

The State of MFEM

MFEM Community Workshop
October 26, 2023

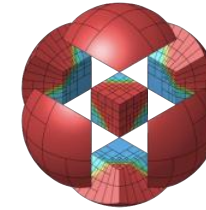
Tzanio Kolev
LLNL



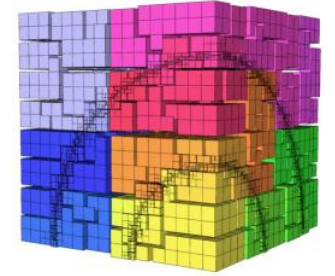
MFEM

Cutting-edge algorithms for powerful applications on HPC architectures

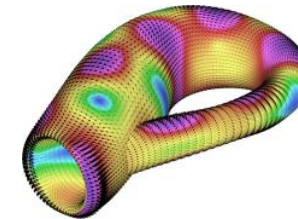
- **Flexible discretizations on unstructured grids**
 - Triangular, quadrilateral, tetrahedral and hexahedral meshes.
 - Local conforming and non-conforming AMR, mesh optimization.
 - Bilinear/linear forms for variety of methods: Galerkin, DG, DPG, ...
- **High-order and scalable**
 - Arbitrary-order 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. 100+ example codes and miniapps.
 - Part of FASTMath, ECP/CEED, xSDK, OpenHPC, E4S, ...



