The State of MFEM

MFEM Community Workshop
October 22, 2024

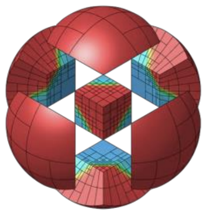
Tzanio Kolev
LLNL

LLNL-PRES-856457
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

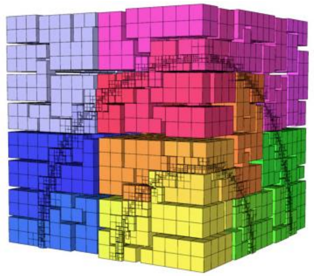
MFEM Finite Element Library

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 *new*
  - Bilinear/linear forms for variety of methods: Galerkin, DG, DPG, HDG, … *new*
- **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 *new*
  - 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 *new*
- **Open source**
  - Available on GitHub under BSD license, many example codes and miniapps
  - Part of FASTMath, ECP/CEED, xSDK, OpenHPC, E4S, …

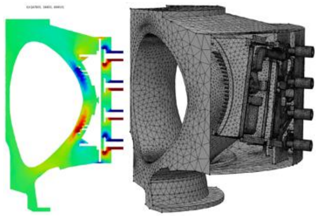


High-order curved elements



Parallel non-conforming AMR

ECP ASC CEED EXASCALE DISCRETIZATIONS FASTMATH xSDK



Core-edge tokamak

2020 R&D 100 FINALIST

Heart modeling

Compressible flow ALE simulations

MFEM: A modular finite element methods library, CAMWA 2020
High-performance finite elements with MFEM, IJHPCA, 2024

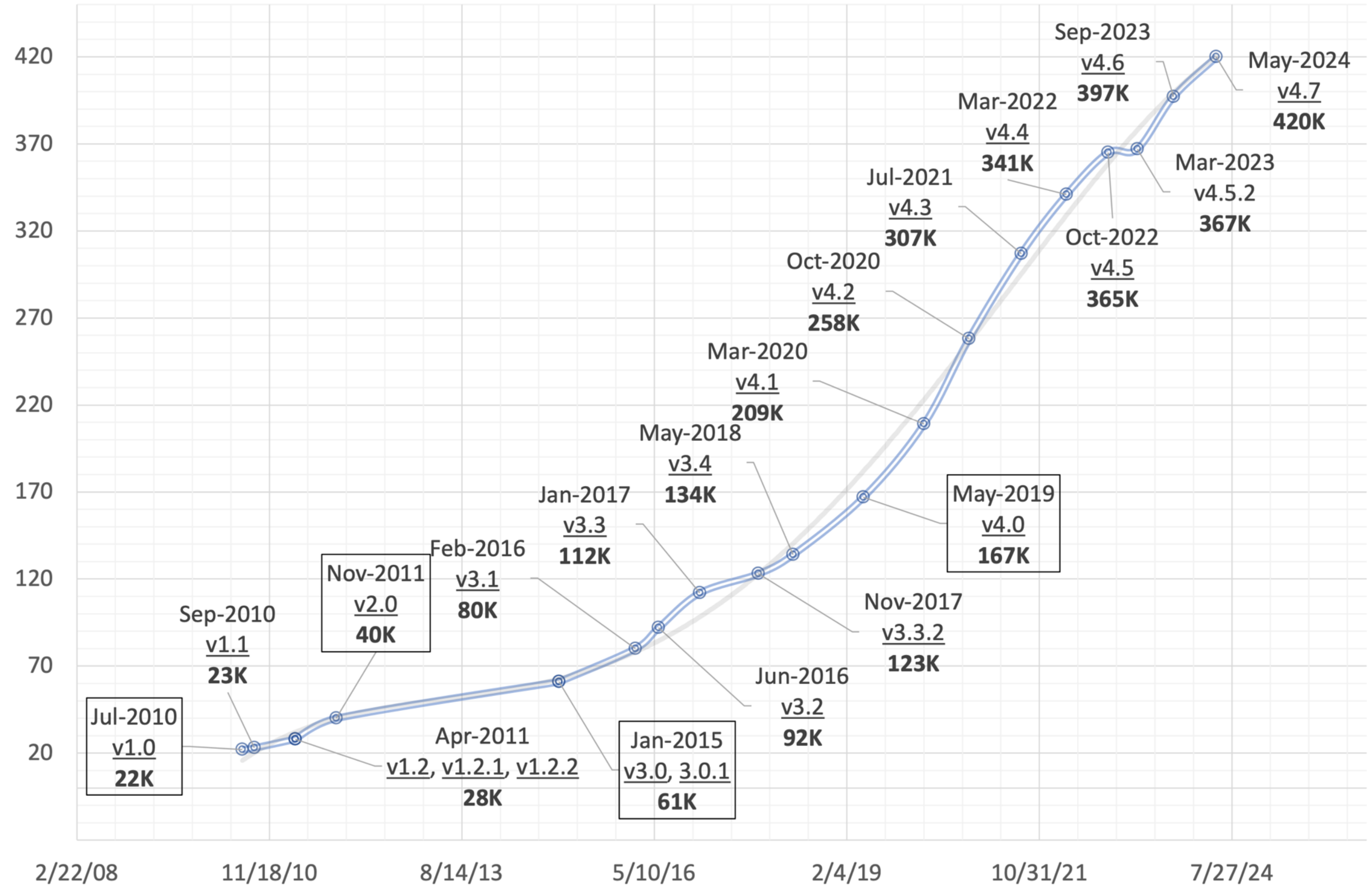
A Brief History

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**, **@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 exascale computing developments: GPUs, partial assembly, matrix-free
- **2024 – El Capitan, AD, Applications**

The Source Code is Growing

SLOC in MFEM releases over the last 14 years



mfem-4.7.tgz	v4.7	May 2024	3.8M	420K	
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

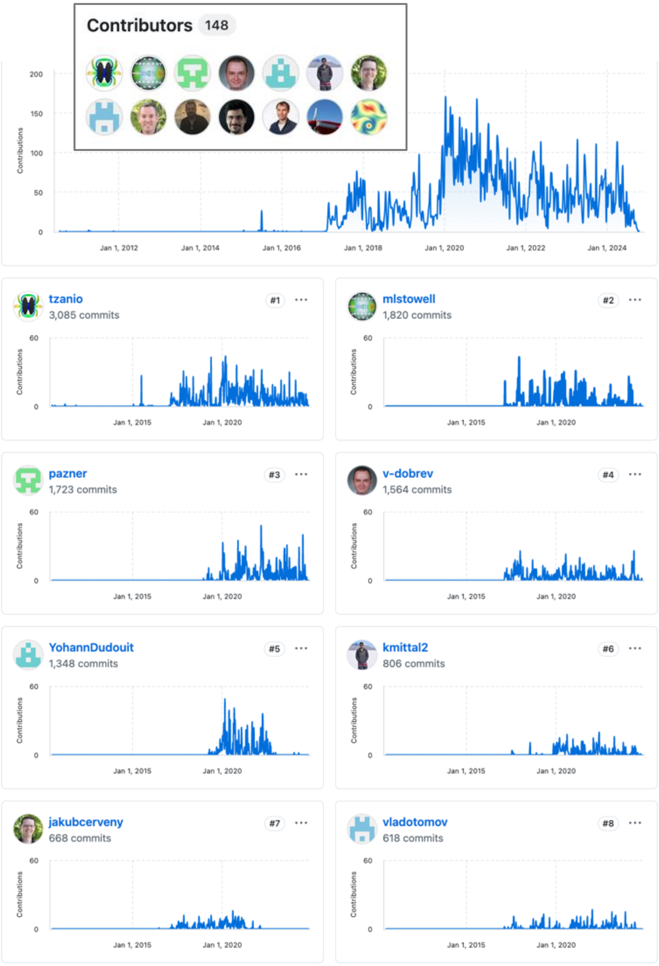
- **149** contributors
- **694** people in the mfem organization – *join to contribute + receive announcements*
- **1691** stars – *thank you!* ⭐ Starred 1.7k

Downloads

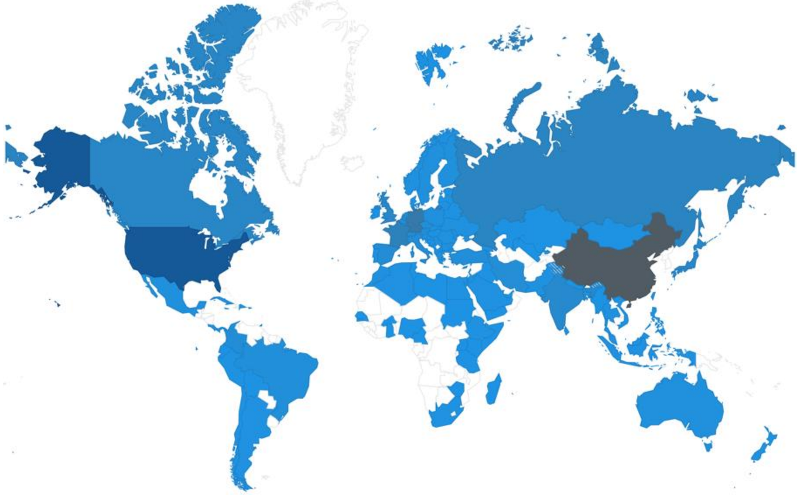
- **150+** unique visitors / day
- **200+** downloads + clones / day
- **100K+** / year
- **120+** countries total

