The Journal of Open Source Software

Adamantine 1.0: A Thermomechanical Simulator for Additive Manufacturing

Bruno Turcksin [1¶] **and Stephen DeWitt** [1]

1 Oak Ridge National Laboratory, Oak Ridge, TN, USA ¶ Corresponding author

DOI: 10.21105/joss.07017

Software

- Review
- Repository
- Archive

Editor: Lucy Whalley

Reviewers:

- @mohd-afeef-badri
- @ericneiva
- @FelixFroelich
- @lukashof

Submitted: 20 June 2024
Published: 10 October 2024

Summary

`Adamantine` is a thermomechanical simulation code that is written in C++ and built on top of deal.II (Arndt et al., 2023), p4est(Burstedde et al., 2011), ArborX (Lebrun-Grandié et al., 2020), Trilinos (Team, 2020 (accessed May 22, 2020)), and Kokkos (Trott et al., 2022). `Adamantine` was developed with additive manufacturing in mind and it is particularly well adapted to simulate fused filament fabrication, directed energy deposition, and powder bed fusion. `Adamantine` employs the finite element method with adaptive mesh refinement to solve a nonlinear anisotropic heat equation, enabling support for various additive manufacturing processes. It can also perform elastoplastic and thermoelastoplastic simulations. It can handle materials in three distinct phases (solid, liquid, and powder) to accurately reflect the physical state during different stages of the manufacturing process. To enhance simulation accuracy, `adamantine` incorporates data assimilation techniques (Asch et al., 2016). This allows it to integrate experimental data from sensors like thermocouples and infrared (IR) cameras. This combined approach helps account for errors arising from input parameters, material properties, models, and numerical calculations, leading to more realistic simulations that reflect what occurs in a particular print.

Statement of Need

Manufacturing "born-qualified" components, i.e., parts ready for critical applications straight from the printer, requires a new approach to additive manufacturing (AM). This vision demands not only precise simulations for planning the build but also real-time adjustments throughout the process to obtain the desired thermomechanical evolution of the part. Currently, setting AM process parameters is an expert-driven, often trial-and-error process. Material changes and geometry complexities can lead to unpredictable adjustments in parameters, making a purely empirical approach slow and expensive. We can overcome this by using advanced simulations for both planning and adaptive control.

`Adamantine`, a thermomechanical simulation tool, offers a solution to process parameter planning and adjustment in AM. During the planning phase, its capabilities can be leveraged to predict the thermomechanical state and optimize process parameters for the desired outcome. For adaptive control, `adamantine` utilizes data from IR cameras and thermocouples. This data is integrated using the Ensemble Kalman Filter (EnKF) method, allowing the simulation to constantly adapt and reflect the actual build process.

With a continuously refined simulation, `adamantine` can predict the final thermomechanical state of the object with greater accuracy. This simulation-enhanced monitoring capability enables a human operator or an adaptive control algorithm to adjust to the build parameters mid-print, if needed, to ensure that printed parts conform to the necessary tolerances.

While other open-source software like AdditiveFOAM (Coleman et al., 2023) excels at heat and mass transfer simulations in additive manufacturing, and commercial options like Abaqus (Dassault Systèmes Simulia Corp., 2024) and Ansys (ANSYS Inc., 2024) offer comprehensive thermomechanical capabilities, `adamantine` stands out for its unique ability to incorporate real-world data through data assimilation. This feature allows for potentially more accurate simulations, leading to better process optimization and final part quality.

Simulated Physics

Thermal simulation

`Adamantine` solves an anisotropic version of standard continuum heat transfer model used in additive manufacturing simulations (Keller et al., 2017; Megahed et al., 2016). The model includes the change of phases between powder, liquid, and solid and accounts for latent heat release for melting/solidification phase transformations. It assumes the presence of a "mushy" zone, i.e., the liquidus and the solidus are different, as is generally the case for alloys. The heat input by the laser, electron beam, electric-arc, or other process-specific heat source is introduced using a volumetric source term (Goldak et al., 1984; Knapp et al., 2023). Adiabatic, convective, and radiative boundary conditions are implemented, with the option to combine convective and radiative boundary conditions.

Mechanical simulation

`Adamantine` can perform elastoplastic simulations. The plastic model is the linear combination of the isotropic and kinematic hardening described in Borja (2013). This allows us to model both the change in yield stress and the Bauschinger effect.