High-order curved elements



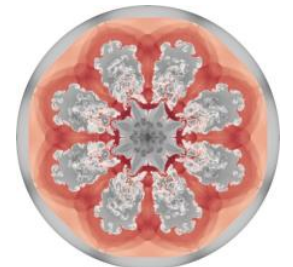
Parallel non-conforming AMR



Surface meshes



Heart modeling



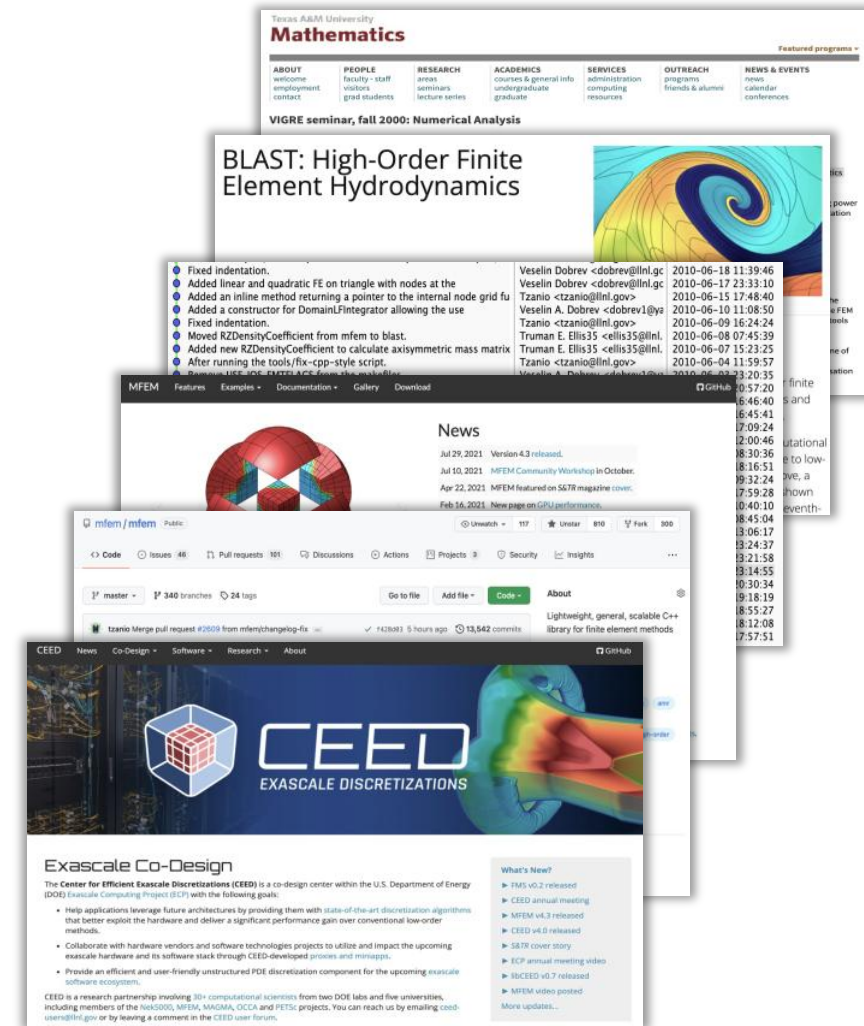
Compressible flow ALE simulations



A Brief History

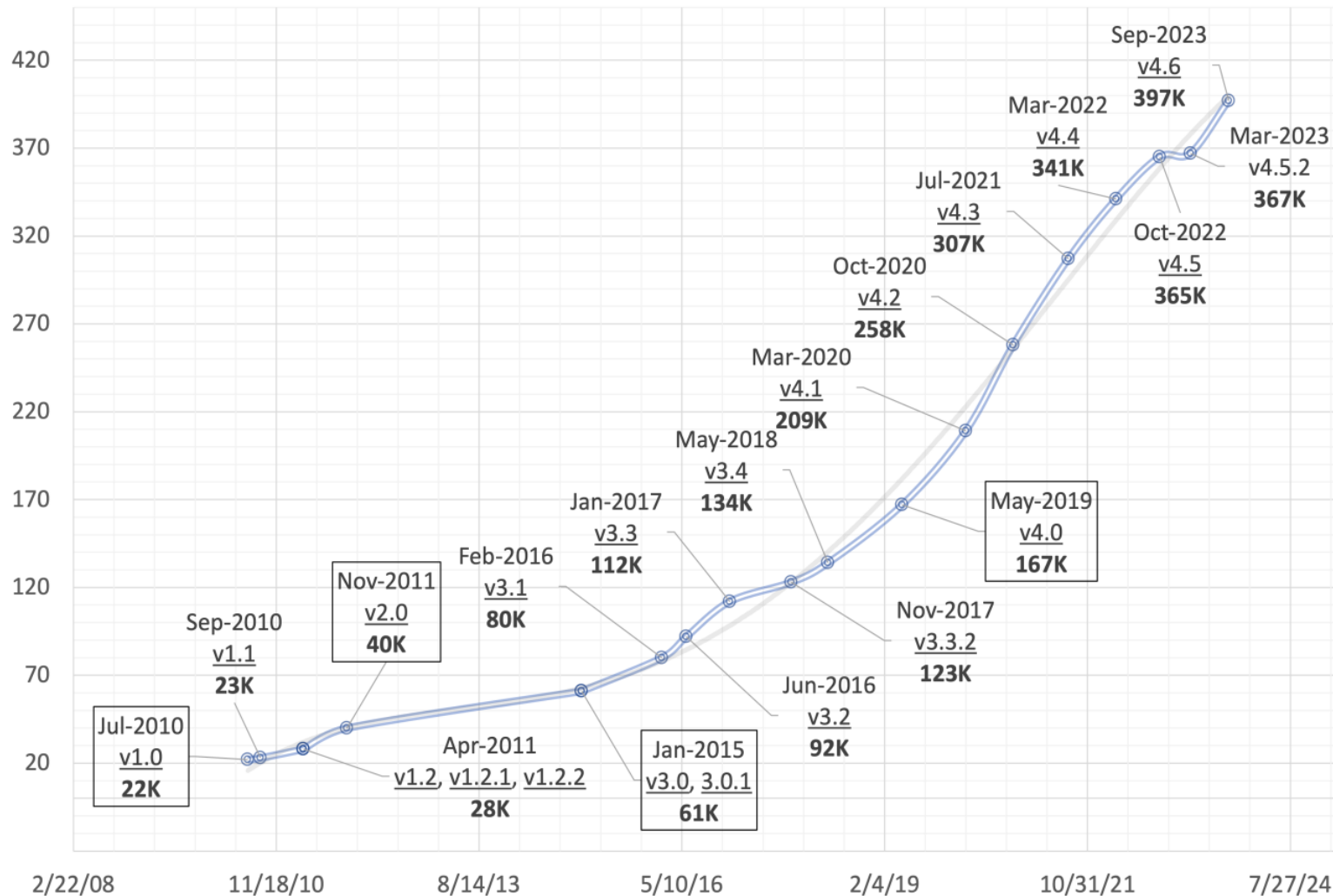
We've been doing this for a long time

- **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 created in May 2010
 - Some of the original contributors: @v-dobrev, @tzanio, @rieiben1, @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 is Growing

SLOC in MFEM releases over the last 13 years



mfem-4.6.tgz	v4.6	Sep 2023	3.6M	397K	
mfem-4.5.2.tgz	v4.5.2	Mar 2023	3.3M	367K	
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 is Growing

GitHub, downloads, and workshop stats

GitHub

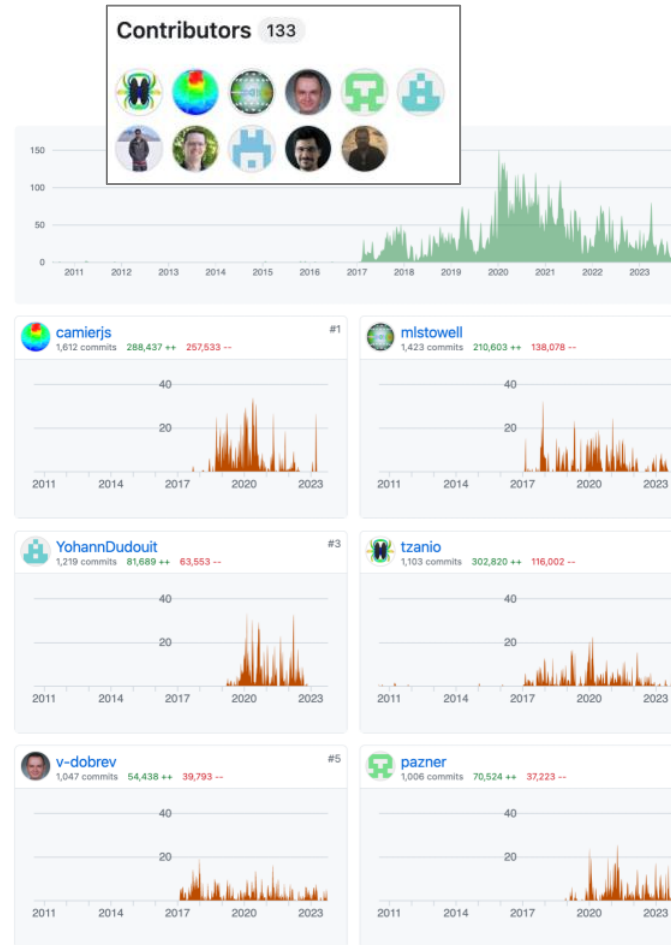
- **133** contributors
- **250** lines of code / day
- **629** people in the mfem organization – *join to contribute + receive announcements*
- **150** unique visitors / day
- **1390** stars – *thank you!*

Downloads

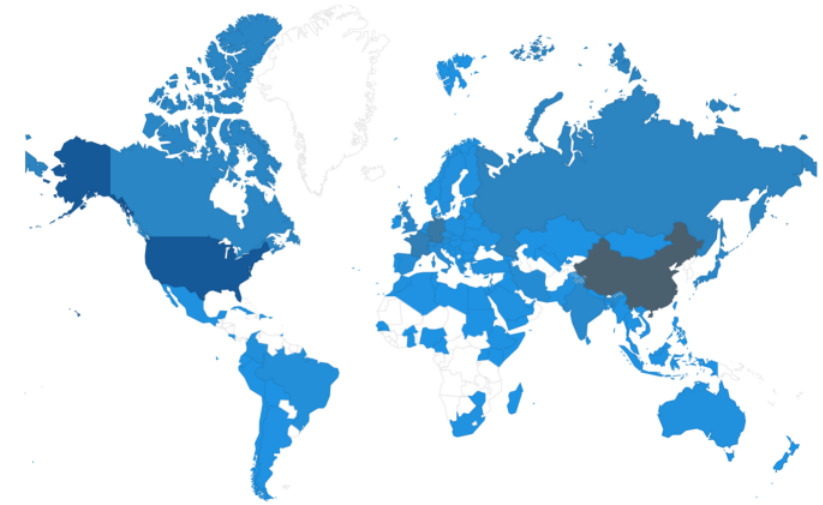
- **250** downloads + clones / day · **91K** / year
- **115** countries total

