM

Cutting-edge algorithms for powerful applications on HPC architectures

- **Flexible discretizations on unstructured grids**
  - Triangular, quadrilateral, tetrahedral and hexahedral meshes.
  - Local conforming and non-conforming AMR, mesh optimization.
  - 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 GPU-accelerated.
  - Enables application development from laptops to exascale machines.
- **Built-in solvers and visualization**
  - Integrated with: HYPRE, SUNDIALS, PETSc, SLEPc, SUPERLU, …
  - AMG preconditioners for full de Rham complex, geometric MG
  - Support for GPU solvers from: HYPRE, PETSc, AmgX
  - Accurate and flexible visualization with VisIt, ParaView and GLVis
- **Open source**
  - Available on GitHub under BSD license. 75+ example codes and miniapps.
  - Part of FASTMath, ECP/CEED, xSDK, OpenHPC, E4S, …

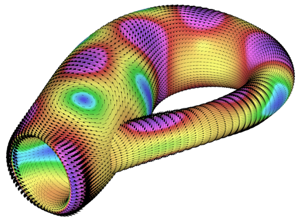


High-order curved elements



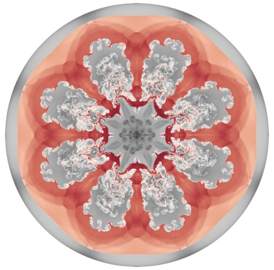
Parallel non-conforming AMR

ECP ASC CEED EXASCALE DISCRETIZATIONS FASTMATH



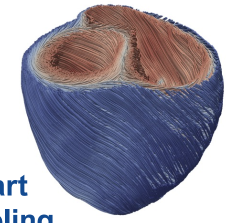
Surface meshes

xSDK



Compressible flow ALE simulations

2020 R&D 100 FINALIST

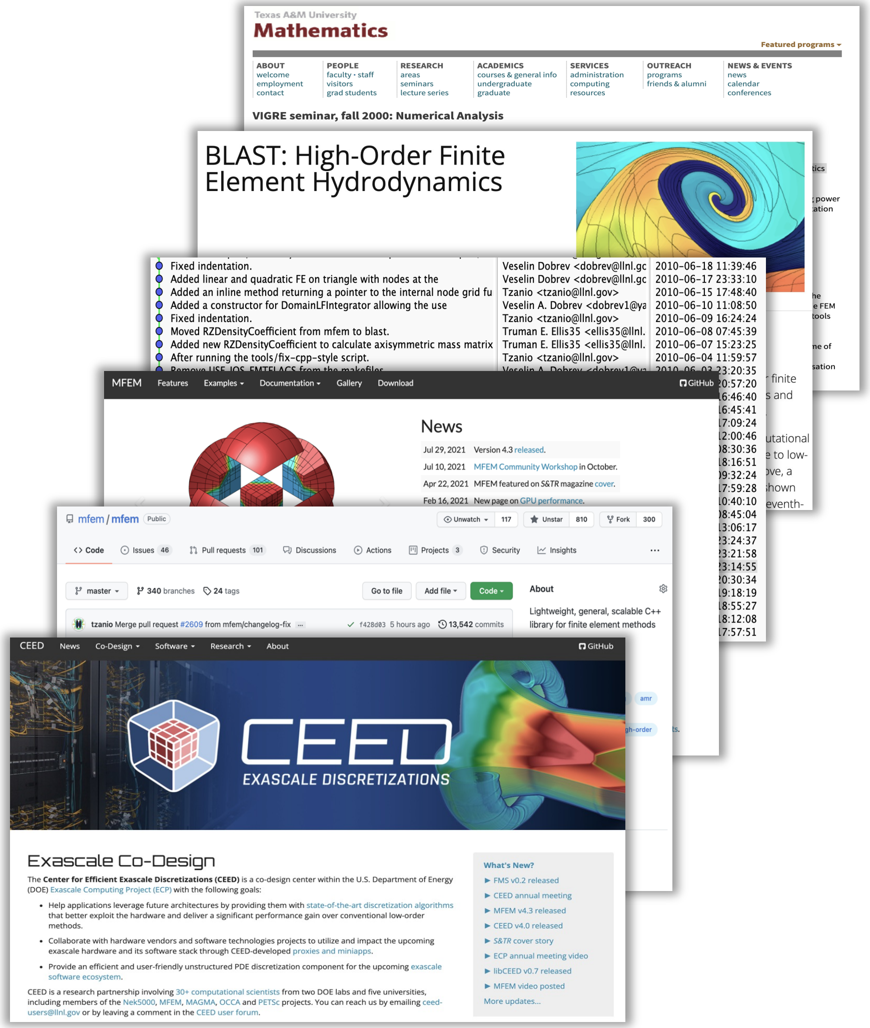


Heart modeling

A Brief History

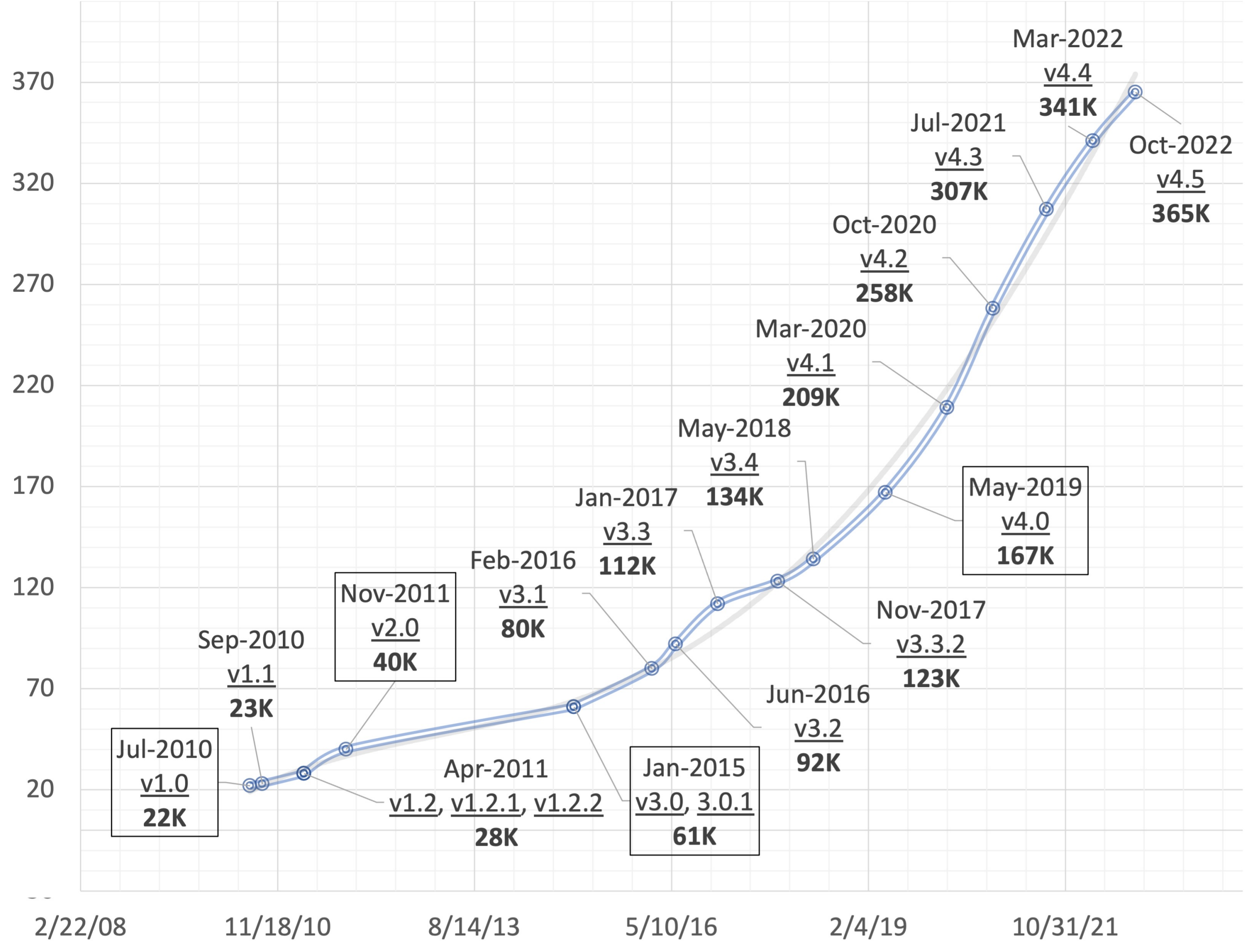
We've been doing this for a long time

- **2000 – "VIGRE seminar: Numerical Analysis," Texas A&M University**
  - Research code: AggieFEM/aFEM
  - Some of the original contributors: **@v-dobrev**, **@tzanio**, **@stomov**
  - Used in summer internships at LLNL
- **2010 – BLAST project at LLNL**
  - Motivated high-order, non-conforming AMR and parallel scalability developments
  - MFEM repository starts in May 2010
  - Some of the original contributors: **@v-dobrev**, **@tzanio**, **@rieben1**, **@trumanellis**
