SoftwareX 26 (2024) 101690

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

3.0 - MOOSE: Enabling massively parallel multiphysics simulations☆

Guillaume Giudicelli [a,*], Alexander Lindsay [a], Logan Harbour [a], Casey Icenhour [a], Mengnan Li [a], Joshua E. Hansel [a], Peter German [a], Patrick Behne [a], Oana Marin [a], Roy H. Stogner [a], Jason M. Miller [a], Daniel Schwen [a], Yaqi Wang [a], Lynn Munday [a], Sebastian Schunert [a], Benjamin W. Spencer [a], Dewen Yushu [a], Antonio Recuero [a], Zachary M. Prince [a], Max Nezdyur [a], Tianchen Hu [b], Yinbin Miao [b], Yeon Sang Jung [b], Christopher Matthews [c], April Novak [e], Brandon Langley [f], Timothy Truster [g], Nuno Nobre [h], Brian Alger [a], David Andrs [a], Fande Kong [a], Robert Carlsen [a], Andrew E. Slaughter [a], John W. Peterson [d], Derek Gaston [a], Cody Permann [a]

[a] Idaho National Laboratory, Idaho Falls, ID 83415, United States
[b] Argonne National Laboratory, Lemont, IL 60439, United States
[c] Los Alamos National Laboratory, Los Alamos, NM 87545, United States
[d] Akselos Inc., 2101 West Blvd., Houston, TX 77042, United States
[e] Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, United States
[f] Oak Ridge National Laboratory, Oak Ridge, TN 37830, United States
[g] Department of Civil and Environmental Engineering, University of Tennessee, Knoxville, TN 37996, United States
[h] STFC Hartree Centre, Daresbury Laboratory, Daresbury WA4 4AD, United Kingdom

ARTICLE INFO

Keywords:
Framework
Finite-element
Finite-volume
Parallel
Multiphysics
Multiscale

ABSTRACT

The development of MOOSE has kept accelerating since the last release, with over 2,100 pull requests merged over the last 30 months that involved nearly fifty contributors across close to a dozen institutions internationally. The growth in MOOSE's capabilities and downstream applications is reflected in the growth of the community. User support provided on the GitHub discussions forum has steadily increased to nearly 50 daily interactions. New simulation projects, notably to model advanced nuclear reactor and fusion devices, are driving a significant expansion of the capabilities. This paper reports on these developments, with several major released features, new physics modules, and key improvements to the user experience and simulation workflow.

Metadata

Nr	Code metadata description	*Please fill in this column*
C1	Current code version	*v3*
C2	Permanent link to code/repository used for this code version	https://github.com/idaholab/moose
C3	Permanent link to reproducible capsule	https://github.com/idaholab/moose/tree/2024–02–12-release
C4	Legal code license	*GNU LGPL*
C5	Code versioning system used	*git*
C6	Software code languages, tools and services used	*C++, MPI, OpenMP, Python*
C7	Compilation requirements, operating environments and dependencies	*Requirements:* GCC/Clang C++17 compliant compiler; 16 GB memory (debug builds); 64-bit x86 + Apple Silicon support; 30 GB disk space *Operating environments:* Linux, macOS > 10.12 Dependencies: PETSc, libMesh
C8	If available, link to developer documentation/manual	https://mooseframework.org
C9	Support email for questions	https://github.com/idaholab/moose/discussions

DOI of original article: https://doi.org/10.1016/j.softx.2022.101202.
☆ **Refers to:** https://doi.org/10.1016/j.softx.2020.100430
* Corresponding author.
E-mail address: guillaume.giudicelli@inl.gov (G. Giudicelli).

https://doi.org/10.1016/j.softx.2024.101690
Received 26 February 2024; Received in revised form 10 March 2024; Accepted 12 March 2024
Available online 22 March 2024
2352-7110/

1. Description of the software-update

1.1. User-oriented changes

The development of new capabilities in MOOSE [1] is driven by user and developer needs. A long-term demand was to render MOOSE able to restart between different simulation types. For example, the first simulation may be an eigenvalue computation of a steady-state dominant eigenmode, and the second simulation, restarting from the first, may be a transient simulation of a perturbation of the initial steady state. This process is widely used in advanced reactor analysis to simulate accidental transients [2].

We added a new capability to mark a solution as "invalid." This allows developers to guard against a solution being "out of bounds," for example in a material property correlation, and inform the user without stopping the simulation, or overwhelming the console if the out-of-bounds situation repeats itself over a large portion of the domain. A solution is allowed to be invalid during a nonlinear solve - but by default not once the solve has converged.

Efforts have continued to expand the meshing capabilities of MOOSE. In addition to the developments in the Reactor module [3] targeted toward nuclear applications, several application-agnostic capabilities were added. MOOSE is now able to handle general axisymmetric coordinates, as well as leverage poly2tri [4] through libMesh [5] to generate 2D triangular meshes inside arbitrarily shaped curves. MOOSE integrated diagnostics to report on numerous issues or unsupported features in external meshes, such as elements or sides with locally negative Jacobians or non-conformalities, and to correct several of these issues automatically.