2023 Community Workshop

- **272** researchers
- **134** organizations
- **33** countries



Top contributors as of Oct 2023



MFEM has been downloaded from 115 countries

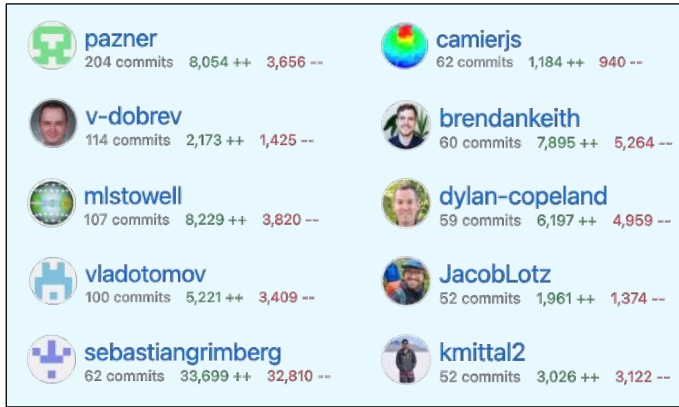
mfem.org	MFEM Community Workshop	October 2023
001 Aaron Fisher	Lawrence Livermore National Laboratory	fisher47@llnl.gov
002 Abdellatif Semmouri	FST, Sultan Moulay Slimane University	abd_semmouri@yahoo.fr
003 Abdelmajid Ezzine	Faculty of Sciences, Mohammed V University in Rabat	abdelmajid.ezzine@unsr.ac.ma
004 Abdesslam Ouaziz	University Sidi Mohammed Ben Abdellah	abdesslam.ouaziz1994@gmail.com
005 Achraf El Omari	Hassan II University of Casablanca	achraf.elomari-etu@etu.univh2c.ma
006 Achraf Zinihi	Faculty of Sciences and Technics, Moulay Ismail University of Mekne	a.zinihi@edu.unl.ac.ma
007 Adel Babbah	abdelmalek essadi university	a.babbah@uae.ac.ma
008 Aditya Parik	Utah State University	aditya.parik@usu.edu
009 Adolfo Rodriguez	Kappa Engineering	adantra@gmail.com
010 Adrian Butscher	Autodesk	adrian.butscher@autodesk.com
011 Ahdia Achabbak	Faculty of the science	ahdia.achabbak@etu.uue.ac.ma
012 Alberto Padovan	University of Illinois at Urbana-Champaign	padovan@illinois.edu
013 Alejandro Muñoz	Universidad de Granada	almona@ugr.es
014 Alex Lindsay	Idaho National Laboratory	alexander.lindsay@inl.gov
015 Alexander Blair	UK Atomic Energy Authority	alexander.blair@ukaea.uk
016 Alexander Grayver	ETH Zurich	agrayver@ethz.ch

2022 Community workshop had 258 registrations

Latest Releases Were Team Efforts

Versions 4.5.2 + 4.6 stats

- Released **Mar + Sep, 2023**
- 11** months in development
- 39** contributors
- 234** PRs merged
- 243** issues closed
- 58K** new lines of code
- 2180** number of commits
- Many new features:**
 - NURBS + TMOP meshing improvements
 - new H(div) matrix-free solver
 - SubMesh support for H(curl) and H(div)
 - HIP support for PETSc, SUNDIALS
 - stochastic PDEs, k-d tree, ultraweak DPG



Top 10 contributors to the last releases



The mfem-4.5.2+4.6 CHANGELOG has 45 entries



MFEM contributors on GitHub

Examples

The first stop for new users

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) Finite Elements Discretization Solver

Example 1: Laplace Problem

This example code demonstrates the use of MFEM to define a simple isotropic Laplace problem

$$-\Delta u = 1$$

with homogeneous Dirichlet boundary conditions. Specifically, we discretize the problem using a 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

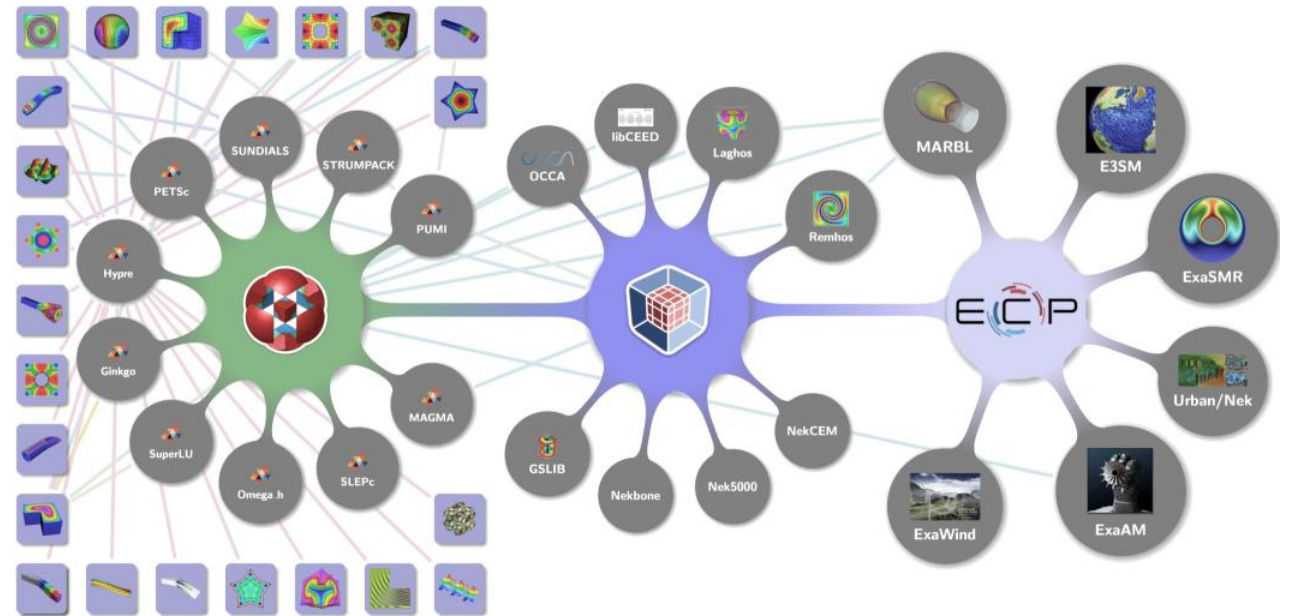
$$-\operatorname{div}(\sigma(\mathbf{u})) = 0$$