  - Project website **mfem.org** goes live in August 2015
- **2017 – Development moved to GitHub**
  - First GitHub commits in February 2017
  - Team expands to include many new developers at LLNL and externally
- **2017 – CEED project in the ECP**
  - Motivated partial assembly, GPU, and exascale computing developments



The Source Code Has Grown Significantly

SLOC in MFEM releases over the last 12 years



mfem-4.5.tgz	v4.5	Oct 2022	3.3M	365K	
mfem-4.4.tgz	v4.4	Mar 2022	3.0M	341K	
mfem-4.3.tgz	v4.3	Jul 2021	2.8M	307K	
mfem-4.2.tgz	v4.2	Oct 2020	2.4M	258K	
mfem-4.1.tgz	v4.1	Mar 2020	7.9M	209K	
mfem-4.0.tgz	v4.0	May 2019	5.2M	167K	*GPU support*
mfem-3.4.tgz	v3.4	May 2018	4.4M	134K	
mfem-3.3.2.tgz	v3.3.2	Nov 2017	4.2M	123K	*mesh optimization*
mfem-3.3.tgz	v3.3	Jan 2017	4.0M	112K	
mfem-3.2.tgz	v3.2	Jun 2016	3.3M	92K	*dynamic AMR, HPC miniapps*
mfem-3.1.tgz	v3.1	Feb 2016	2.9M	80K	*fem ↔ linear system interface*
mfem-3.0.1.tgz	v3.0.1	Jan 2015	1.1M	61K	
mfem-3.0.tgz	v3.0	Jan 2015	1.1M	61K	*non-conforming AMR*
mfem-2.0.tgz	v2.0	Nov 2011	308K	40K	*arbitrary order spaces, NURBS*
mfem-v1.2.2.tgz	v1.2.2	Apr 2011	240K	28K	
mfem-v1.2.1.tgz	v1.2.1	Apr 2011	240K	28K	
mfem-v1.2.tgz	v1.2	Apr 2011	240K	28K	*MPI parallelism based on hypre*
mfem-v1.1.tgz	v1.1	Sep 2010	166K	23K	
mfem-v1.0.tgz	v1.0	Jul 2010	160K	22K	*initial release*

The Community Has Grown Significantly

GitHub, downloads, and workshop stats

GitHub

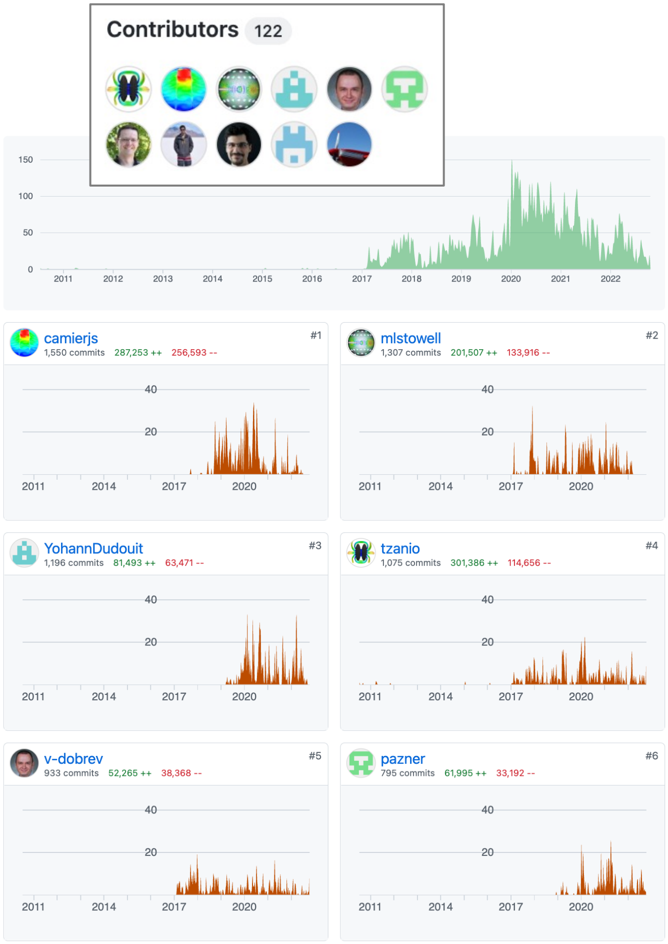
- **122** contributors
- **100** commits / week
- **541** people in the mfem organization – *join to contribute* + *receive announcements*
- **150** visitors / day
- **1040** stars – *thank you!*

Downloads

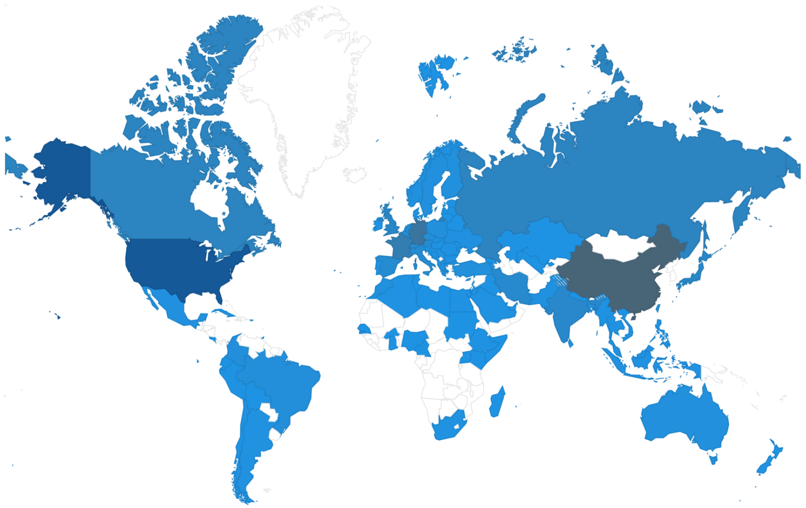
- **180** downloads + clones / day · **65K** / year
- **108** countries total