2024 Community Workshop

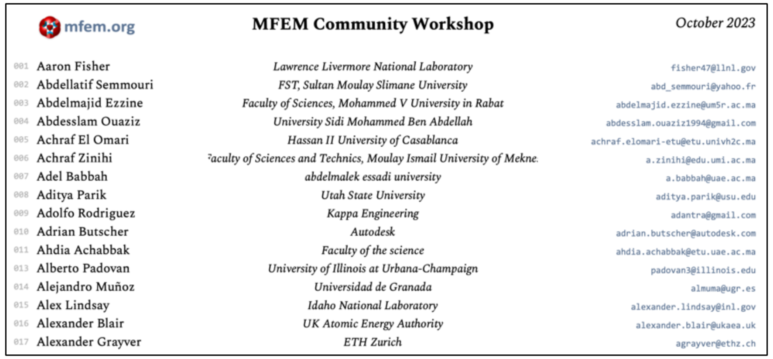
- **200+** researchers
- **100+** organizations
- **25+** countries



Top contributors as of Oct 2024



MFEM has been downloaded from 121 countries



mfem.org **MFEM Community Workshop** *October 2023*

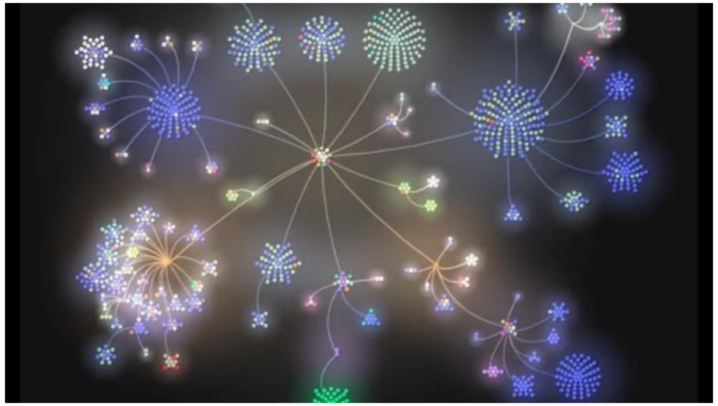
001	Aaron Fisher	*Lawrence Livermore National Laboratory*	fisher47@llnl.gov
002	Abdellatif Semmouri	*FST, Sultan Moulay Slimane University*	abd_semmouri@yahoo.fr
003	Abdelmajid Ezzine	*Faculty of Sciences, Mohammed V University in Rabat*	abdelmajid.ezzine@um5r.ac.ma
004	Abdesslam Ouaziz	*University Sidi 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	*aculty of Sciences and Technics, Moulay Ismail University of Mekne*	a.zinihi@edu.umi.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.uae.ac.ma
013	Alberto Padovan	*University of Illinois at Urbana-Champaign*	padovan3@illinois.edu
014	Alejandro Muñoz	*Universidad de Granada*	almuma@ugr.es
015	Alex Lindsay	*Idaho National Laboratory*	alexander.lindsay@inl.gov
016	Alexander Blair	*UK Atomic Energy Authority*	alexander.blair@ukaea.uk
017	Alexander Grayver	*ETH Zurich*	agrayver@ethz.ch

Community workshops have 200+ registrations

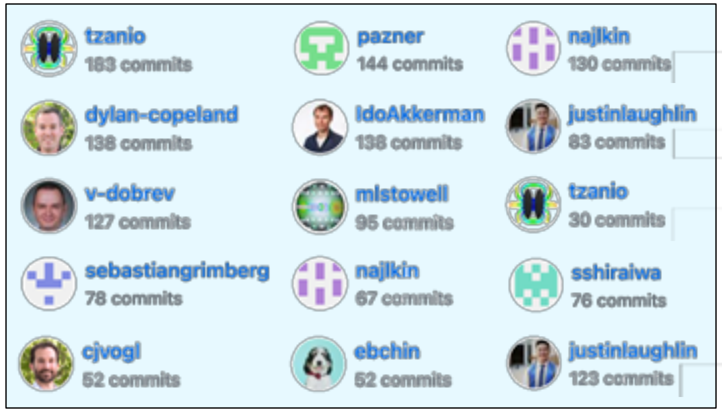
Latest Releases Were Team Efforts

Version 4.7 stats

- Released **May, 2024**
- **7** months in development
- **42** contributors
- **166** PRs merged
- **155** issues closed
- **25K** new lines of code
- **1694** commits
- **Many new features:**
  - — meshing, NURBS improvements
  - — cutFEM, hyperbolic conservation laws
  - — single precision support
  - — GPU-accelerated DG diffusion
  - — runtime device selection with *hypre*



The making of mfem-4.7 video on YouTube



Top contributors to latest releases

New GLVis releases!

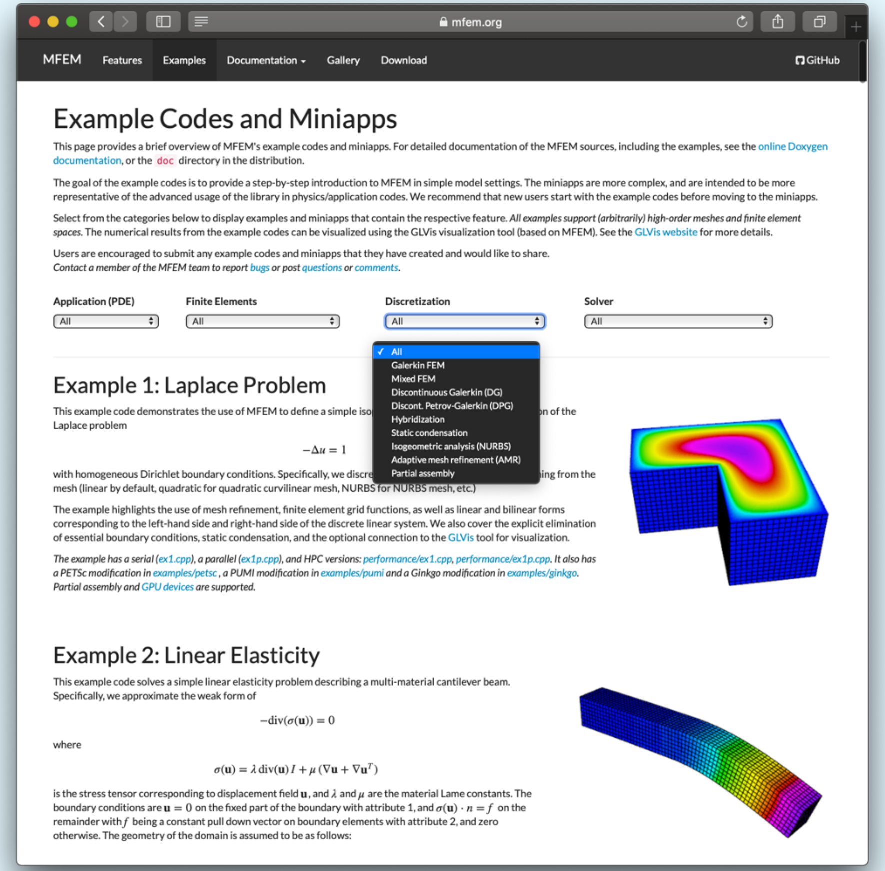
- **4.3** in August, **4.3.2** in September (more than 2 years since glvis-4.2)
- **Bugfixes, new features:**
  - — visualization of quadrature data
  - — support for integral elements
  - — 1D elements embedded in 2D/3D
  - — improved auto refinement
  - — new font and number formatting options
- Updated **pyglvis**, glvis.org/live

New PyMFEM releases!

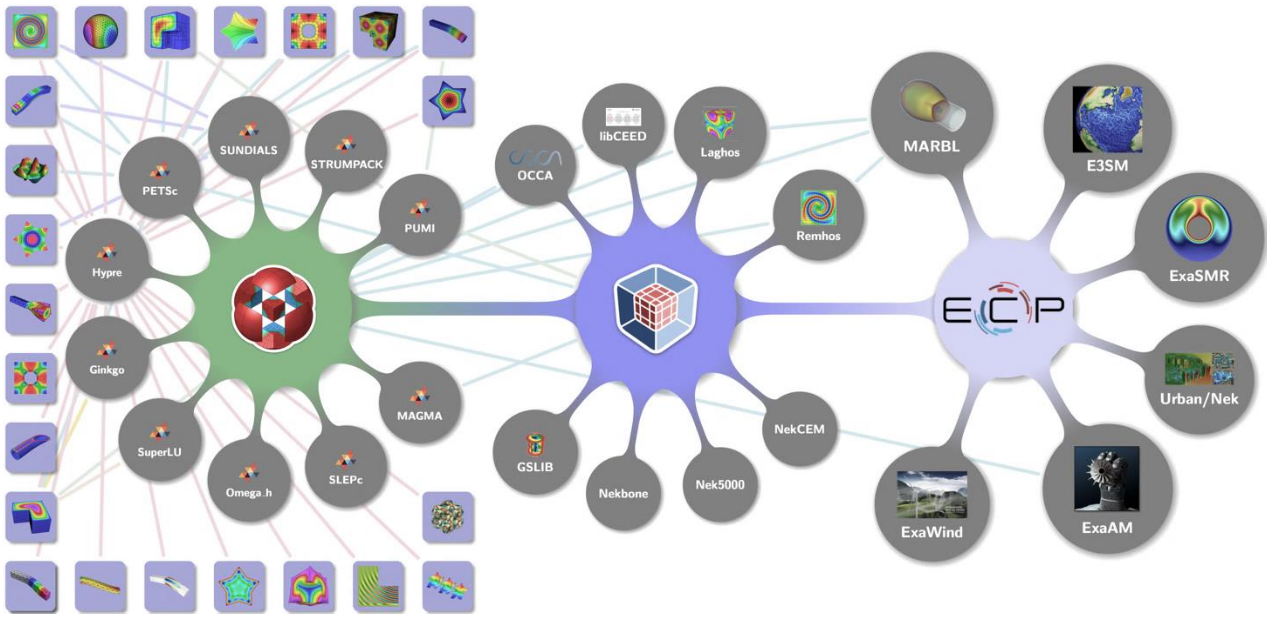
- **4.7** in August, **4.6** in January
  - — improved testing, Python examples