where

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

is the stress tensor corresponding to displacement field \mathbf{u} , and λ and μ are the material Lamé constants. The boundary conditions are $\mathbf{u} = 0$ on the fixed part of the boundary with attribute 1, and $\sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{f}$ on the remainder with \mathbf{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



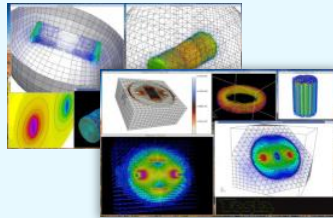
- 38 example codes, most with both serial + parallel versions
- Tutorials to learn MFEM features
- Starting point for new applications
- Show integration with many external packages, miniapps

Miniapps

More advanced, ready-to-use physics solvers

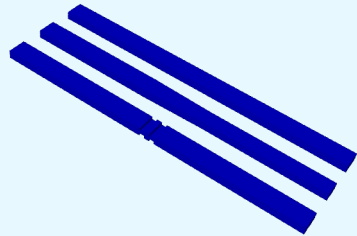
Volta, Tesla, Maxwell and Joule Miniapps *Static and transient electromagnetics*

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

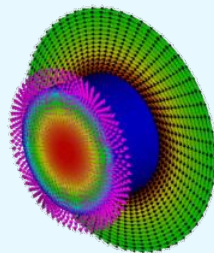


- **Maxwell** · *transient full-wave EM*

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



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



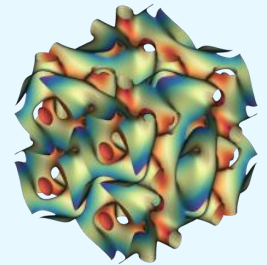
mfem.org/electromagnetics

Navier Miniapp

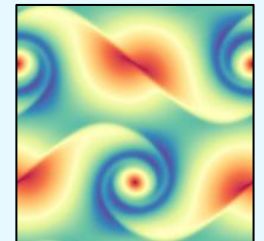
Transient incompressible Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f}$$
$$\nabla \cdot \mathbf{u} = 0$$

- 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



3D Taylor-Green vortex, 7th order

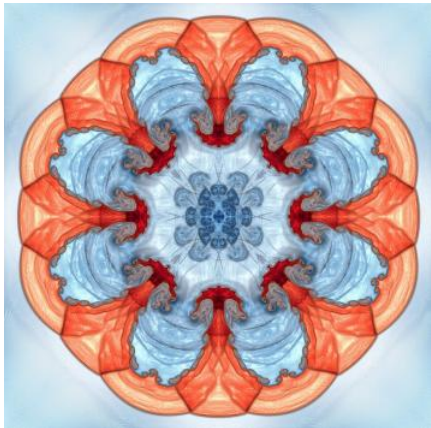


Double shear layer, 5th order, Re = 100000

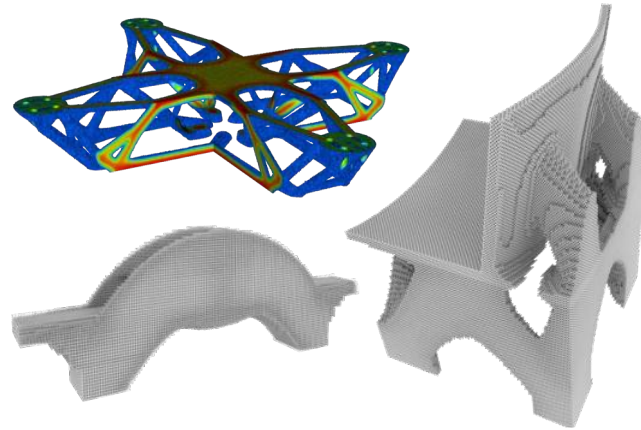
mfem.org/fluids

Applications

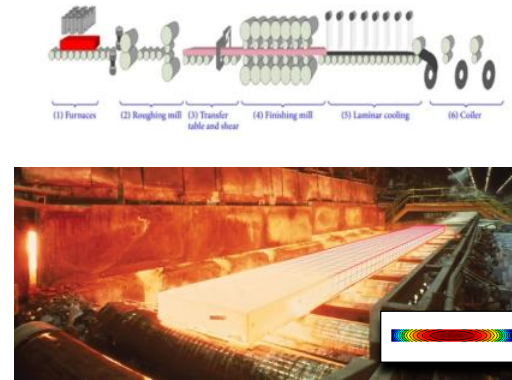
Some of the large-scale simulation codes powered by MFEM



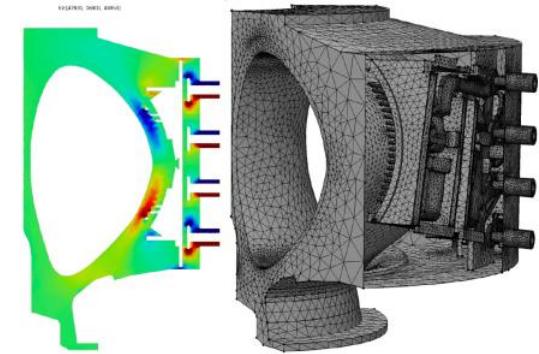
Inertial confinement fusion (BLAST, LLNL)