2022 Community Workshop

- **219** researchers
- **120** organizations
- **34** countries



Top contributors as of Oct 2022



MFEM has been downloaded from 108 countries

mfem.org **MFEM Community Workshop** *October 2022*

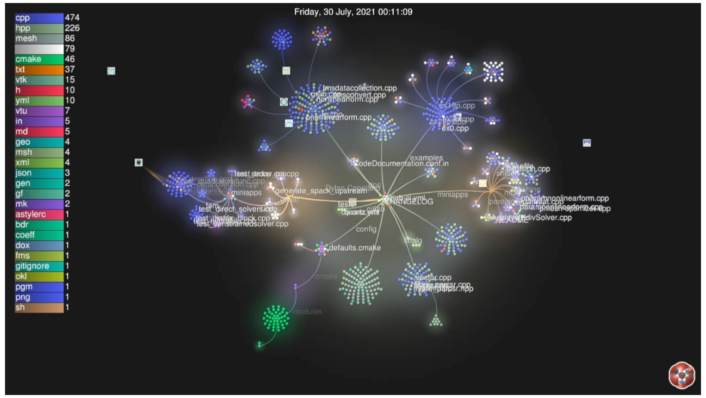
001	Aaron Fisher	*Lawrence Livermore National Laboratory (USA)*	fisher47@llnl.gov
002	Aayushman Raina	*IIT Guwahati (India)*	aayushman.raina@iitg.ac.in
003	Abhinav Gupta	*IIT Roorkee (India)*	iitrabhi@gmail.com
004	Abhishek Verma	*Applied Materials (USA)*	AbhishekKumar_Verma@amat.com
005	Adolfo Rodriguez	*OpenSim Technology (USA)*	adolfo@opensim.technology
006	Adriano Cortes	*Federal University of Rio de Janeiro (Brazil)*	adriano@nacad.ufrj.br
007	Alexander Blair	*UK Atomic Energy Authority (United Kingdom)*	alexander.blair@ukaea.uk
008	Alexander Grayver	*ETH Zurich (Switzerland)*	agrayver@ethz.ch
009	Alexander Ts.	*University of Illinois, Urbana Champaign (Greece)*	mseitw@gmail.com
010	Ali Alavi	*Fortress Technology Solutions (Canada)*	vecimaman@gmail.com
011	Alicia Elliott	*Google (USA)*	Aliciamelliott@gmail.com
012	Alvaro Sanchez Villar	*Princeton Plasma Physics Laboratory (USA)*	alvsanch@ing.uc3m.es
013	Andreas Meier	*Friedrich-Alexander-Universität Erlangen-Nürnberg (Germany)*	Andreas.Meier@fau.de
014	Andreas Schafelner	*Radon Institute (Austria)*	andreas.schafelner@ricam.oeaw.ac.at
015	Andres Martinez	*Universidad Nacional de Colombia (Colombia)*	aerubianoma@unal.edu.co
016	Andres Valdez	*Pennsylvania State University (USA)*	arvaldez@psu.edu
017	Andrew Gillette	*Lawrence Livermore National Laboratory (USA)*	gillette7@llnl.gov

2022 Community workshop had 219 registrations

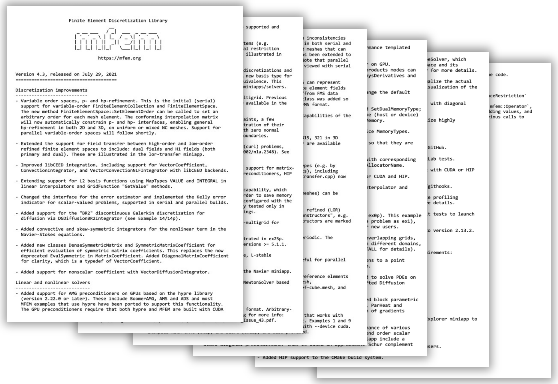
Latest Releases Were Team Efforts

Versions 4.4 + 4.5 stats

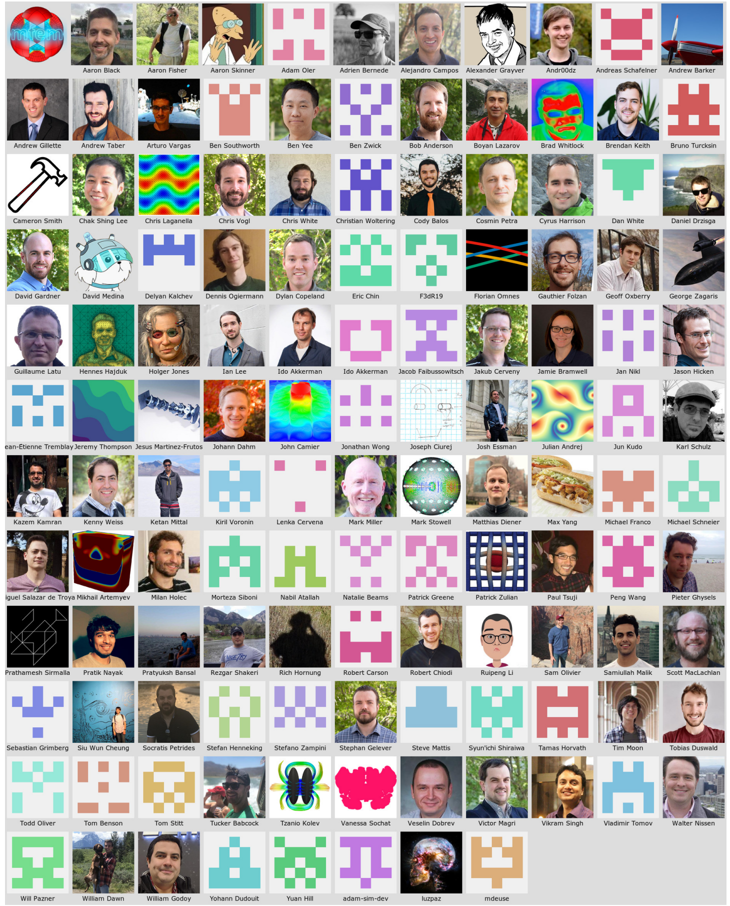
- Released **Mar 21** + **Oct 22, 2022**
- **15** months in development
- **73** contributors
- **579** PRs merged
- **366** issues closed
- **58K** new lines of code
- **3900** number of commits
- **Many new features:**
  - GPU kernels for DG, LOR, linear forms
  - AMG solvers on AMD GPUs
  - Submesh extraction, hr-adaptivity
  - AD for nonlinear elasticity (Hooke)
  - Enzyme, Algoim, ParMoonolith support



The making of MFEM versions 4.4 and 4.5
youtu.be/fHC019J1CWU


Finite Element Discretization Library
mfem
https://mfem.org
Version 4.3, released on July 29, 2021

