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, ...
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 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, 356K
- **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

MFEM has been downloaded from 115 countries

Top contributors as of Oct 2023

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

- 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

- **Volta**
  \[- \nabla \cdot \mathbf{e} \nabla \phi = \rho - \nabla \cdot \mathbf{\bar{P}}\]

- **Tesla**
  \[\nabla \times \mu^{-1} \nabla \times \mathbf{A} = \mathbf{\bar{J}} + \nabla \times \mu^{-1} \mu_0 \mathbf{\bar{M}}\]

- **Maxwell** · transient full-wave EM
  \[\frac{\partial (\mathbf{E})}{\partial t} = \nabla \times (\mu^{-1} \mathbf{B}) - \sigma \mathbf{E} - \mathbf{\bar{J}}\]
  \[\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}\]

- **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

mfem.org/fluids

3D Taylor-Green vortex, 7th order
Double shear layer, 5th order, Re = 100000
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 (Cardio, 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, 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 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

- **MFEM BP1 (atomics) @ V100**
  - Throughput (GDOF/s) vs. time (ns)
  - Configurations: $p = 1, 2, 3, 4, 5, 6$

- **MFEM BP1 (atomics) @ MI100**
  - Throughput (GDOF/s) vs. time (ns)
  - Configurations: $p = 1, 2, 3, 4, 5, 6$

- **MFEM BP1 (dmma) @ A100**
  - Throughput (GDOF/s) vs. time (ns)
  - Configurations: $p = 4, 5, 6$

- **MFEM BP1 (atomics) @ H100**
  - Throughput (GDOF/s) vs. time (ns)
  - Configurations: $p = 1, 2, 3, 4, 5, 6$
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

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

Website: mfem.org
Software: github.com/mfem
Publications: mfem.org/publications
Email: mfem@llnl.gov

MFEM is a free, lightweight, scalable C++ library for finite element methods.

Features
- 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 many more.

MFEM is used in many projects, including BLAST, Cardioid, VisIt, RF/SiDAC, FASTMath, uC3D, and CEED in the Exascale Computing Project. See also our Gallery, Publications, and News pages.

Latest Release
- New features
- Examples
- Code documentation
- Sources

Download mfem-4.3.tgz

Older releases
- Python wrapper

Documentation
- Building MFEM
- Getting Started
- Finite Elements
- Performance
- New users should start by examining the example codes.
- We also recommend using GID/Y for visualization.

Contact
- Use the Github issue tracker to report bugs or post questions or comments.
- See the About page for citation information.

News
- Jul 30, 2021: MFEM Community Workshop in October.
- Apr 22, 2021: MFEM featured on SIAM magazine cover.

Explore our publications

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