Examples

The first stop for new users



mfem.org/examples



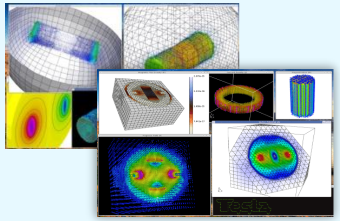
- 40 example codes, most with both serial + parallel versions
- Tutorials to learn MFEM features
- Starting point for new applications
- Show integration with many external packages, miniapps

Miniapps

More advanced, ready-to-use physics solvers

Volta, Tesla, Maxwell and Joule Miniapps

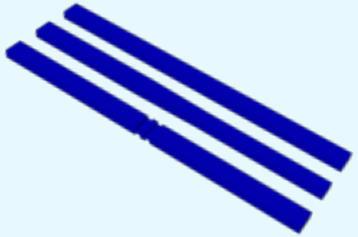
Static and transient electromagnetics



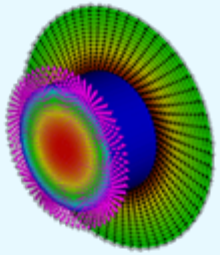
- **Volta** $-\nabla \cdot \epsilon \nabla \varphi = \rho - \nabla \cdot \vec{P}$
- **Tesla** $\nabla \times \mu^{-1} \nabla \times \vec{A} = \vec{J} + \nabla \times \mu^{-1} \mu_0 \vec{M}$
- **Maxwell** · *transient full-wave EM*

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

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



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



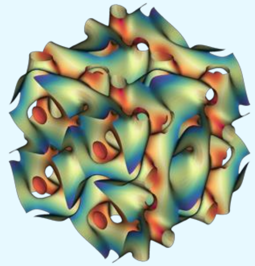
mfem.org/electromagnetics

Navier Miniapp

Transient incompressible Navier-Stokes equations

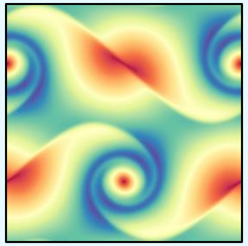
$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$$

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



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

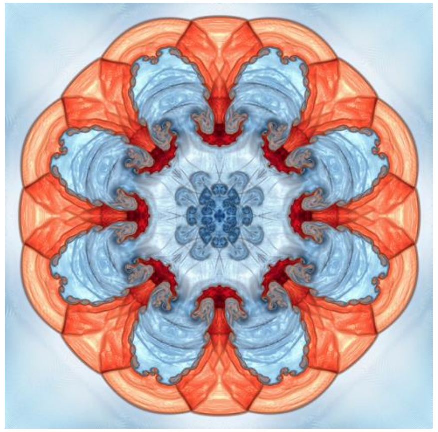


Double shear layer, 5th order, Re = 100000

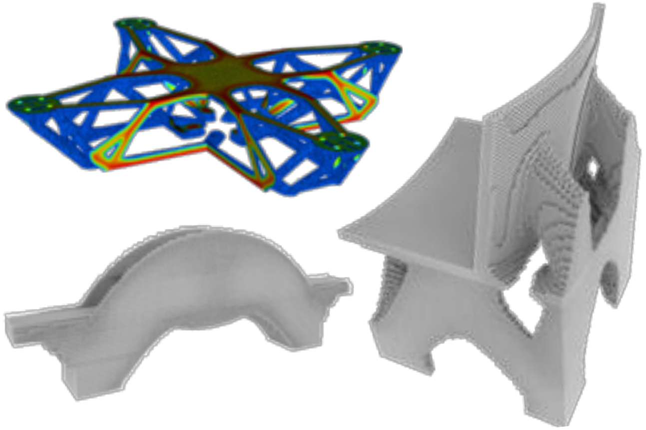
mfem.org/fluids

Applications

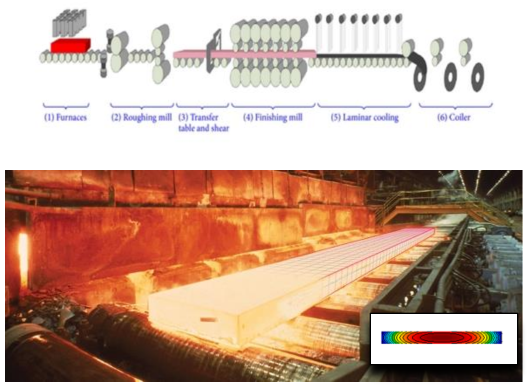
Some of the large-scale simulation codes powered by MFEM



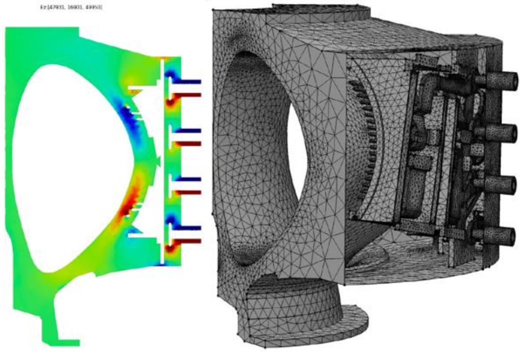
Inertial confinement fusion (BLAST, LLNL)



Topology optimization for additive manufacturing (LiDO, LLNL)



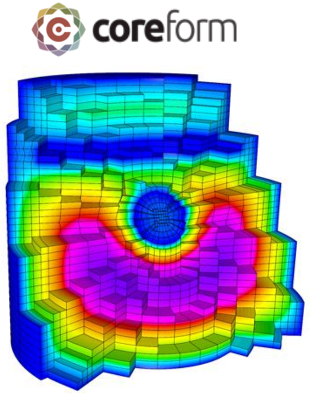
Hot strip mill slab modeling (U.S. Steel)



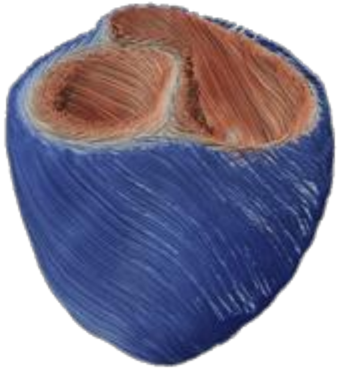
Core-edge tokamak EM wave propagation (SciDAC, RPI)

Electric aircraft design (RPI)

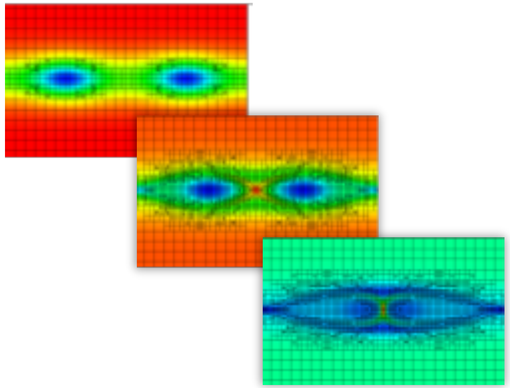
MRI modeling (Harvard Medical)



NURBS meshing and IGA (Coreform LLC, SBIR)



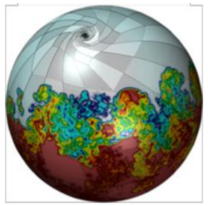
Heart modeling (Cardioid, LLNL/IBM)



Adaptive MHD island coalescence (SciDAC, LANL)

BLAST: High-Order ALE Multiphysics at LLNL

New algorithms enable us to solve more complex problems each year

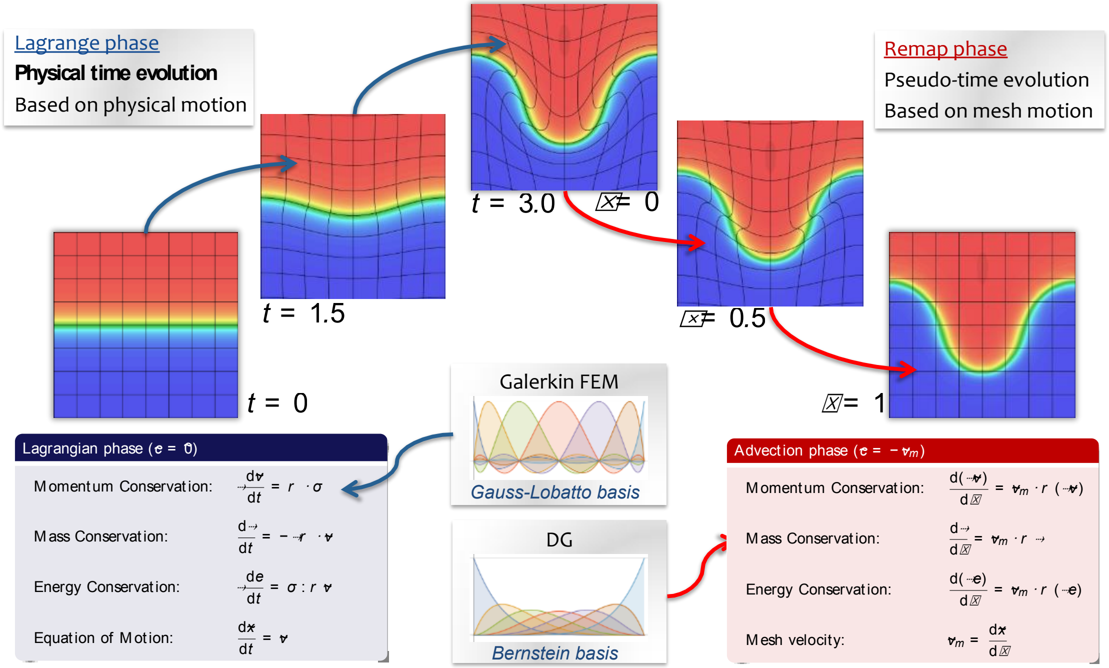


- **Large-scale multi-physics in BLAST code @ LLNL**
  - Compressible hydro + rad. diffusion + EM diffusion
  - Split ALE discretization
  - Explicit hydrodynamics + implicit diffusion
- **de Rham complex connect different physics**

$$H(grad) \xrightarrow{\nabla} H(curl) \xrightarrow{\nabla\times} H(div) \xrightarrow{\nabla\cdot} L_2$$