Topology optimization for additive manufacturing (LiDO, LLNL)



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



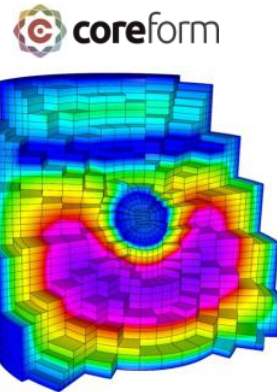
Core-edge tokamak EM wave propagation (SciDAC, RPI)



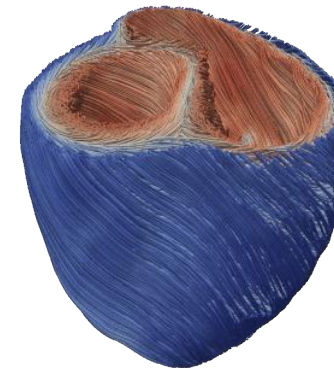
Electric aircraft design (RPI)



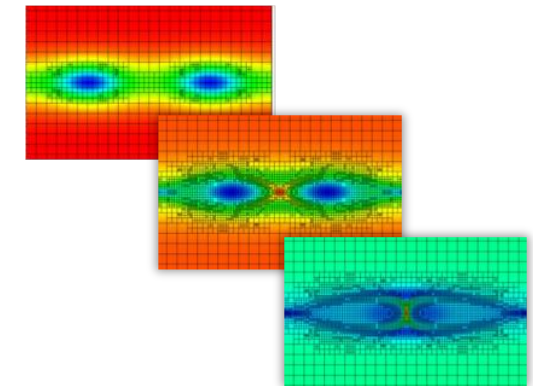
MRI modeling (Harvard Medical)



NURBS meshing and IGA (Coreform LLC, SBIR)



Heart modeling (Cardiod, LLNL/IBM)



Adaptive MHD island coalescence (SciDAC, LANL)

Adaptive Mesh Refinement

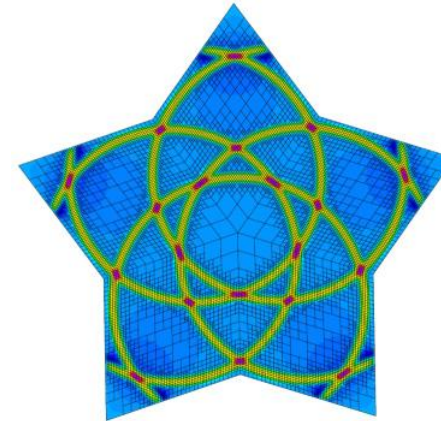
MFEM's unstructured AMR infrastructure

- **AMR on library level**

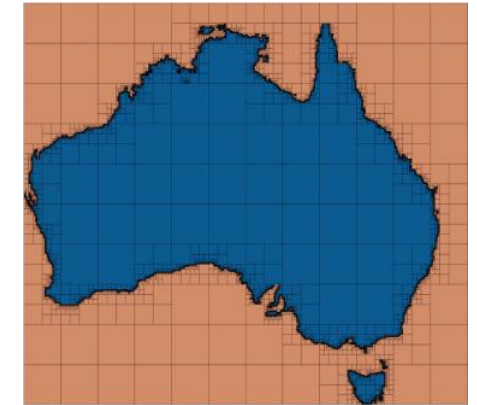
- Conforming local refinement on simplex meshes
- Non-conforming refinement for quad/hex meshes
- Initial hp-refinement

- **General approach**

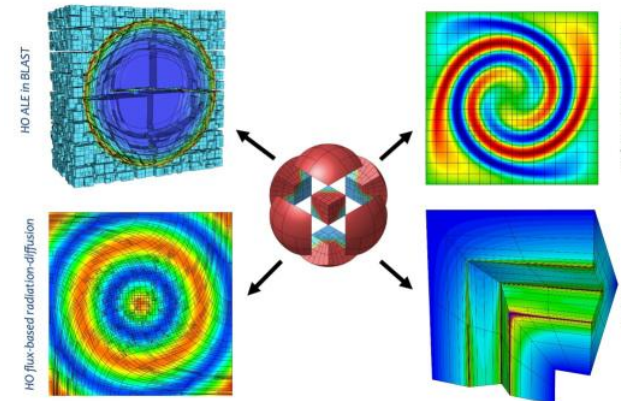
- Any high-order finite element space, H^1 , $H(\text{curl})$, $H(\text{div})$, on any high-order curved mesh
- 2D and 3D · hexes, prisms, tets
- Arbitrary order hanging nodes
- Anisotropic refinement
- Derefinement
- Serial and parallel, including parallel load balancing
- Independent of the physics
- Easy to incorporate in applications