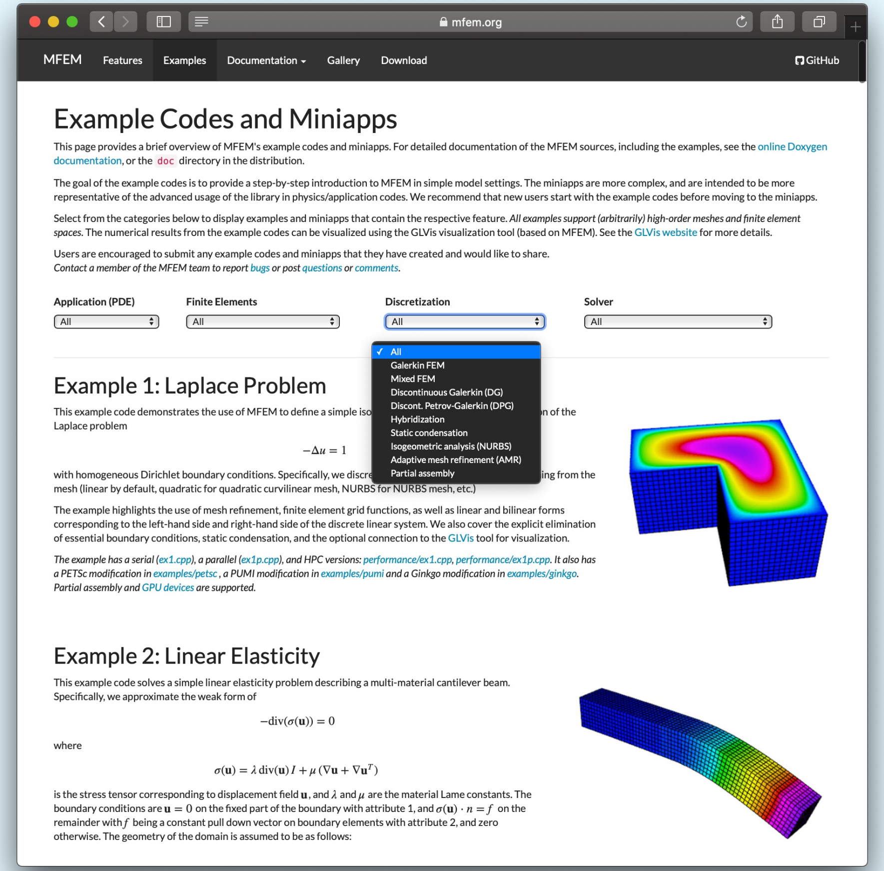
The mfem-4.4+4.5 CHANGELOG has 70 entries



MFEM contributors on GitHub

Examples

The first stop for new users



mfem.org

MFEM Features Examples Documentation Gallery Download GitHub

Example Codes and Miniapps

This page provides a brief overview of MFEM's example codes and miniapps. For detailed documentation of the MFEM sources, including the examples, see the online Doxygen documentation, or the doc directory in the distribution.

The goal of the example codes is to provide a step-by-step introduction to MFEM in simple model settings. The miniapps are more complex, and are intended to be more representative of the advanced usage of the library in physics/application codes. We recommend that new users start with the example codes before moving to the miniapps.

Select from the categories below to display examples and miniapps that contain the respective feature. *All examples support (arbitrarily) high-order meshes and finite element spaces.* The numerical results from the example codes can be visualized using the GLVis visualization tool (based on MFEM). See the GLVis website for more details.

Users are encouraged to submit any example codes and miniapps that they have created and would like to share.
Contact a member of the MFEM team to report bugs or post questions or comments.

Application (PDE): All | Finite Elements: All | Discretization: All | Solver: All

✓ All
Galerkin FEM
Mixed FEM
Discontinuous Galerkin (DG)
Discont. Petrov-Galerkin (DPG)
Hybridization
Static condensation
Isogeometric analysis (NURBS)
Adaptive mesh refinement (AMR)
Partial assembly

Example 1: Laplace Problem

This example code demonstrates the use of MFEM to define a simple iso… …n of the Laplace problem

$$-\Delta u = 1$$

with homogeneous Dirichlet boundary conditions. Specifically, we discre… …ing from the mesh (linear by default, quadratic for quadratic curvilinear mesh, NURBS for NURBS mesh, etc.)

The example highlights the use of mesh refinement, finite element grid functions, as well as linear and bilinear forms corresponding to the left-hand side and right-hand side of the discrete linear system. We also cover the explicit elimination of essential boundary conditions, static condensation, and the optional connection to the GLVis tool for visualization.

The example has a serial (ex1.cpp), a parallel (ex1p.cpp), and HPC versions: performance/ex1.cpp, performance/ex1p.cpp. It also has a PETSc modification in examples/petsc , a PUMI modification in examples/pumi and a Ginkgo modification in examples/ginkgo. Partial assembly and GPU devices are supported.

Example 2: Linear Elasticity

This example code solves a simple linear elasticity problem describing a multi-material cantilever beam. Specifically, we approximate the weak form of

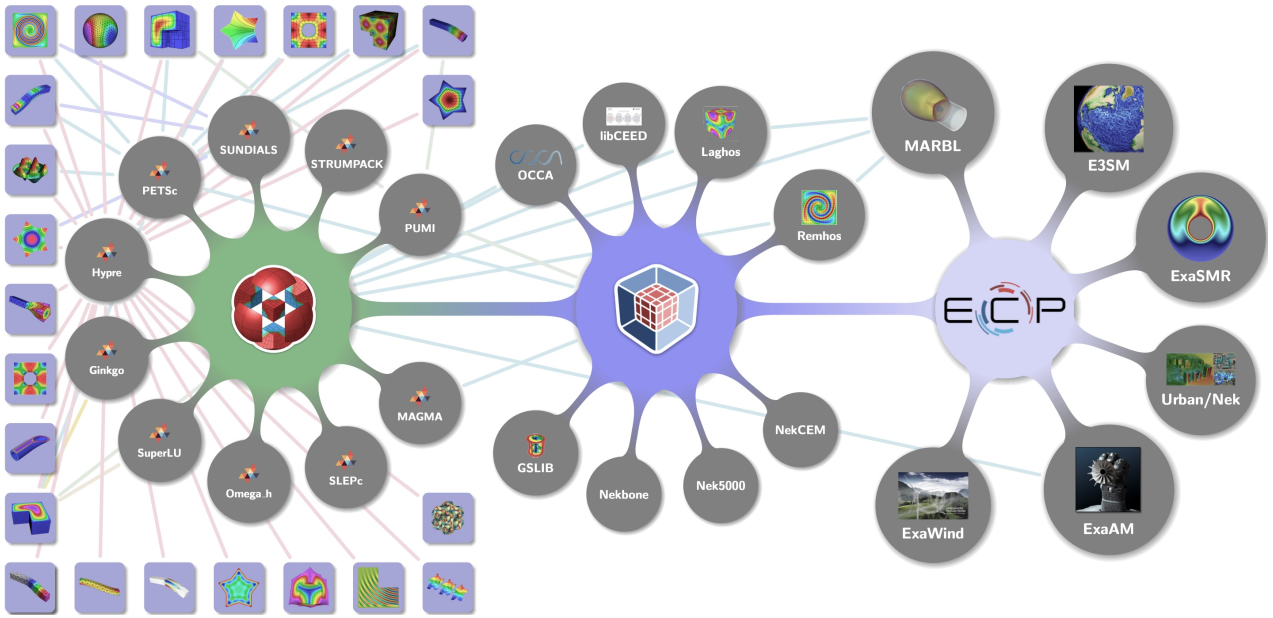
$$-\mathrm{div}(\sigma(\mathbf{u})) = 0$$

where

$$\sigma(\mathbf{u}) = \lambda\, \mathrm{div}(\mathbf{u})\, I + \mu\, (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

is the stress tensor corresponding to displacement field $\mathbf{u}$, and λ and μ are the material Lame constants. The boundary conditions are $\mathbf{u} = 0$ on the fixed part of the boundary with attribute 1, and $\sigma(\mathbf{u}) \cdot n = f$ on the remainder with f being a constant pull down vector on boundary elements with attribute 2, and zero otherwise. The geometry of the domain is assumed to be as follows:

mfem.org/examples



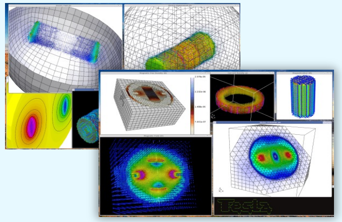
- 33 example codes, most with both serial + parallel versions
- Tutorials to learn MFEM features
- Starting point for new applications
- Show integration with many external packages, miniapps