"nodes"	"edges"	"faces"	"elems"
High-order kinematics	High-order MHD	High-order rad. diffusion	High-order thermodynamics

- **High-order finite elements on high-order meshes**
  - Critical for robustness, symmetry, conservation
  - Better match for new hardware
  - Need new (interesting!) R&D for full benefits:
    - meshing, discretizations, linear solvers, AMR, …



R. Anderson, V. Dobrev, Tz. Kolev, R. Rieben and V. Tomov, High-Order Multi-Material ALE Hydrodynamics, SISC., 40(1):B32-B58, **2018**

V. Dobrev, Tz. Kolev, R. Rieben and V. Tomov, Multi-material closure model for high-order finite element Lagrangian hydrodynamics, IJNMF, 82(10), pp. 689–706, **2016**

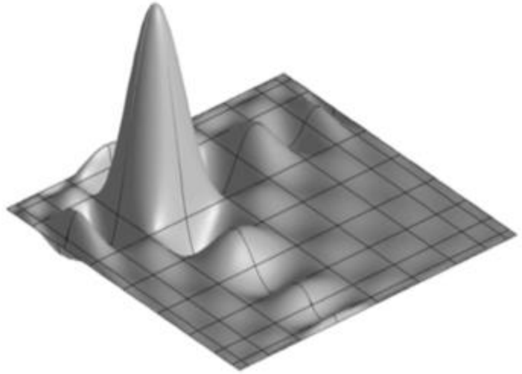
R. Anderson, V. Dobrev, Tz. Kolev and R. Rieben, Monotonicity in High-Order Curvilinear Finite Element ALE Remap, IJNMF, (77), pp. 249-273, **2015**

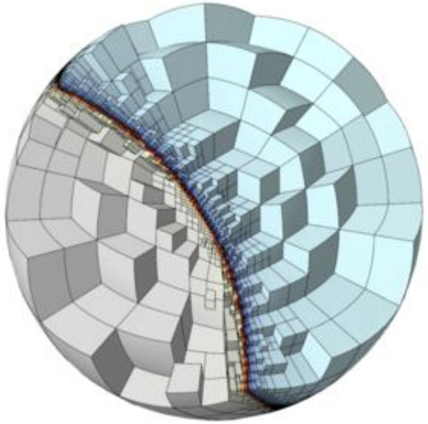
V. Dobrev, Tz. Kolev and R. Rieben, High-Order Curvilinear Finite Element Methods for Lagrangian Hydrodynamics, SISC, (34), pp.B606–B641, **2012**

Co-Design motives

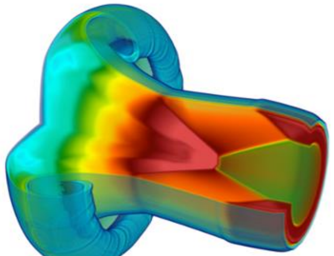
- PDE-based simulations on **unstructured grids**
- **high-order** and **spectral** finite elements

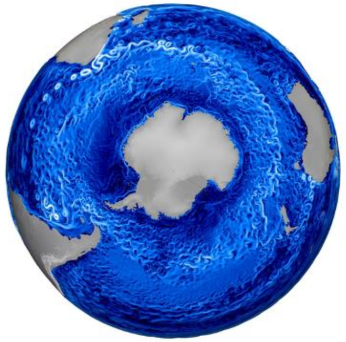
  ✓ *any order space on any order mesh* ✓ *curved meshes,*
  ✓ *unstructured AMR* ✓ *optimized low-order support*


10th order basis function

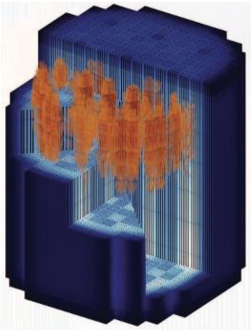

non-conforming AMR, 2nd order mesh

Target applications


Compressible flow (MARBL)


Climate (E3SM)


Urban systems (Urban)


Modular Nuclear Reactors (ExaSMR)

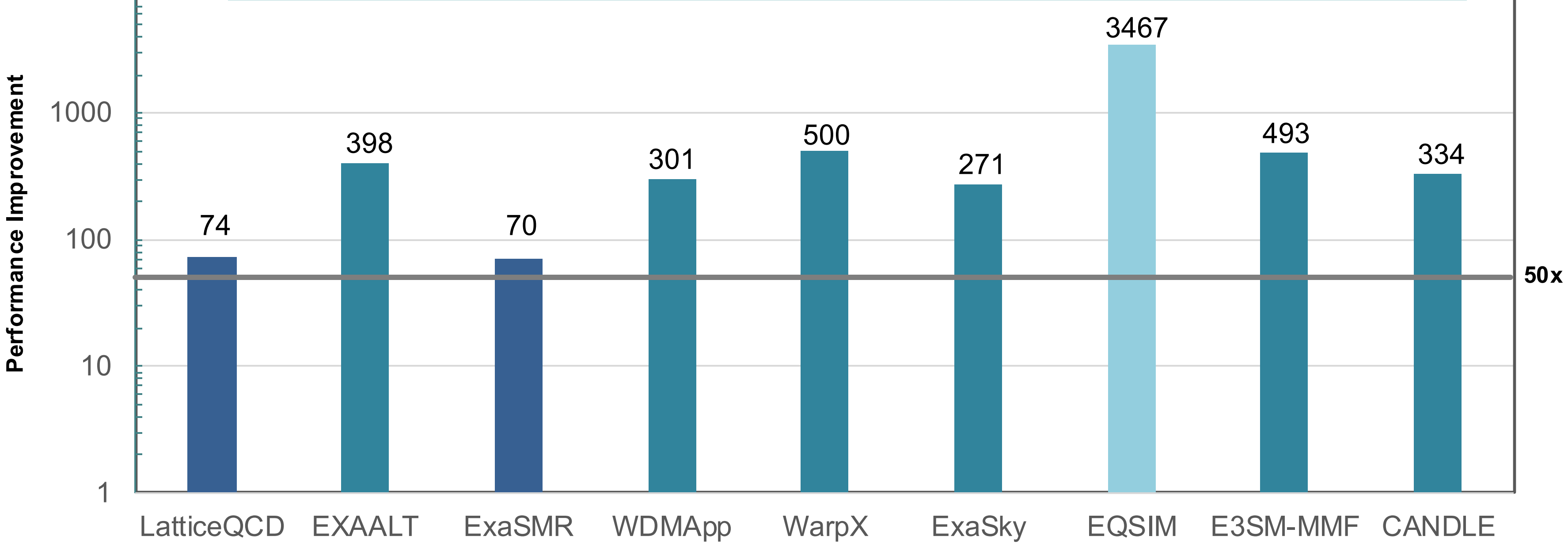
Wind Energy (ExaWind)


Additive Manufacturing (ExaAM)