Example 15



Shaper miniapp

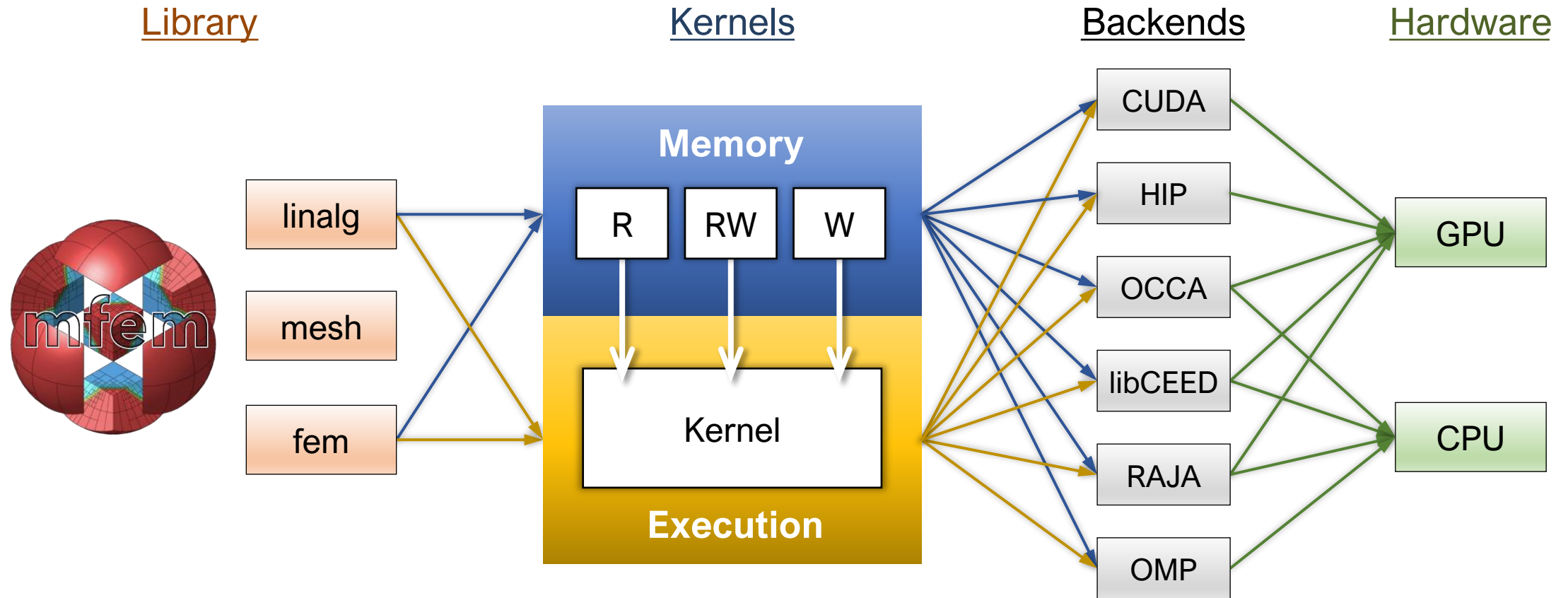


Same AMR algorithms can be applied to a variety of high-order physics



GPU Support

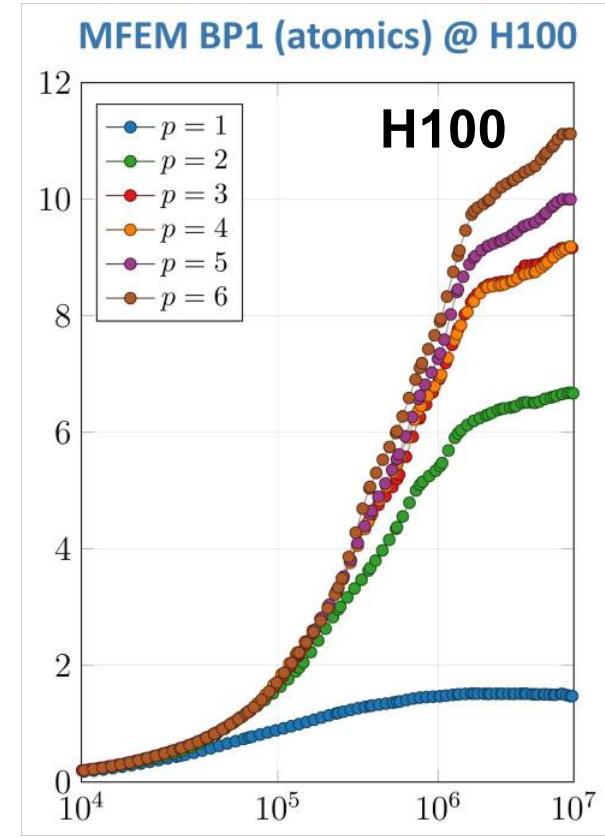
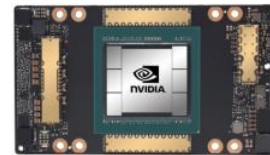
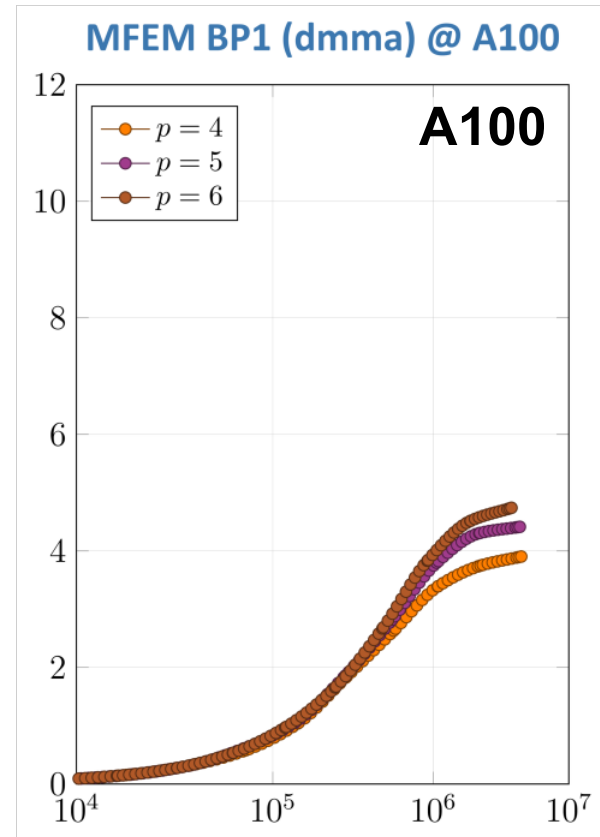
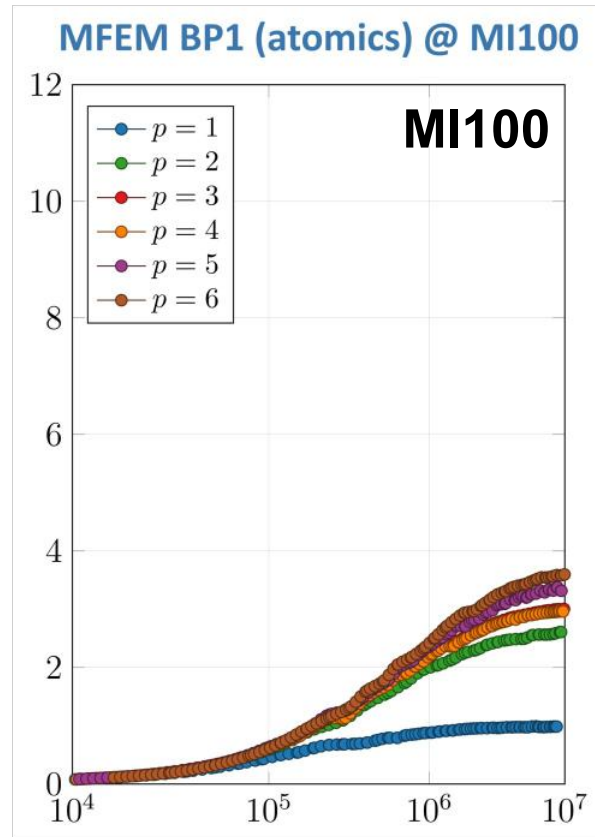
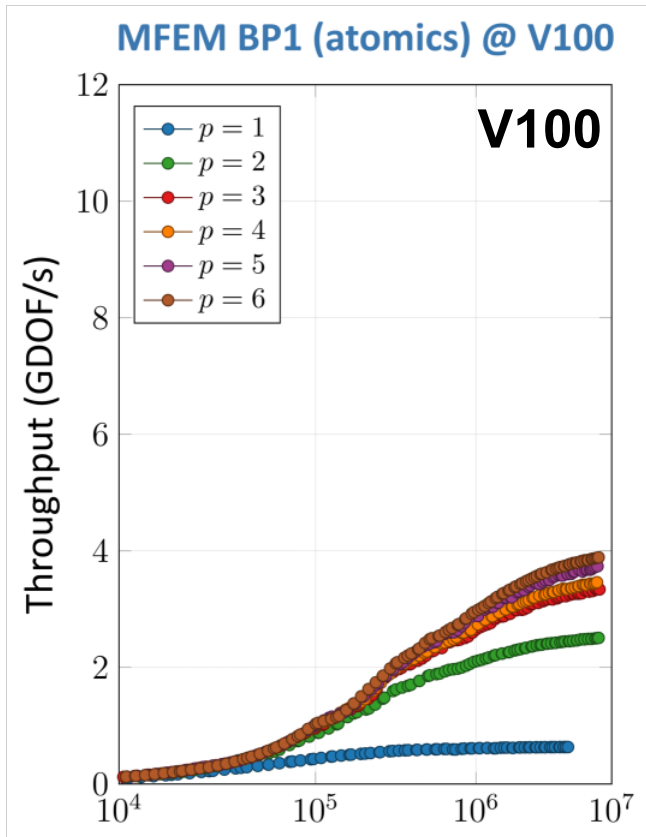
MFEM has provided GPU acceleration for over 5 years (since mfem-4.0)