Miniapps

More advanced, ready-to-use physics solvers

Volta, Tesla, Maxwell and Joule Miniapps

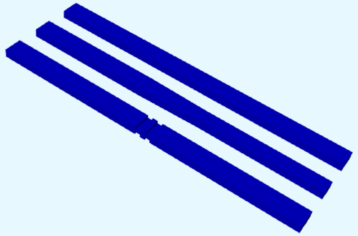
Static and transient electromagnetics



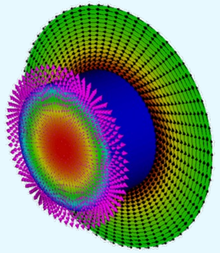
- **Volta** $-\nabla \cdot \epsilon \nabla \varphi = \rho - \nabla \cdot \vec{P}$
- **Tesla** $\nabla \times \mu^{-1} \nabla \times \vec{A} = \vec{J} + \nabla \times \mu^{-1} \mu_0 \vec{M}$
- **Maxwell** · *transient full-wave EM*

$$\frac{\partial(\epsilon \vec{E})}{\partial t} = \nabla \times (\mu^{-1} \vec{B}) - \sigma \vec{E} - \vec{J}$$

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}$$



- **Joule** · *transient magnetics + Joule heating*
- Arbitrary order elements + meshes
- Adaptive mesh refinement



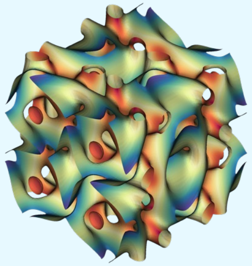
mfem.org/electromagnetics

Navier Miniapp

Transient incompressible Navier-Stokes equations

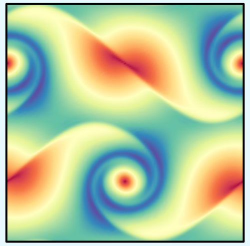
$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$$

$$\nabla \cdot \boldsymbol{u} = 0$$



3D Taylor-Green vortex, 7th order

- Arbitrary order elements
- Arbitrary order curvilinear mesh elements
- Adaptive IMEX (BDF-AB) time-stepping algorithm up to 3rd order
- State-of-the-art HPC performance
- GPU acceleration
- Convenient user interface

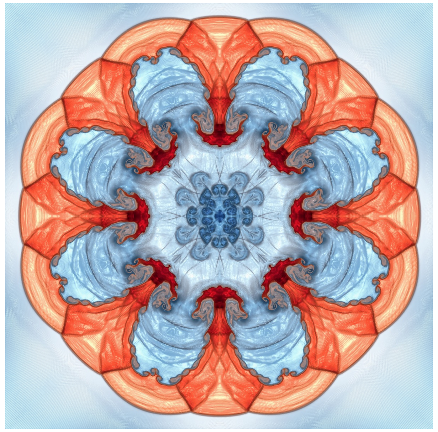


Double shear layer, 5th order, Re = 100000

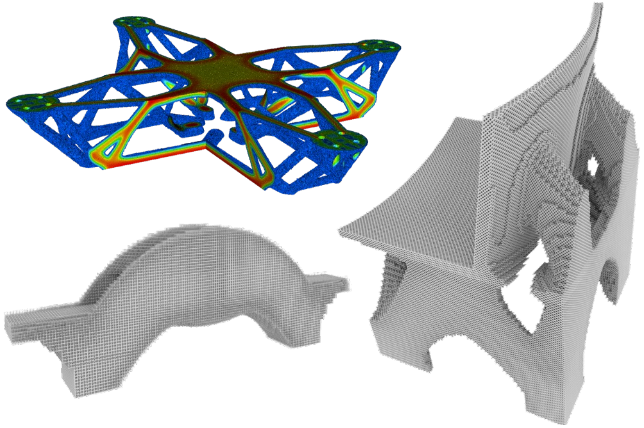
mfem.org/fluids

Applications

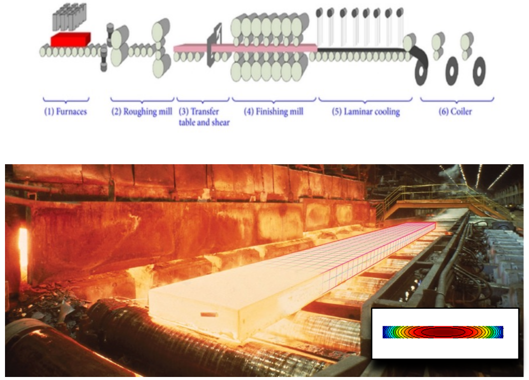
Some of the large-scale simulation codes powered by MFEM



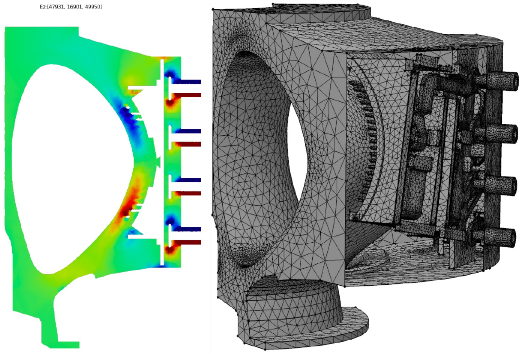
Inertial confinement fusion (BLAST)



Topology optimization for additive manufacturing (LiDO)



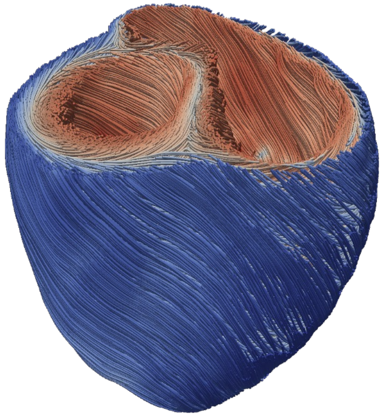
Hot strip mill slab modeling (U.S. Steel)



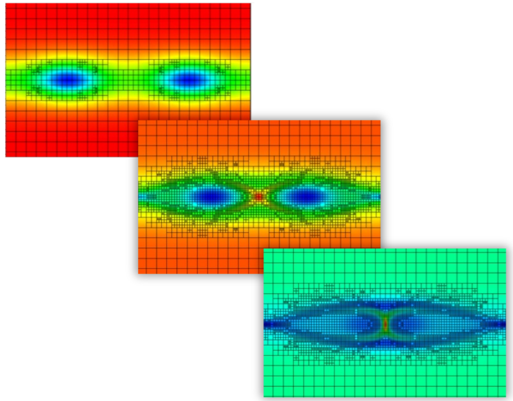
Core-edge tokamak EM wave propagation (SciDAC, RPI)

Electric aircraft design (RPI)

MRI modeling (Harvard Medical)



Heart modeling (Cardioid)

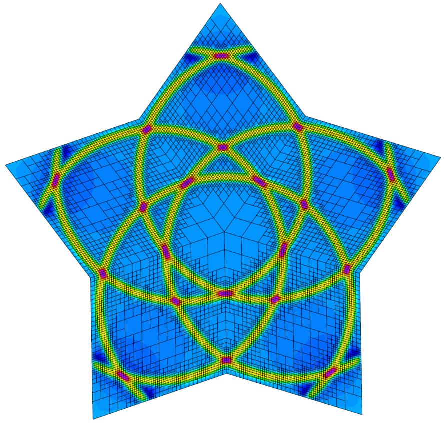


Adaptive MHD island coalescence (SciDAC, LANL)