2024: ECP Application Results Exceeded Expectations

Additional 1.5 –70x improvements due to improved algorithms

9 out of 11 KPP-1 projects surpassed ambitious 50x performance target

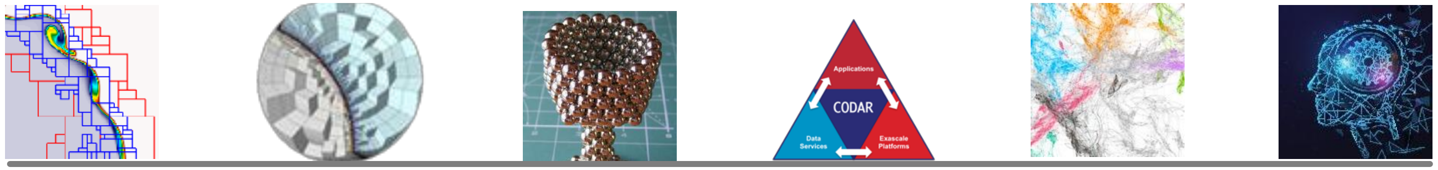


7 out of 10 SC KPP-2 projects demonstrated exascale capability

3 out of 4 NNSA KPP-2 projects demonstrated exascale capability

Codesign played a critical role

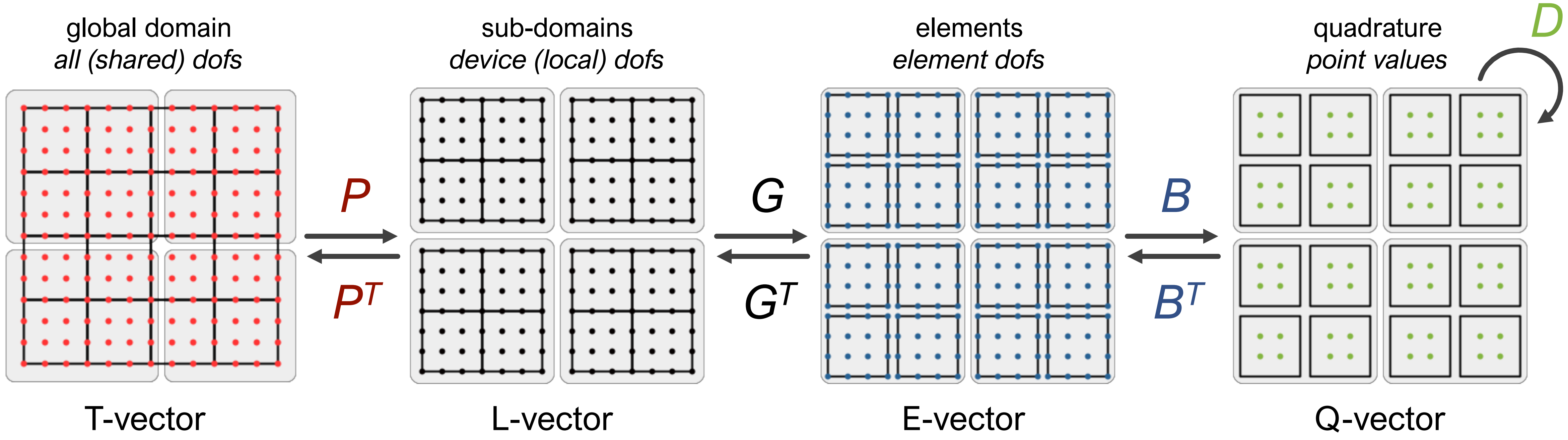
AMReX CEED CoPA CODAR ExaGraph ExaLearn



FEM Operator Decomposition + Partial Assembly for HPC

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

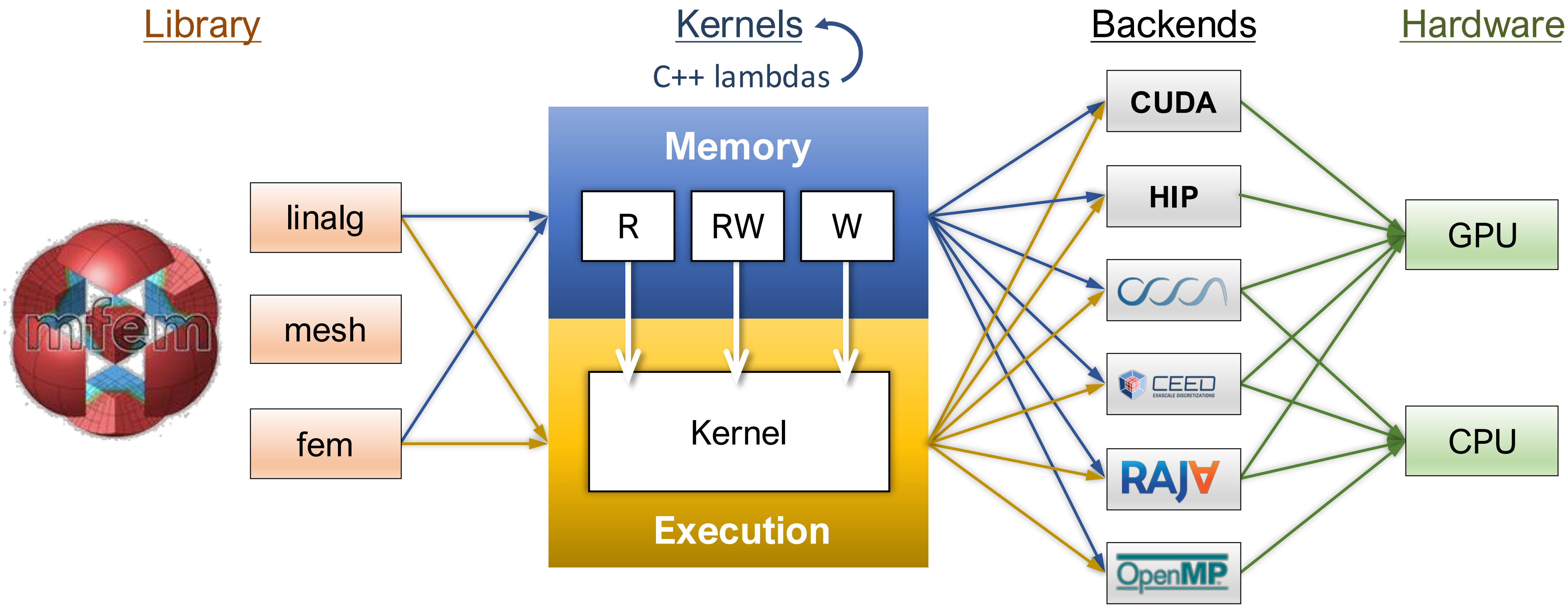
$$A = P^T G^T B^T D B G P$$



- **Partial assembly = store only D, evaluate B** NO ASSEMBLY REQUIRED
- **Optimal memory, near-optimal FLOPs compared to A**
- **Enables AMR, HO, GPUs**
- **AD-friendly**

GPU Support

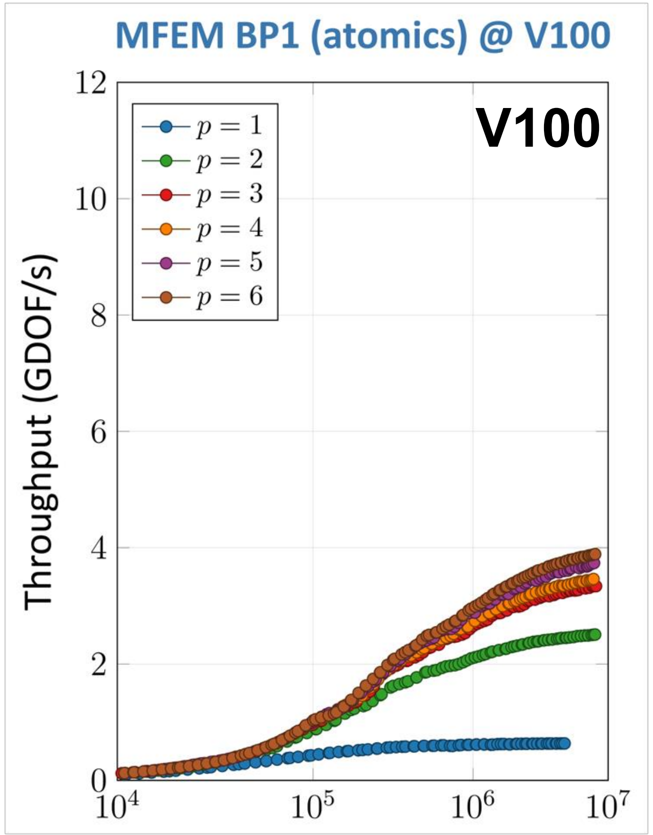
MFEM has provided GPU acceleration for over 5 years (since mfem-4.0)

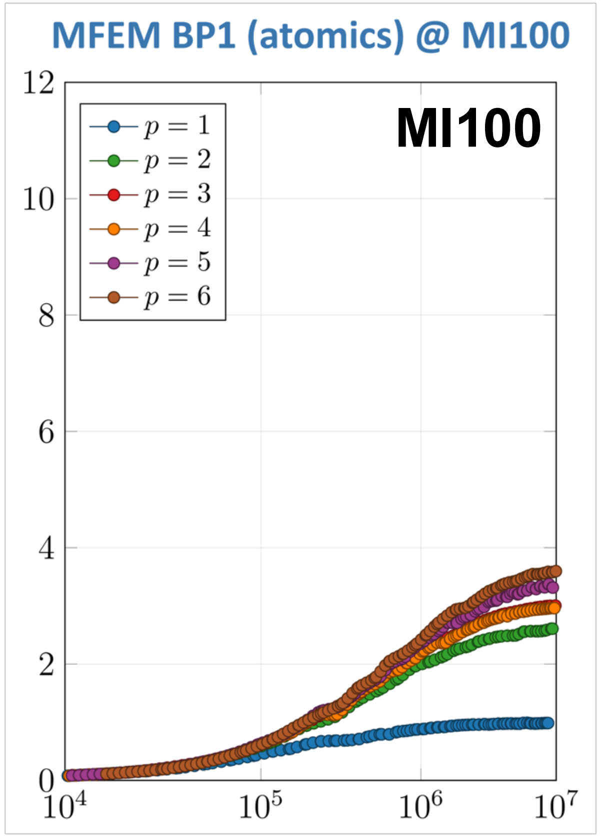


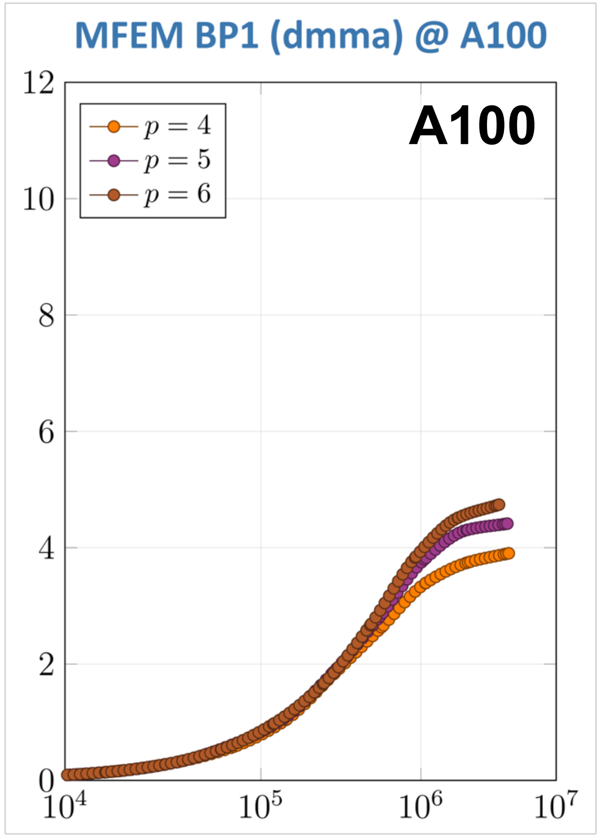
- memory manager
- runtime-selectable backends
- WIP: SYCL support