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

GPU Support

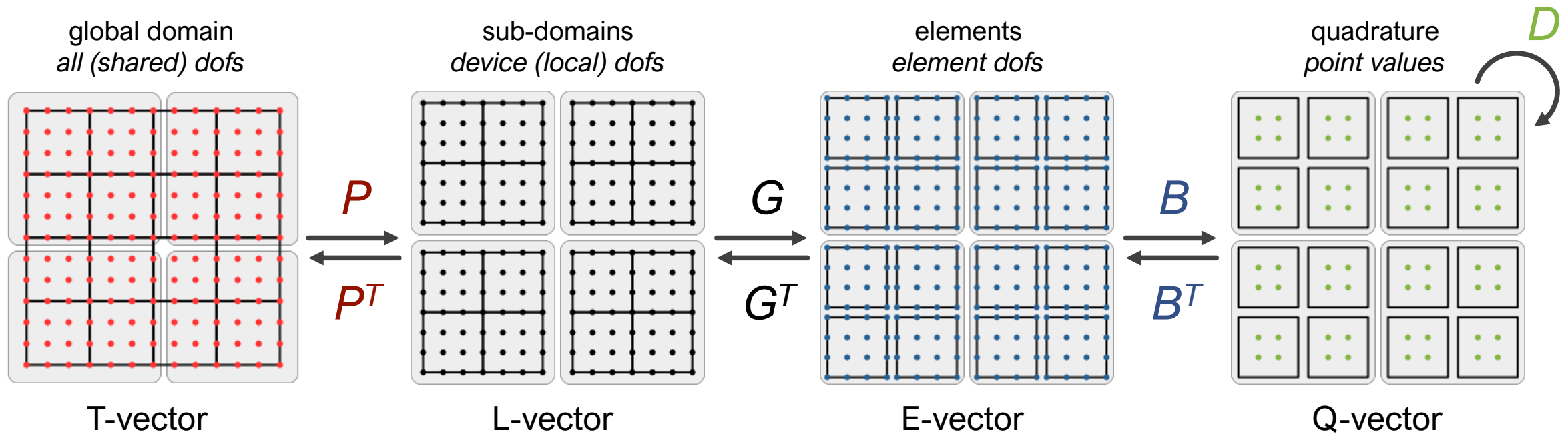
Recent GPU kernel improvements in MFEM



FEM Operator Decomposition + Partial Assembly

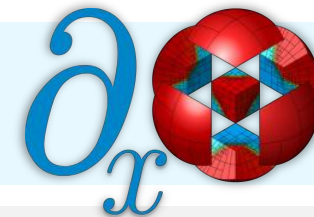
Decompose A into **parallel**, mesh, **basis**, and **geometry/physics** parts

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



- Partial assembly = store only D , evaluate B
- Optimal memory, near-optimal FLOPs compared to A