MOOSE now supports p-refinement, e.g., refining or coarsening the order of the finite element basis functions. Prior to using p-refinement, libMesh can be instructed to add mid-edge and mid-face nodes to a mesh; these are then tied to the degrees of freedom shared around edges or faces. Only one refinement type, i.e. h- or p-refinement, may be used in a single simulation.

MOOSE now supports additional elements for higher order finite element analysis, notably the 21-node prism and the 14-node tetrahedron. MOOSE also supports spline-based isogeometric analysis (IGA) simulations. IGA meshes can be read from files in the Exodus format. Simulations on these meshes can compute solutions in either the spline basis or a derived Rational-Bernstein-Bezier basis.

A large effort has been placed on increasing the flexibility of MOOSE simulations. The new WASP parser [6] was deployed. It matches the features of the previous HIT parser, adds input-file-include capabilities and powers the new built-in MOOSE Language Server enabling input file auto-completion and diagnostics for many text editors. WASP is also used in MooseServer, a class that can process controllable parameters live during a simulation. The handling of time steps has been reworked, with a new time stepper composition system. Several time steppers can now suggest the time step to analyze various regimes of a transient simulation with different time discretizations. The new "Times" system can store and modify a vector of times, to be able to select the next time step dynamically in a simulation.

1.2. Application coupling changes

MOOSE applications can be coupled with great ease, and fields are passed between simulations using transfers. This technique has been refined with the introduction of a new base class for transfers called the general field transfer. The new derived transfer classes support a wide variety of features, including the ability to: transfer between two child applications, transfer array variables and higher order variables, automatically detect floating-point indetermination during transfers, handle coordinate system transformations between applications, and restrict transfers to block boundary and mesh divisions. The new "MeshDivision" system lets users match regions of the mesh that are indexed in the mesh division to child applications with the same index.

The new "Positions" system enables the creation of distributed applications at various programmatically set locations, such as at element centroids or at positions loaded from one or more files. The new system allows application developers to create new ways of distributing calculations spatially, which is a common technique for multiscale calculations.

MOOSE is now able to create multiple nonlinear systems. On the same mesh, or on parts of the same mesh, two different equations can now be solved independently from the same input file. The coupling can then be iterated to convergence. This avoids duplicating fields, and the memory cost of doing so, in two applications.

The dynamic loading capabilities of MOOSE have been greatly improved. This enables users without access to the source code of a closed source application to leverage external objects from dynamic libraries in their simulations without recompiling to include the application they do not have source access to.

1.3. Application developer-oriented changes

Several improvements have been made to facilitate the development of MOOSE-based applications. One notable change to the development process of these applications is the addition of several debug logs for application developers to better understand what actions are performed by MOOSE behind the scenes. MOOSE-based simulations can now be set to output the objects created during the problem setup and every object executed during execution loops, for example during the nonlinear and linear iterations.

This release saw the deployment of functors. A Functor is a base class for variables, functions, functor material properties and postprocessor classes, that enables the on-the-fly computation of field quantities at arbitrary locations such as elements, nodes or element faces and for arbitrary states such as current or previous time-step or solver iteration. This generality was first employed for face evaluations of advected quantities in finite volume fluid flow simulations, and Functors now enable finite volume computations on displaced meshes and finite element - finite volume coupled calculations [7].

1.4. Performance improvements

MOOSE is now able to repartition problems using stateful material properties. Stateful material properties are used notably in solid mechanics to represent quantities with a history effect, such as strain. Repartitioning is challenging because previous states of properties have to be communicated between partitions, but it makes adaptive mesh refinement more efficient in parallel.

MOOSE can now compute the contribution of kernels and boundary conditions to the residual and Jacobian in a nonlinear iteration of a Newton solve from a single loop. Automatic differentiation [8] provides the Jacobian as the residual is computed.

1.5. MOOSE modules development

The MOOSE repository hosts modules which serve as building blocks to many downstream applications. New modules were created for electromagnetics [9], transport of scalar quantities, optimization, and solid properties. The heat conduction module has been renamed as the heat transfer module to better describe its capabilities. The tensor mechanics module has similarly been renamed to solid mechanics. The optimization module provides interfaces to the PETSc [10] and TAO [11] libraries. It can be used for inverse optimization, usually for finding the simulation input parameters that best fit experimental results.