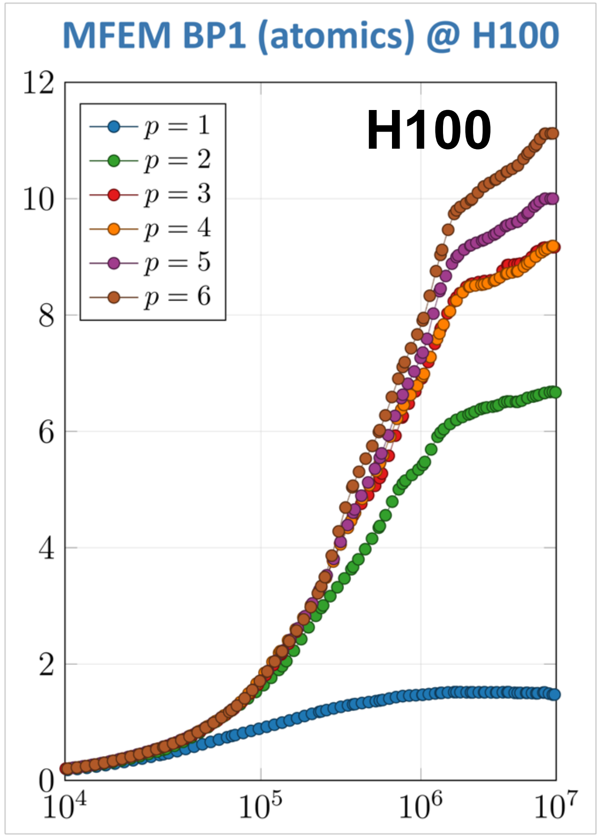
Performance-Portable GPU Finite Element Kernels

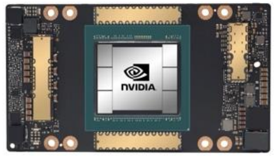
MFEM results on the CEED bake-off problems







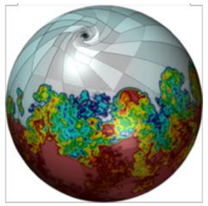


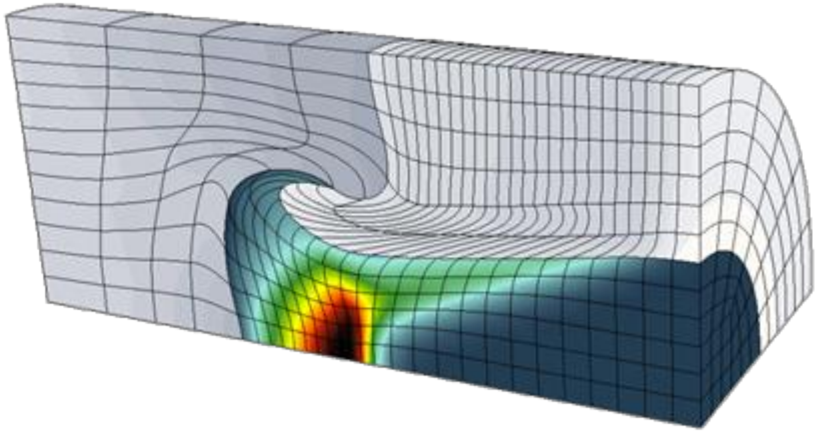


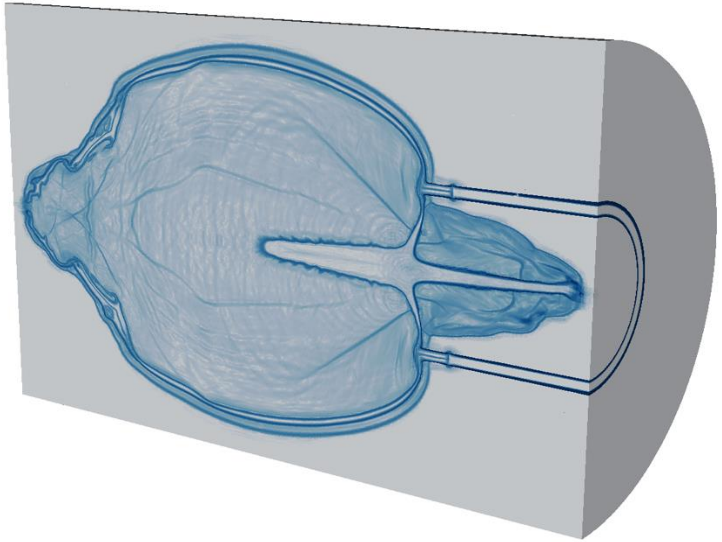
Scalability of high-performance PDE solvers, IJHPCA 2020

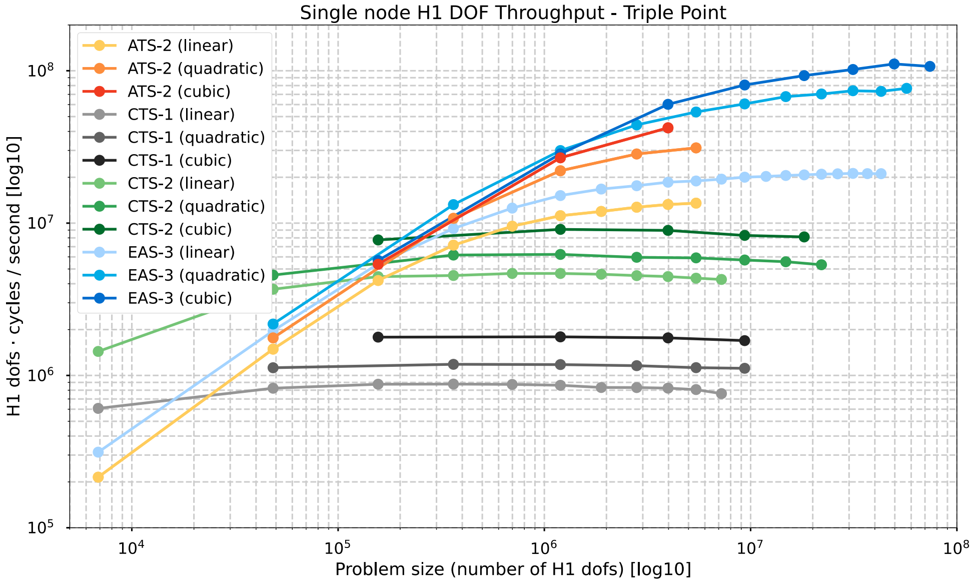
Partially Assembled Methods Perform Better on GPUs

High-order elements yield higher throughput in BLAST









3D throughput / 1 node
CPU-based systems vs
NVIDIA V100 (**ATS-2**)

AMD MI250X (**EAS-3**)

PA CPU/GPU

500 cycles, ALE period = 50

Phase	FA CPU	PA CPU	PA 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)

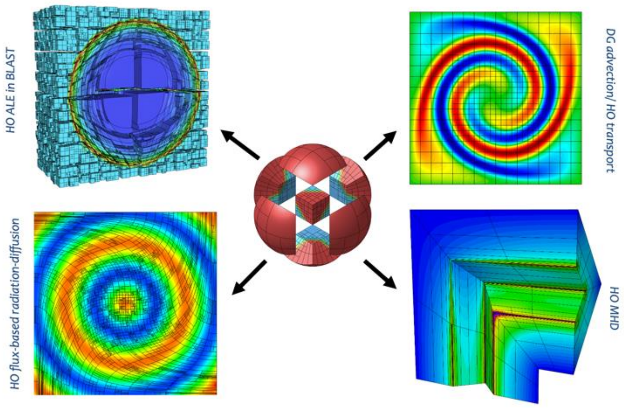
***Matrix-free approaches for GPU acceleration of a high-order finite element hydrodynamics application using MFEM, Umpire, and RAJA**, IJHPCA 2022*

Adaptive Mesh Refinement on Unstructured Grids

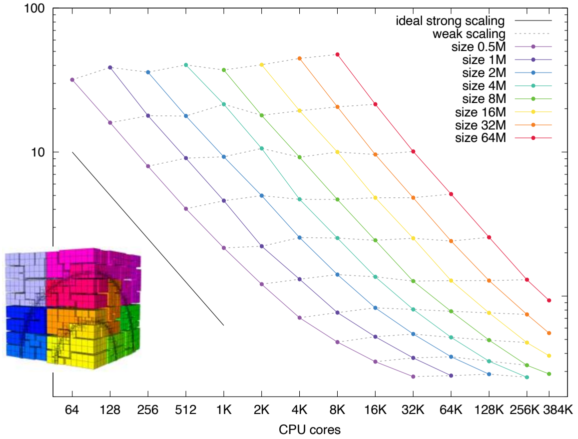
Same AMR algorithms applied to a variety of high-order physics

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

- **AMR on library level**
  - Conforming local refinement on simplex meshes
  - Non-conforming refinement for *all element types*
- **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 (serial)
  - Derefinement
  - Serial and parallel, including parallel load balancing
  - Independent of the physics
  - Easy to incorporate in applications
- **Coming soon**
  - hp-FEM · general anisotropic · SubMesh + AMR