Thermomechanical simulation

Thermomechanical simulations in `adamantine` are performed with one-way coupling from the temperature evolution to the mechanical evolution. We neglect the effect of deformation on the thermal simulation. An extra term in the mechanical simulation accounts for the eigenstrain associated with by thermal expansion of the material (Fung & Tong, 2001; Megahed et al., 2016).

Data Assimilation

Data assimilation "is the approximation of a true state of some physical system at a given time by combining time-distributed observations with a dynamic model in an optimal way" (Asch et al., 2016). `Adamantine` leverages this technique to enhance the accuracy of simulations during and after prints with in-situ characterization. It also ties the simulation results to the particular events (e.g. resulting for stochastic processes) for a specific print.

We have implemented a data assimilation algorithm called the Ensemble Kalman Filter (Asch et al., 2016). This statistical technique incorporates experimental observations into a simulation to provide the best estimate (in the Bayesian sense) of the state of the system that reflects uncertainties from both data sources. EnKF requires to perform an ensemble of simulations with slightly different input model parameters and/or initial conditions. The EnKF calculation and the coordination of simulations of ensemble members are done from inside `adamantine`.

Algorithmic Choices

Time integration

Adamantine includes several options for time integration methods that it inherits from the deal.II library (Arndt et al., 2023). These are: forward Euler, 3rd order explicit Runge-Kutta, 4th order explicit Runge-Kutta, backward Euler, implicit midpoint, Crank-Nicolson, and singly diagonally implicit Runge-Kutta.

Matrix-free finite element formulation

Adamantine uses a variable-order finite element spatial discretization with a matrix-free approach (Kronbichler & Kormann, 2012). This approach calculates the action of an operator directly, rather than explicitly storing the full (sparse) system matrix. This matrix-free approach significantly reduces computational cost, especially for higher-degree finite elements.

MPI support

While mechanical and thermomechanical simulations are limited to serial execution, thermal and EnKF ensemble simulations can use MPI. Thermal simulations can be performed using an arbitrary number of processors. For EnKF ensemble simulations, the partitioning scheme works as follows:

- If the number of processors (Nproc) is less than or equal to the number of EnKF ensemble members (N), adamantine distributes the simulations evenly across the processors. All processors except the first will handle the same number of simulations. The first processor might take on a larger workload if a perfect split is not possible
- Adamantine can leverage more processors than there are simulations, but only if Nproc is a multiple of N. This ensures that all the simulations are partitioned in the same way.

MPI support for mechanical and thermomechanical simulations are a subject of ongoing work.

GPU support

Adamantine includes partial support for GPU-accelerated calculations through the use of the Kokkos library. The evaluation of the thermal operator can be performed on the GPU. The heat source is computed on the CPU. The mechanical simulation is CPU only. Performing the entire computation on the GPU is the subject of ongoing work.

Mesh

Adamantine uses a purely hexahedral mesh. It has limited internal capabilities to generate meshes. For complex geometries, adamantine can load meshes created by mesh generators. The following formats are supported: unv format from the SALOME mesh generator (SMESH) (*SMESH Documentation*, 2024), UCD, VTK (Schroeder et al., 2006), Abaqus (Dassault Systèmes Simulia Corp., 2024) file format, DB mesh, msh file from Gmsh (Geuzaine & Remacle, 2009), mphtxt format from COMSOL (COMSOL AB, 2024), Tecplot (Tecplot Inc., 2024), assimp (*The Asset-Importer-Lib Documentation*, 2024), and ExodusII (*ExodusII Finite Element Data Model, Version 00*, 2005). The generated mesh should be conformal. During the simulation, adamantine can adaptively refine the mesh near the heat source using the forest of octrees approach (Arndt et al., 2023; Burstedde et al., 2011), where each element in the initial mesh can be refined as an octree.

Additional Information

An in-depth discussion of the governing equations and examples showcasing the capabilities ofadamantine can be found at https://adamantine-sim.github.io/adamantine

Acknowledgments

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan).