CRediT authorship contribution statement

Guillaume Giudicelli: Writing – review & editing, Writing – original draft, Software. **Alexander Lindsay:** Writing – review & editing, Writing – original draft, Software, Project administration. **Logan Harbour:** Software. **Casey Icenhour:** Writing – review & editing, Software. **Mengnan Li:** Software. **Joshua E. Hansel:** Writing – review & editing, Writing – original draft, Software. **Peter German:** Software. **Patrick Behne:** Software. **Oana Marin:** Writing – review & editing, Software. **Roy H. Stogner:** Software. **Jason M. Miller:** Software. **Daniel Schwen:** Software, Methodology. **Yaqi Wang:** Software. **Lynn Munday:** Software. **Sebastian Schunert:** Software, Project administration. **Benjamin W. Spencer:** Writing – review & editing, Software, Project administration. **Dewen Yushu:** Software, Project administration. **Antonio Recuero:** Software. **Zachary M. Prince:** Software. **Max Nezdyur:** Software. **Tianchen Hu:** Software. **Yinbin Miao:** Software. **Yeon Sang Jung:** Software. **Christopher Matthews:** Software. **April Novak:** Software. **Brandon Langley:** Software. **Timothy Truster:** Software. **Nuno Nobre:** Software. **Brian Alger:** Software. **David Andrs:** Software. **Fande Kong:** Software. **Robert Carlsen:** Software. **Andrew E. Slaughter:** Software. **John W. Peterson:** Software. **Derek Gaston:** Project administration, Funding acquisition. **Cody Permann:** Supervision, Software, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The MOOSE team would like to acknowledge the following contributors to the core of the framework: Lise Charlot, Ruijie Liu, Nathaniel Peat, Chandrakanth Bolisetti, Stephen Novascone, Jacob Badger, Kyle Gamble, Larry Aagesen, Adam X. Zabriskie, Joshua Hanophy, Stephanie Pitts, Daniel VanWasshenova, Andrea Rovinelli, Swetha Veeraraghavan, Mark Messner, Ling Zou, Rob Lefebvre, Michael Tonks, John Mangeri, Andrea Jokisaari, Paolo Balestra, Som L. Dhulipala, Jason Hales, Mauricio Tano, Jan Vermaak, Patrick Albert, Daniel Levario, Trent Savage, Benjamin Spaude, Joseph Silberman, Jake Bowhay, Shikhar Kumar, Jesse Carter, Chaitanya Bhave, Karthikeyan Chockalingam, Bo Zeng, Parikshit Bajpai, Nathan Miller, Sourabh Kadambi, Sudipta Biswas, Christopher Green, Wen Jiang, Andy Wilkins.

This research was partially funded by the Office of Nuclear Energy of the U.S. Department of Energy NEAMS project, under Contract No. DE-NE0008983. This research work was partially prepared for DOE through Idaho National Laboratory (INL)'s LDRD Program under the DOE Idaho Operations Office. This research made use of the resources of the High-Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy under Contract No. DE-AC07-05ID14517.

References

[1] Lindsay Alexander, Gaston Derek, Permann Cody, et al. 2.0 - MOOSE: enabling massively parallel multiphysics simulations. SoftwareX 2022;20:101202. https://doi.org/10.1016/j.softx.2022.101202.
[2] Matthews Christopher, Laboure Vincent, DeHart Mark, Hansel Joshua, Andrs David, Wang Yaqi, Ortensi Javier, Martineau Richard C. Coupled multiphysics simulations of heat pipe microreactors using DireWolf. Nucl Technol 2021;207(7):1142–62.
[3] Shemon, et al. MOOSE reactor module: an open-source capability for meshing nuclear reactor geometries. Nucl Sci Eng 2023:1–25. https://doi.org/10.1080/00295639.2022.2149231.
[4] Hasse, J.N. (2021). poly2tri. Retrieved from GitHub: https://github.com/jhasse/poly2tri.
[5] Kirk Benjamin S, Peterson John W, Stogner Roy H, Carey Graham F. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. Eng Comput 2006;22(3–4):237–54. https://doi.org/10.1007/s00366-006-0049-3.
[6] Lefebvre Robert A, Langley Brandon R, Lefebvre Jordan P, USDOE. Workbench analysis sequence processor. Comput Softw 2017. https://doi.org/10.11578/dc.20201125.2. November 16, https://www.osti.gov/servlets/purl/1419488.
[7] Biswas Sudipta, et al. INL/RPT-23-74941. Idaho National Laboratory; 2023.
[8] Lindsay Alexander, Stogner Roy, Gaston Derek, Schwen Daniel, Matthews Christopher, Jiang Wen, et al. Automatic differentiation in MetaPhysicL and its applications in MOOSE. Nucl Technol 2021;207:905–22. https://doi.org/10.1080/00295450.2020.1838877.
[9] Icenhour CT, Lindsay Alexander D, Permann Cody J, Martineau Richard C, Green David L, Shannon Steven C. The MOOSE electromagnetics module. SoftwareX 2024;25:101621. https://doi.org/10.1016/j.softx.2023.101621.
[10] Balay, S., Shrirang Abhyanker, Mark F. Adams, et al. 2023. PETSc Web page at https://petsc.org/.
[11] Balay Satish, et al. ANL-21/39 - Revision 3.20. Argonne National Laboratory; 2023. https://doi.org/10.2172/2205494.