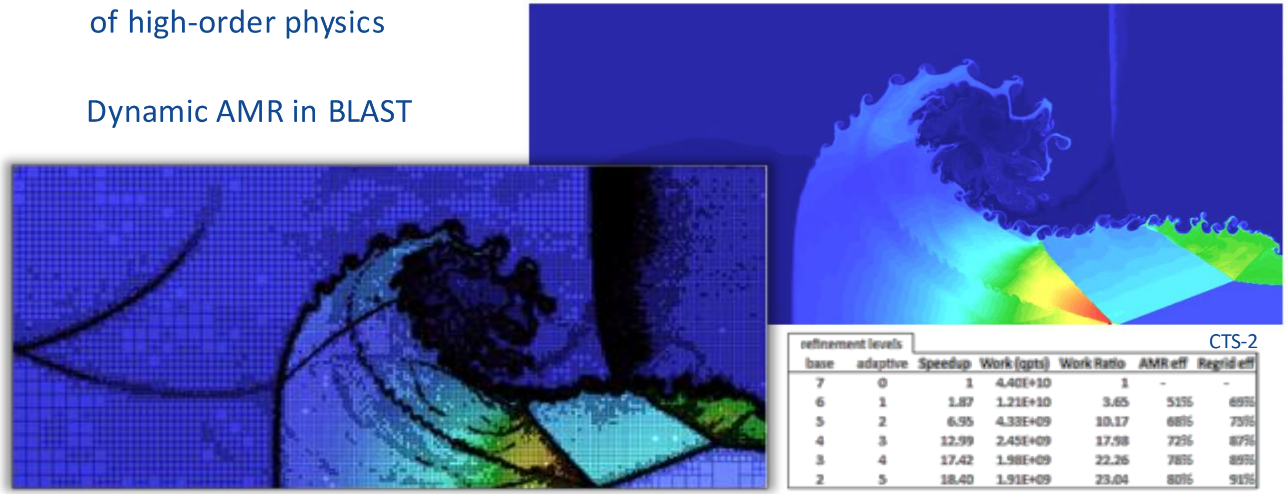


Same AMR algorithms can be applied to a wide variety of high-order physics



Scalability to 400K MPI tasks

Dynamic AMR in BLAST



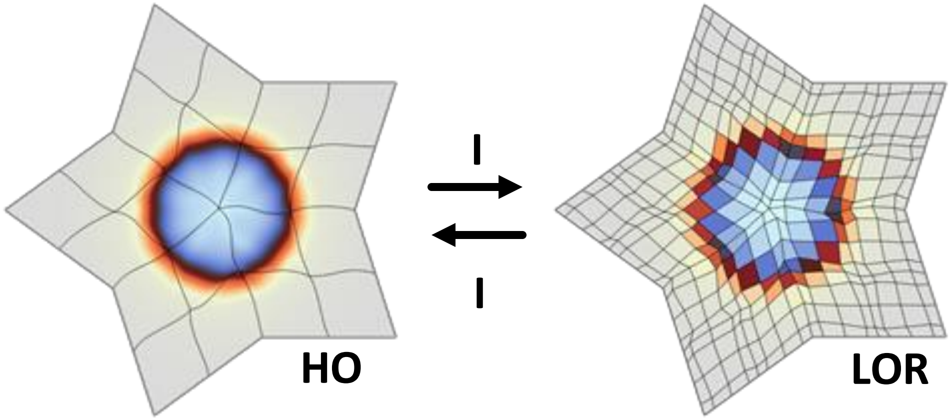
CTS-2

refinement levels						
base	adaptive	Speedup	Work (qpts)	Work Ratio	AMR eff	Regrid eff
7	0	1	4.40E+10	1	-	-
6	1	1.87	1.21E+10	3.65	51%	69%
5	2	6.95	4.33E+09	10.17	68%	73%
4	3	12.99	2.45E+09	17.98	72%	87%
3	4	17.42	1.98E+09	22.26	78%	89%
2	5	18.40	1.91E+09	23.04	80%	91%

Low-Order-Refined (LOR) Solvers

Spectrally equivalent low-order operator on a refined grid

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

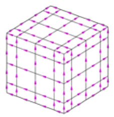


- Pick LOR space and HO basis so **P=R=I** (Gerritsma, Dohrmann)
- $\mathbf{A}_{\mathbf{LOR}}$ is sparse and spectrally equivalent to $\mathbf{A}_{\mathbf{HO}}$

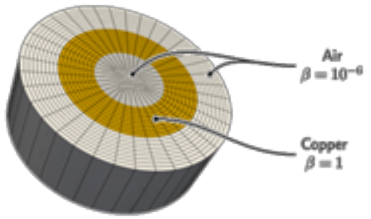
Theorem 2. *Let $M_\star$ and $K_\star$ denote the mass and stiffness matrices, respectively, where $\star$ represents one of the above-defined finite element spaces with basis as in Section 4.3. Then we have the following spectral equivalences, independent of mesh size h and polynomial degree p.*

$$\begin{aligned} M_{V_h} &\sim M_{V_p}, & K_{V_h} &\sim K_{V_p}, \\ M_{\boldsymbol{W}_h} &\sim M_{\boldsymbol{W}_p}, & K_{\boldsymbol{W}_h} &\sim K_{\boldsymbol{W}_p}, \\ M_{\boldsymbol{X}_h} &\sim M_{\boldsymbol{X}_p}, & K_{\boldsymbol{X}_h} &\sim K_{\boldsymbol{X}_p}, \\ M_{Y_h} &\sim M_{Y_{p-1}}, & & \\ M_{Z_h} &\sim M_{Z_p}, & K_{Z_h} &\sim K_{Z_p}. \end{aligned}$$

- $(\mathbf{A}_{\mathbf{HO}})^{-1} \approx (\mathbf{A}_{\mathbf{LOR}})^{-1} \approx \mathbf{B}_{\mathbf{LOR}}$ - can use AMG, AMS, ADS



$\nabla \times \nabla \times \boldsymbol{u} + \beta \boldsymbol{u} = \boldsymbol{f}$

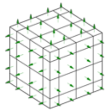


LOR–AMS

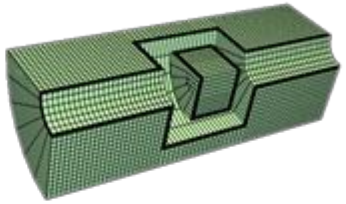
p	Its.	Assembly (s)	AMG Setup (s)	Solve (s)	# DOFs	# NNZ
2	41	0.082	0.277	0.768	516,820	1.65×10^7
3	63	0.251	0.512	2.754	1,731,408	5.64×10^7
4	75	0.679	1.133	7.304	4,088,888	1.34×10^8
5	62	1.574	2.185	11.783	7,968,340	2.61×10^8
6	89	3.336	4.024	30.702	13,748,844	4.51×10^8

Matrix-Based AMS

p	Its.	Assembly (s)	AMG Setup (s)	Solve (s)	# DOFs	# NNZ
2	39	0.140	0.385	1.423	516,820	5.24×10^7
3	44	1.368	1.572	9.723	1,731,408	4.01×10^8
4	49	9.668	5.824	45.277	4,088,888	1.80×10^9
5	53	61.726	15.695	148.757	7,968,340	5.92×10^9
6	56	502.607	40.128	424.100	13,748,844	1.59×10^{10}



$\nabla(\alpha \nabla \cdot \boldsymbol{u}) - \beta \boldsymbol{u} = \boldsymbol{f}$



	LOR–ADS		Matrix-Based ADS		
p	Runtime (s)	Memory (GB)	Runtime (s)	Memory (GB)	Speedup
2	2.11	0.04	2.98	0.20	1.41×
3	6.64	0.15	22.58	1.84	3.40×
4	17.40	0.35	114.35	9.13	6.57×
5	43.70	0.68	422.74	32.21	9.67×
6	92.76	1.18	1324.94	91.09	14.28×

GPU Performance of LOR Solvers

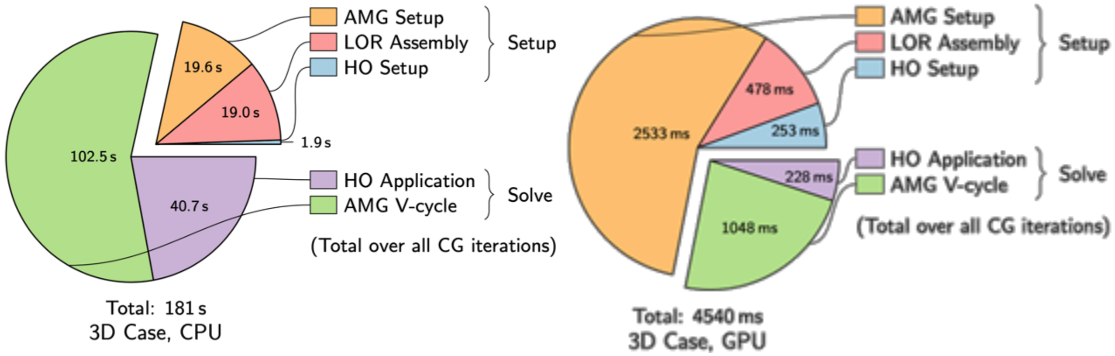
Efficient matrix-free solvers for PA operators

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

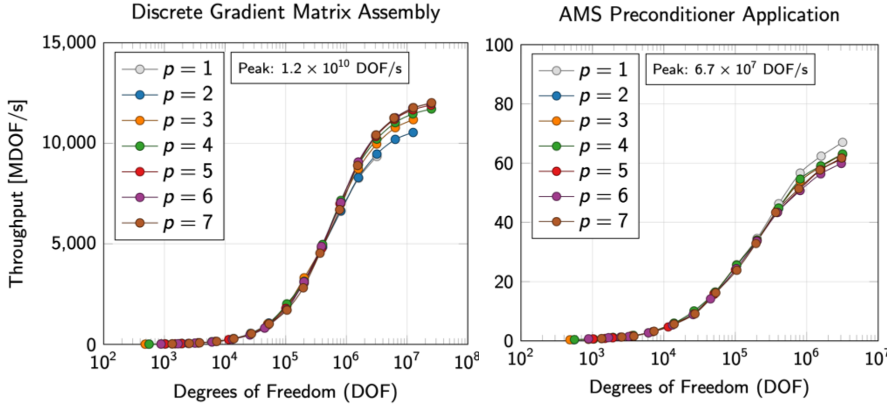
- Using LOR + *hypre*'s AMG, AMS and ADS solvers in MFEM on the GPU is **one line of code**
  —MFEM is the FE interface to *hypre* for many apps

```
// For example:
// if 'a' is H1 diffusion...
LORSolver<HypreBoomerAMG> lor_amg(a, ess_dofs);
// if 'a' is ND curl-curl...
LORSolver<HypreAMS> lor_ams(a, ess_dofs);
// if 'a' is RT div-div...
LORSolver<HypreADS> lor_ads(a, ess_dofs);
```