Adaptive Mesh Refinement

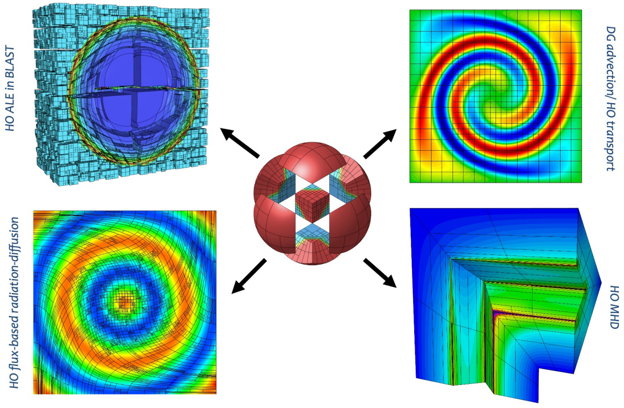
MFEM's unstructured AMR infrastructure

- **AMR on library level**
  - Conforming local refinement on simplex meshes
  - Non-conforming refinement for quad/hex meshes
  - Initial hp-refinement
- **General approach**
  - Any high-order finite element space, H1, H(curl), H(div), on any high-order curved mesh
  - 2D and 3D ▪ hexes, prisms, tets
  - Arbitrary order hanging nodes
  - Anisotropic refinement
  - Derefinement
  - Serial and parallel, including parallel load balancing
  - Independent of the physics
  - Easy to incorporate in applications



Example 15

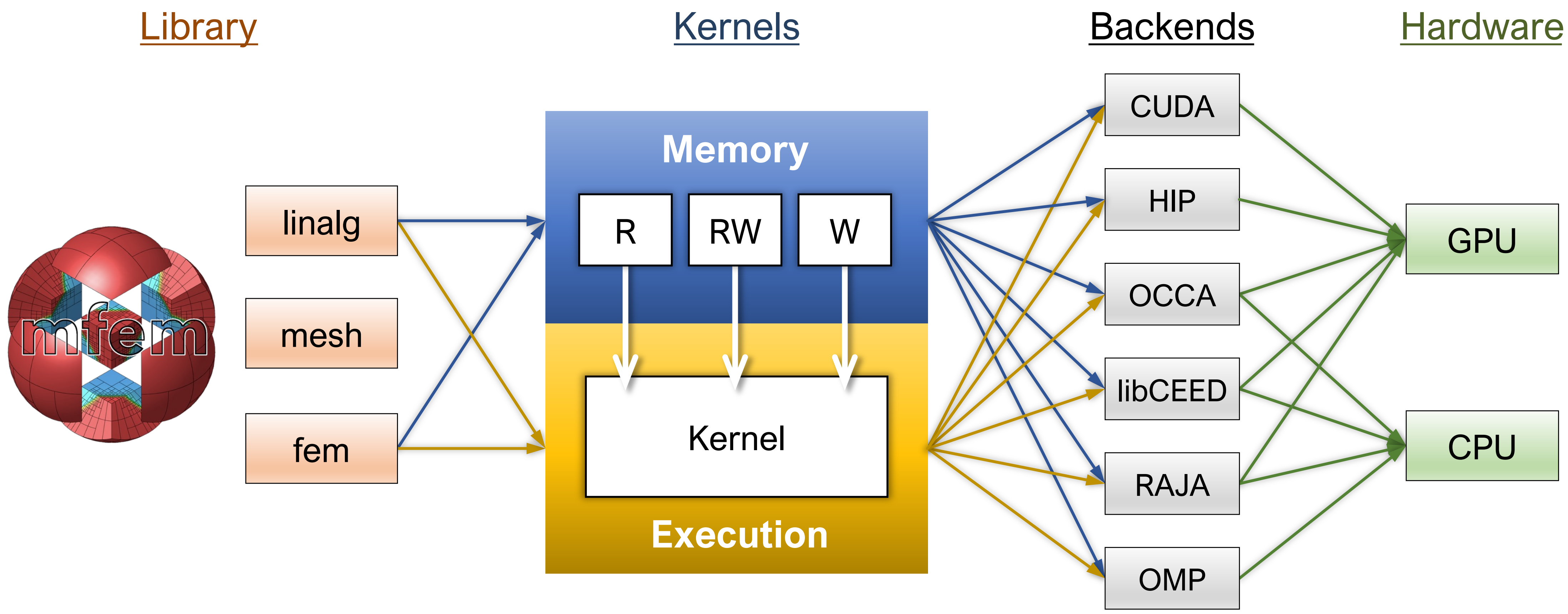
Shaper miniapp



Same AMR algorithms can be applied to a variety of high-order physics

GPU Support

MFEM has provided GPU acceleration for over 3 years (since mfem-4.0)

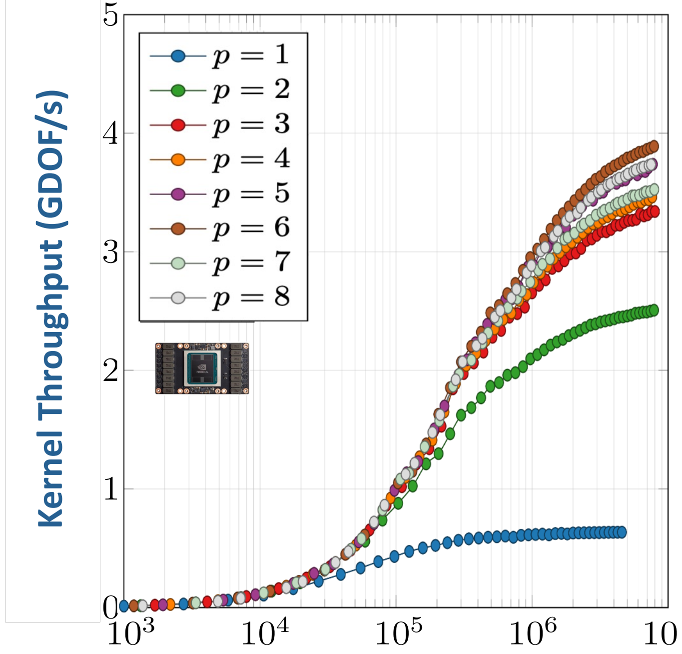


- **Backends are runtime selectable, can be mixed**
- **Coming soon: support for Intel/SYCL**