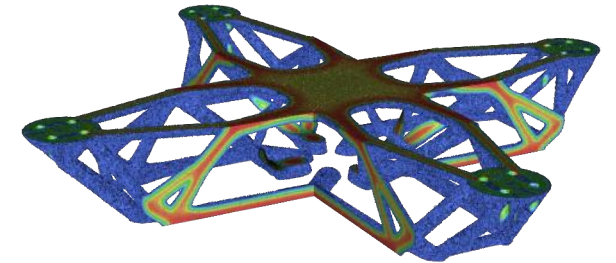
- AD-friendly
- MFEM + Enzyme



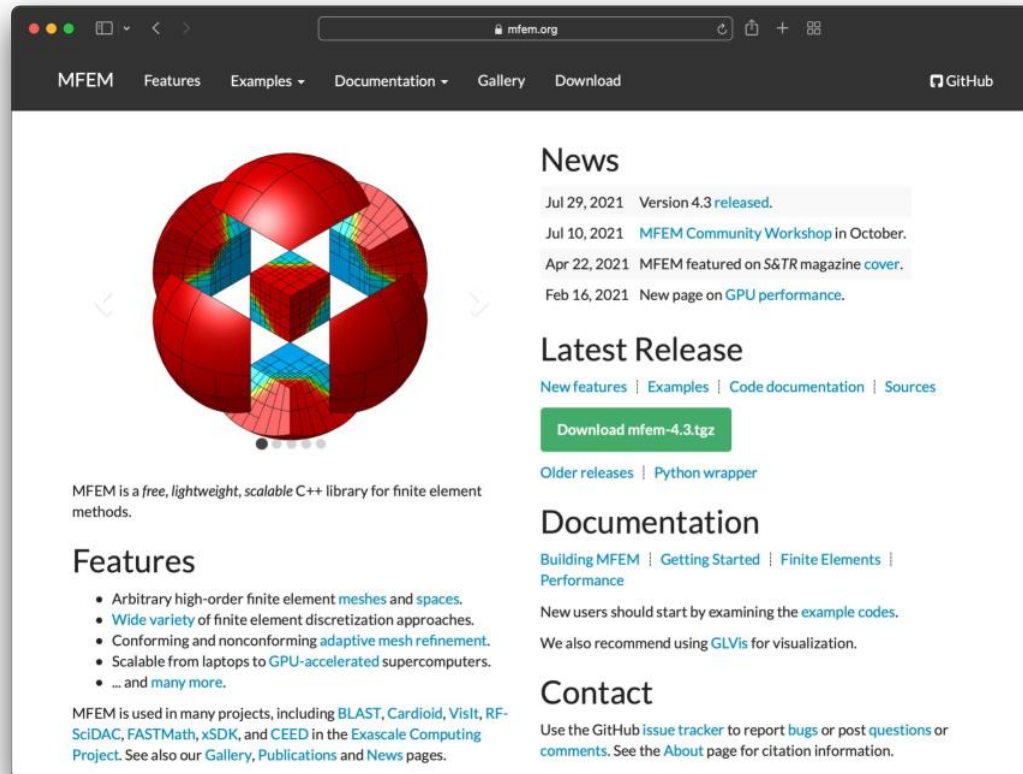
Roadmap for Next Year

Plans for FY24

- **GPU support**
 - Performance on AMD GPU: Frontier + El Capitan
 - Continued GPU porting and performance improvements
- **Applications**
 - Automatic differentiation · Design optimization
 - Compressible and incompressible flow
 - Fusion: both magnetic and ICF
 - Contact · 4D · mixed meshes · new collaborations
- **Code quality**
 - Improve documentation, testing
 - Additional examples + miniapps
- **New releases**
 - v4.7 in Mar · start work on v5.0 – *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



The screenshot shows the MFEM website homepage. The navigation bar includes links for MFEM, Features, Examples, Documentation, Gallery, Download, and GitHub. The main content area features a large 3D visualization of a red sphere with a blue and green mesh. Below this, there is a 'News' section with four entries: 'Version 4.3 released' (Jul 29, 2021), 'MFEM Community Workshop in October' (Jul 10, 2021), 'MFEM featured on S&TR magazine cover' (Apr 22, 2021), and 'New page on GPU performance' (Feb 16, 2021). The 'Latest Release' section highlights 'Download mfem-4.3.tgz' and provides links for 'New features', 'Examples', 'Code documentation', and 'Sources'. A 'Documentation' section includes links for 'Building MFEM', 'Getting Started', 'Finite Elements', and 'Performance'. The 'Features' section lists: 'Arbitrary high-order finite element meshes and spaces', 'Wide variety of finite element discretization approaches', 'Conforming and nonconforming adaptive mesh refinement', 'Scalable from laptops to GPU-accelerated supercomputers', and '...and many more.'. The 'Contact' section encourages using the GitHub issue tracker for bugs and questions.

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](#)



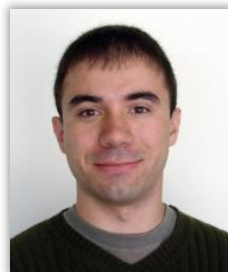
**Aaron
Fisher**
[@acfisher](#)



**Milan
Holec**
[@homijan](#)



**Frank
Wang**
[@jwang125](#)



**Tzanio
Kolev**
[@tzanio](#)



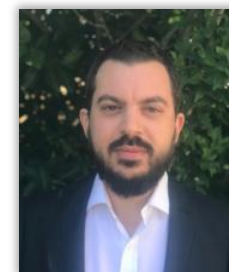
**Boyan
Lazarov**
[@bslazarov](#)



**Ketan
Mittal**
[@kmittal2](#)



**Will
Pazner**
[@pazner](#)



**Socratis
Petrides**
[@psocratis](#)



**Mark
Stowell**
[@mlstowell](#)



**Vladimir
Tomov**
[@vladotomov](#)



**Chris
Vogl**
[@cjvog1](#)

mfem.org



Disclaimer

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