- We have performed **end-to-end GPU acceleration** of the entire solution algorithm
  —Assembly, preconditioner setup, solve phase
  —Details and performance metrics in *End-to-end GPU acceleration of low-order-refined preconditioning for high-order finite element discretizations, IJHPCA, submitted*



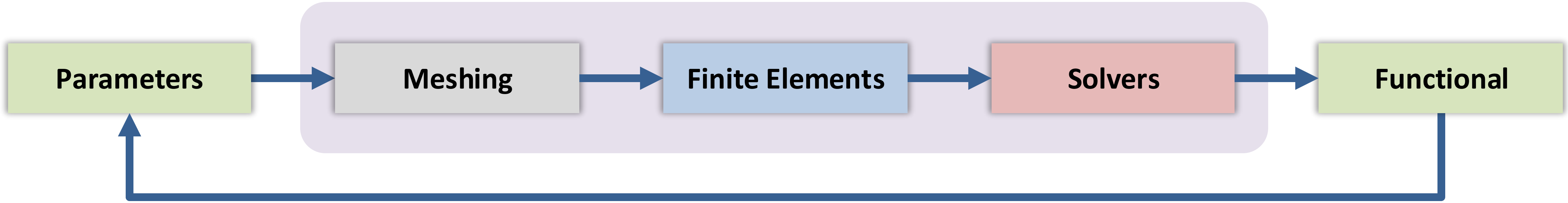
- **Flexibility:** solvers perform well
  —For H^1, H(curl), H(div)
  —With high-order elements
  —On AMR meshes, etc.
- Excellent **strong** and **weak scalability:**
  —Benchmarked up to 1024 GPUs, 1.1 billion DOFs



∂FEM: Autodiff for Partially Assembled Operators

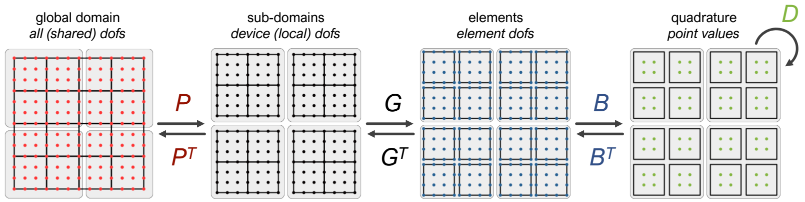
Jacobians and derivatives of FEM operators in a user-friendly way

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



- FEM decomposition

$$\boxed{A = P^T G^T B^T D B G P}$$



- Parameters $\hat{\rho} = B_\rho G_\rho P_\rho \, \rho$
- Parametric nonlinear operator

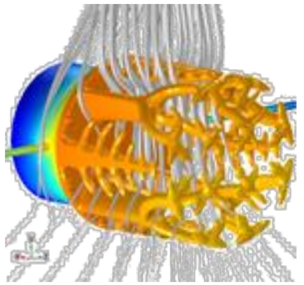
$$\boxed{A(u;\rho) = P^T G^T B^T \, D(\hat{u},\hat{\rho})}$$

- Need to differentiate at Q-points only!

$$\nabla_u A(u;\rho) = P^T G^T B^T \, \nabla_{\hat{u}} D(\hat{u},\hat{\rho}) \, B \, G \, P$$

(Jacobian is FEM decomposed linear operator)

- Differentiate the Q-function *D* with Enzyme!
  - AD at LLVM level, *after* compiler optimization
  - Can mix code from different languages
  - Differentiate across function calls (e.g. EOS)
  - AD with minimal code changes
  - Differentiate only what is necessary



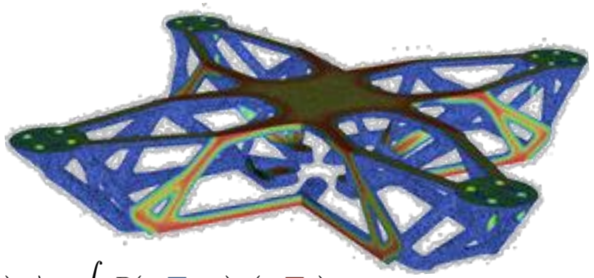
Topology-optimized LED heat sink

MFEM + Enzyme

Roadmap for Next Year

Plans for FY25

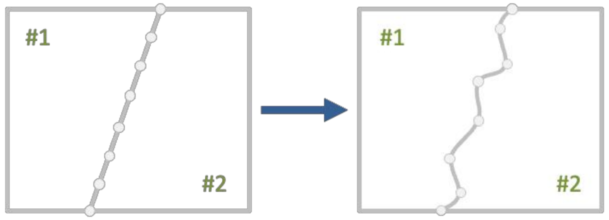
- **El Capitan**
  - Large-scale performance on MI300
  - Application porting and optimizations
- **Differentiable Simulations**
  - dFEM AD in next release
  - AD on GPU · Enzyme collaboration
  - Design optimization · ALE multi-physics
- **R&D**
  - Compressible and incompressible flow · Fusion: both magnetic and ICF
  - AMR improvements · pyramids · high-order simplices · matrix-free solvers
  - Robust meshing, discretizations and solvers for automated workflows
- **New releases**
  - mfem-4.8 in Mar · switch to C++17
- **What would you like to see?**
  - Slack: #meet-the-team · GitHub: github.com/mfem/mfem/issues · Email: mfem@llnl.gov

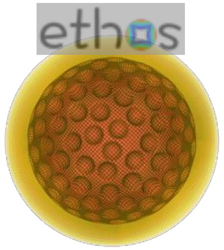


$$\langle F_D(u,\rho), v\rangle = \int_\Omega D(u,\nabla u,\rho)\cdot(v,\nabla v)$$

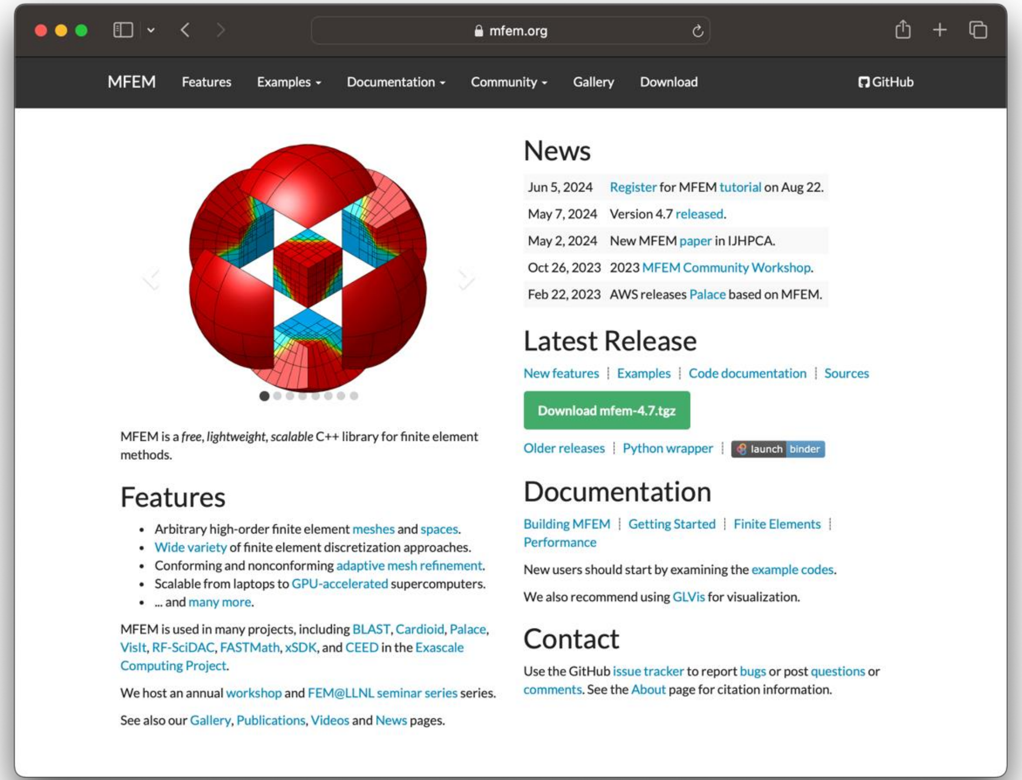
$$\updownarrow$$

$$F_D(u,\rho) = T_v^{\mathrm{T}}\, D_\omega(T_u u, T_\rho \rho) \longrightarrow \frac{dF}{du}(u^*,\rho^*) = T_v^{\mathrm{T}}\, \partial_{\tilde u} D_\omega(\tilde u^*, \tilde\rho^*) T_u$$





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!

Julian
Andrej
@jandrej

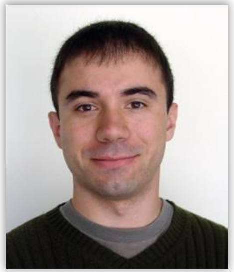
John
Camier
@camierjs

Dylan
Copeland
@dylan-copeland

Veselin
Dobrev
@v-dobrev

Aaron
Fisher
@acfisher

Andrew
Ho
@helloworld922



Tzanio
Kolev
@tzanio

Justin
Laughlin
@justinlaughlin

Boyan
Lazarov
@bslazarov

Ketan
Mittal
@kmittal2

Jan
Nikl
@najlkin

Will
Pazner
@pazner

Socratis
Petrides
@psocratis

Sohail
Reddy
@sohailreddy

Mathias
Schmidt
@SchmidtMat

Mark
Stowell
@mlstowell

Vladimir
Tomov
@vladotomov

Chris
Vogl
@cjvogl

mfem.org

Lawrence Livermore National Laboratory