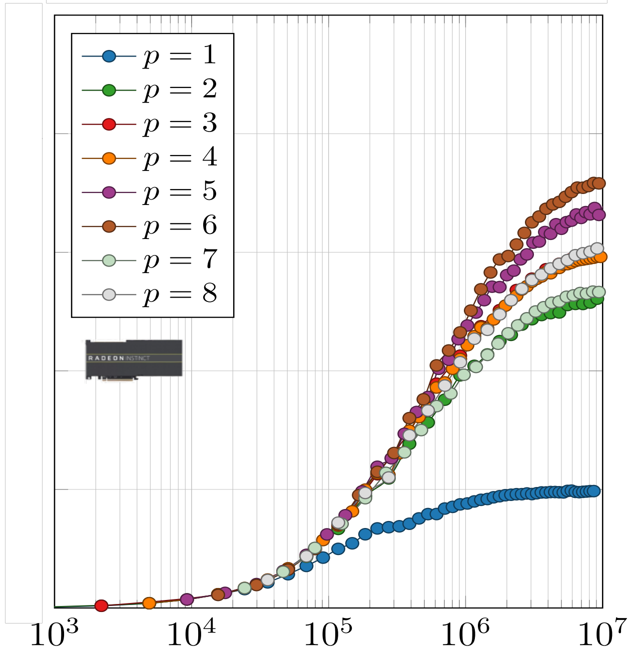
GPU Support

Recent GPU kernel improvements in MFEM

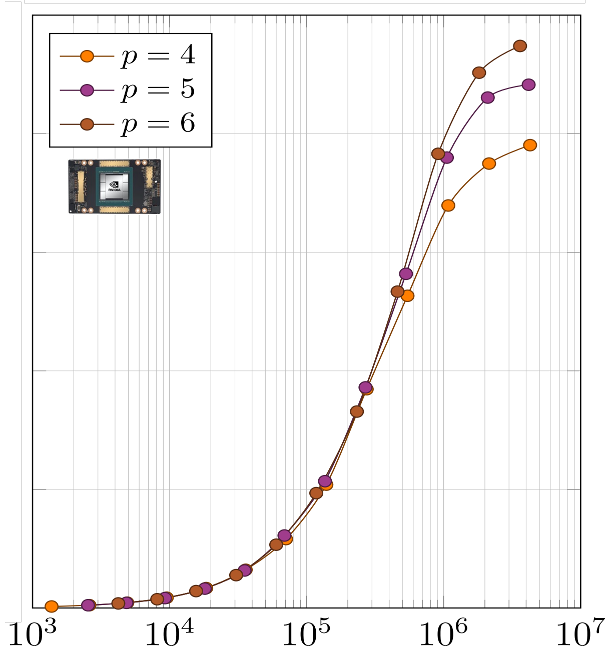
BP1 XFL vs FAST on V100

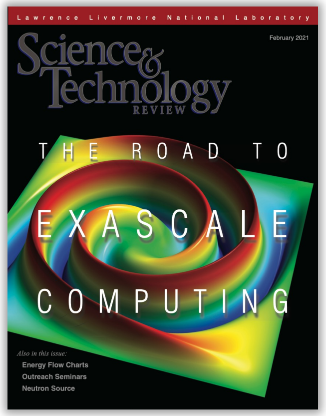


BP1 FAST on MI100



BP1 MMA on A100





- **New MFEM GPU kernels**
- **Have better strong scaling**
- **Perform on both NVIDIA + AMD GPUs**
- **Can utilize tensor cores on A100**

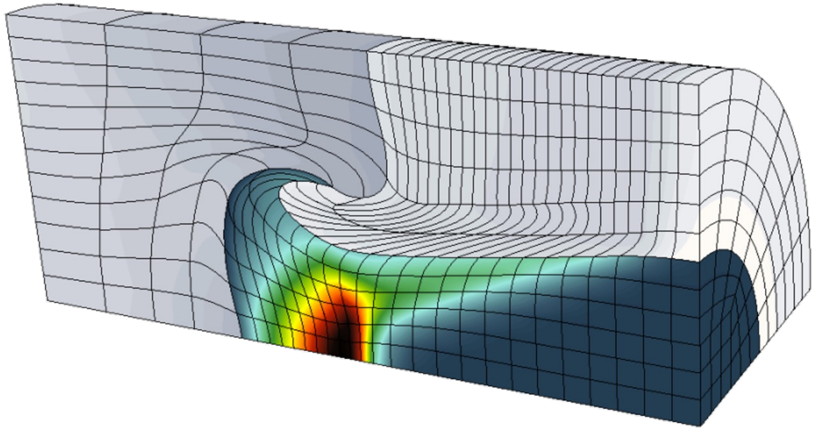
CEED
EXASCALE DISCRETIZATIONS

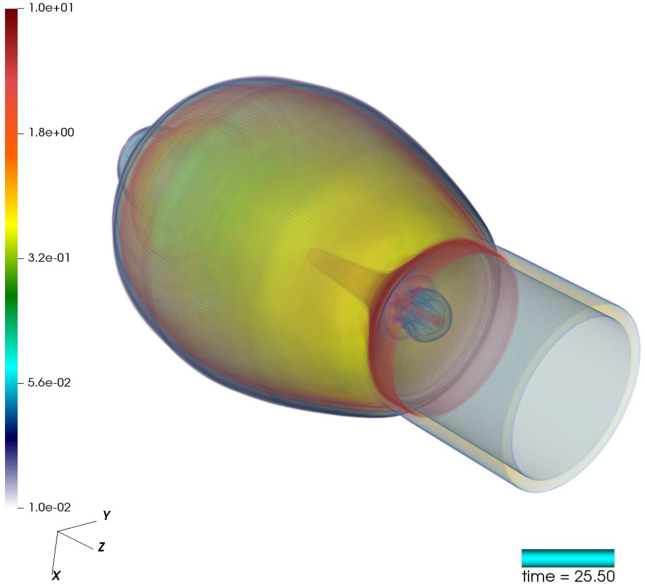
- Benchmarks (BPs)
- Miniapps (Laghos)
- libCEED

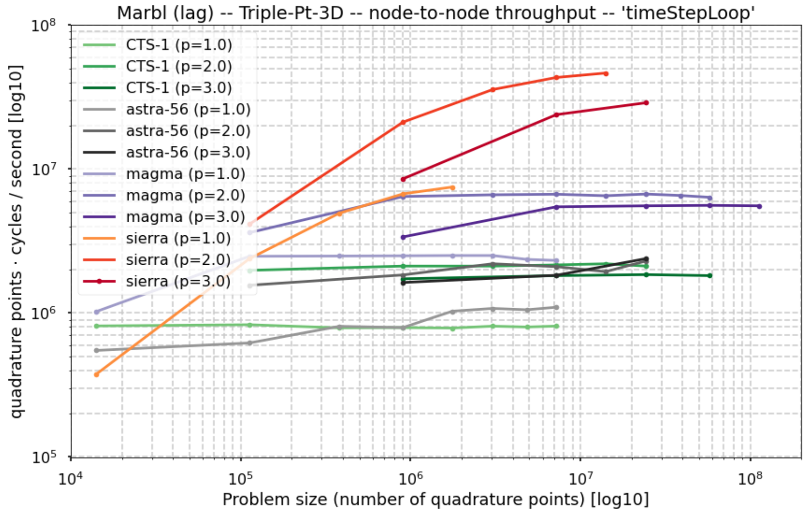
ceed.exascaleproject.org

GPU Support

BLAST Performance on Sierra







3D throughput: CPU-based systems vs NVIDIA V100 (Sierra)

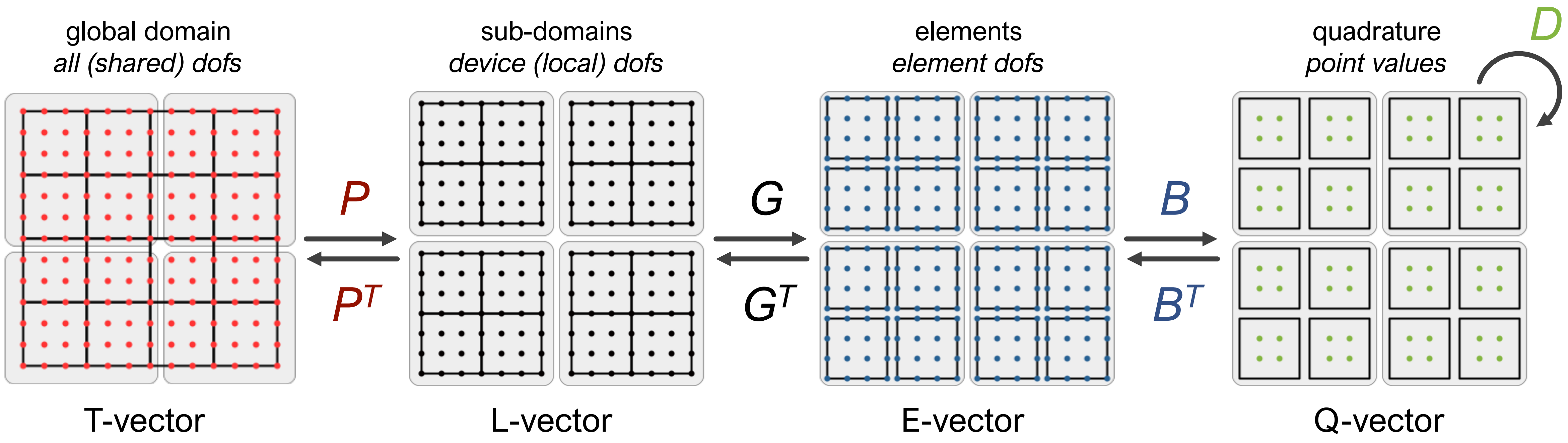
Phase	FA CPU	PA CPU	PA GPU	PA CPU/GPU Speedup
Time Loop	3854.16	2866.54	221.03	12.9
Lagrange	1773.68	1098.42	69.73	15.7
Remesh	557.98	366.24	42.67	8.5
Remap	1513.99	1393.34	100.95	13.8