This research is sponsored by the INTERSECT Initiative and the SEED Program as part of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

ANSYS Inc. (2024). *ANSYS documentation*. ANSYS Inc. https://www.ansys.com/products/structures/ansys-mechanical

Arndt, D., Bangerth, W., Bergbauer, M., Feder, M., Fehling, M., Heinz, J., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.-P., Turcksin, B., Wells, D., & Zampini, S. (2023). The deal.II library, version 9.5. *Journal of Numerical Mathematics*, *31*(3), 231–246. https://doi.org/10.1515/jnma-2023-0089

Asch, M., Bocquet, M., & Nodet, M. (2016). *Data assimilation*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611974546

Borja, R. I. (2013). *Plasticity: Modeling & computation*. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38547-6

Burstedde, C., Wilcox, L. C., & Ghattas, O. (2011). p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, *33*(3), 1103–1133. https://doi.org/10.1137/100791634

Coleman, J., Kincaid, K., Knapp, G. L., Stump, B., & Plotkowski, A. J. (2023). *AdditiveFOAM: Release 1.0* (Version 1.0.0). Zenodo. https://doi.org/10.5281/zenodo.8034098

COMSOL AB. (2024). *COMSOL multiphysics® v6.2*. https://www.comsol.com

Dassault Systèmes Simulia Corp. (2024). *Abaqus documentation*. Dassault Systèmes. https://www.3ds.com/products-services/simulia/products/abaqus/

ExodusII finite element data model, version 00. (2005). https://www.osti.gov/biblio/1230926

Fung, Y., & Tong, P. (2001). *Classical and computational solid mechanics*. World Scientific.

Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, *79*(11), 1309–1331. https://doi.org/10.1002/nme.2579

Goldak, J., Chakravarti, A., & Bibby, M. (1984). A new finite element model for welding heat sources. *Metallurgical Transactions B*, *15*(2), 299–305. https://doi.org/10.1007/BF02667333

Keller, T., Lindwall, G., Ghosh, S., Ma, L., Lane, B. M., Zhang, F., Kattner, U. R., Lass, E. A., Heigel, J. C., Idell, Y., Williams, M. E., Allen, A. J., Guyer, J. E., & Levine, L. E. (2017). Application of finite element, phase-field, and CALPHAD-based methods to additive manufacturing of ni-based superalloys. *Acta Materialia*, *139*, 244–253. https://doi.org/10.1016/j.actamat.2017.05.003

Knapp, G. L., Coleman, J., Rolchigo, M., Stoyanov, M., & Plotkowski, A. (2023). Calibrating uncertain parameters in melt pool simulations of additive manufacturing. *Computational Materials Science*, *218*, 111904. https://doi.org/10.1016/j.commatsci.2022.111904

Kronbichler, M., & Kormann, K. (2012). A generic interface for parallel cell-based finite element operator application. *Computers & Fluids*, *63*, 135–147. https://doi.org/10.1016/j.compfluid.2012.04.012

Lebrun-Grandié, D., Prokopenko, A., Turcksin, B., & Slattery, S. R. (2020). ArborX: A performance portable geometric search library. *ACM Trans. Math. Softw.*, *47*(1). https://doi.org/10.1145/3412558

Megahed, M., Mindt, H.-W., N'Dri, N., Duan, H., & Desmaison, O. (2016). Metal additive-manufacturing process and residual stress modeling. *Integrating Materials and Manufacturing Innovation*, *5*(1), 61–93. https://doi.org/10.1186/s40192-016-0047-2

Schroeder, W., Martin, K., & Lorensen, B. (2006). *The visualization toolkit (4th ed.)*. Kitware. ISBN: 978-1-930934-19-1

SMESH documentation. (2024). https://docs.salome-platform.org/latest/gui/SMESH/index.html

Team, T. Trilinos Project. (2020 (accessed May 22, 2020)). *The Trilinos Project Website*.

Tecplot Inc. (2024). *Tecplot documentation*. https://tecplot.azureedge.net/products/360/current/360-users-manual.pdf

The asset-importer-lib documentation. (2024). https://assimp-docs.readthedocs.io/en/latest/

Trott, C. R., Lebrun-Grandié, D., Arndt, D., Ciesko, J., Dang, V., Ellingwood, N., Gayatri, R., Harvey, E., Hollman, D. S., Ibanez, D., Liber, N., Madsen, J., Miles, J., Poliakoff, D., Powell, A., Rajamanickam, S., Simberg, M., Sunderland, D., Turcksin, B., & Wilke, J. (2022). Kokkos 3: Programming model extensions for the exascale era. *IEEE Transactions on Parallel and Distributed Systems*, *33*(4), 805–817. https://doi.org/10.1109/TPDS.2021.3097283