3D ALE: 36-core CPU vs 4 GPUs (3 nodes)

FEM Operator Decomposition + Partial Assembly

Decompose A into parallel, mesh, basis, and geometry/physics parts

$$A = P^T G^T B^T D B G P$$

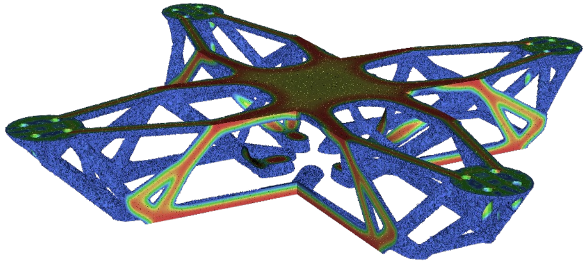


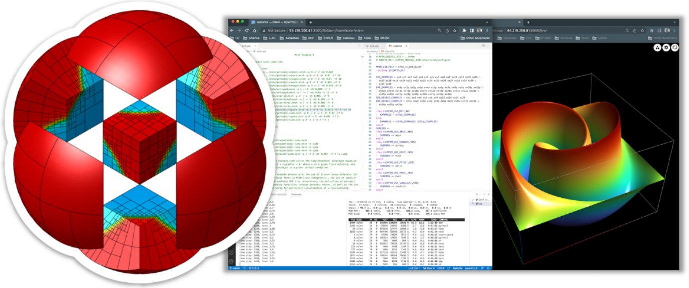
- **Partial assembly = store only D, evaluate B**
- **Optimal memory, near-optimal FLOPs compared to A**
- **AD-friendly**
- **MFEM + Enzyme**

Roadmap for Next Year

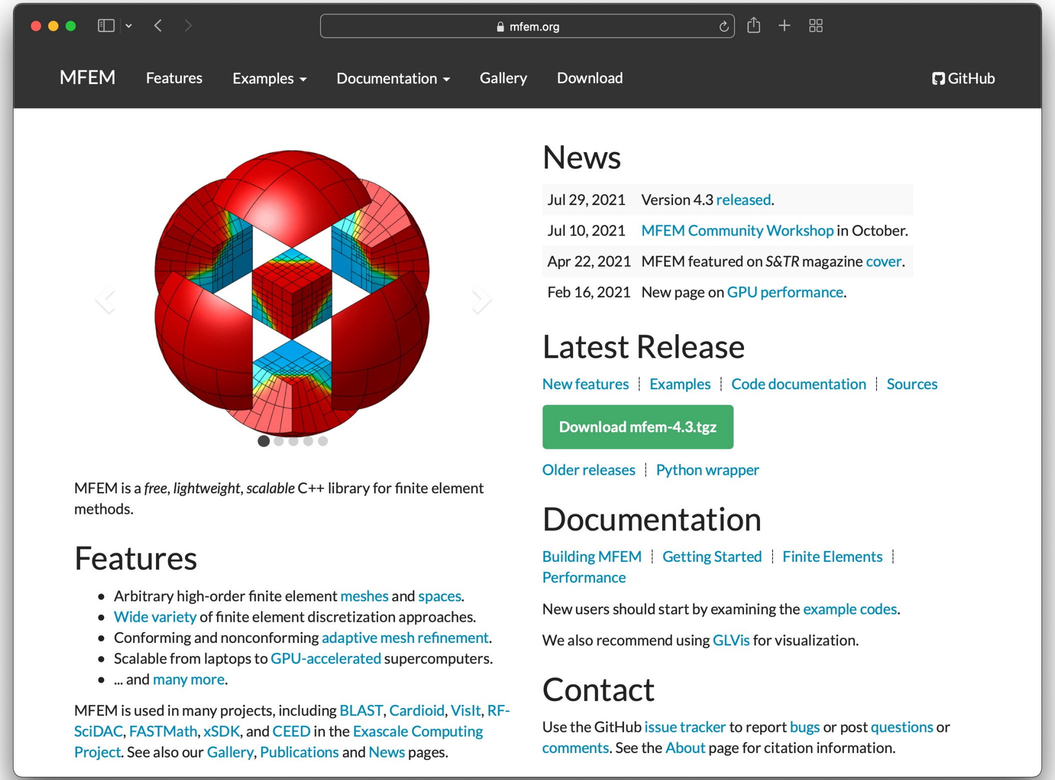
Plans for FY23

- **GPU support**
  - Performance on AMD GPU: Frontier + El Capitan
  - GPU ports of additional integrators · Continued performance improvement
- **Application needs**
  - Automatic differentiation · Design optimizations
  - H(div) preconditioning · Contact problems + solvers · Parallel re-partitioning
  - Cloud computing
  - MFEM in industry · Long-term sustainability
- **Code quality**
  - Improve documentation
  - Additional examples + miniapps
- **New releases**
  - v4.6 in May · v5.0 coming in FY24 – *expect breaking changes!*
- **What would you like to see?**
  - Slack: #meet-the-team · GitHub: github.com/mfem/mfem/issues · Email: mfem@llnl.gov





MFEM Resources



Website:
mfem.org

Software:
github.com/mfem

Publications:
mfem.org/publications

Email:
mfem@llnl.gov

- Contact us with questions + feedback
- Contribute to the code
- Explore our publications

Thank you from the MFEM team at LLNL!

Bob
Anderson
@rw-anderson

Julian
Andrej
@jandrej

Nabil
Atallah
@atallah727

John
Camier
@camierjs

Jakub
Cerveny
@jakubcerveny

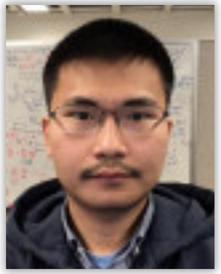
Dylan
Copeland
@dylan-copeland

Veselin
Dobrev
@v-dobrev

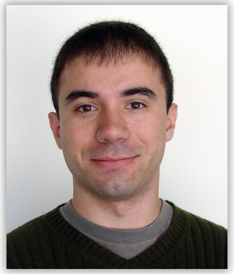
Yohann
Dudouit
@YohannDudouit

Aaron
Fisher
@acfisher

Milan
Holec
@homijan



Frank
Wang
@jwang125



Tzanio
Kolev
@tzanio

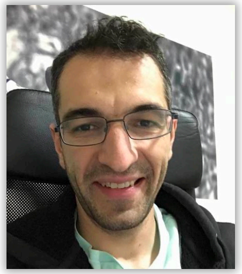
Boyan
Lazarov
@bslazarov

Ketan
Mittal
@kmittal2

Will
Pazner
@pazner

Socratis
Petrides
@psocratis

Mark
Stowell
@mlstowell



Vladimir
Tomov
@vladotomov

Chris
Vogl
@cjvogl

mfem.org

Lawrence Livermore National Laboratory