Scalable multi-physics for fusion reactors with AURORA

OPEN ACCESS
IOP Publishing
Plasma Physics and Controlled Fusion
Plasma Phys. Control. Fusion **65** (2023) 024002 (9pp)
https://doi.org/10.1088/1361-6587/aca998

Scalable multi-physics for fusion reactors with AURORA

H Brooks* **and A Davis**

United Kingdom Atomic Energy Authority, Culham, Abingdon, United Kingdom

E-mail: helen.brooks@ukaea.uk

Received 29 April 2022, revised 22 September 2022
Accepted for publication 7 December 2022
Published 21 December 2022

Abstract

To support the design of fusion reactors for energy production, there is an urgent need to develop multi-physics software tools capable of modelling critical tokamak components as a single cohesive whole. Although some loosely-coupled physics tools do exist, these lack fidelity and will fundamentally fail to capture any closed-loop feedback effects between separate physics modules. To address this need, we introduce a new open-source code: AURORA: A Unified Resource for OpenMC (fusion) Reactor Applications. Anticipating the need for high-fidelity simulation of complex physics and geometry mandates the need to target high performance computing from the outset. AURORA has been built upon two demonstrably scalable open-source codes: MOOSE for the provision of finite element analysis and OpenMC for Monte Carlo neutron transport. In this application, the heat deposited by neutrons is calculated by OpenMC and tallied upon an unstructured mesh, providing a source term for transient heat conduction and thermal expansion. MOOSE calculates the corresponding change in temperature and density on the same mesh, whereafter local temperature and density regions are defined via binning in these variables. Finally, these regions are updated within OpenMC as new materials having modified nuclear cross sections. The procedure is subsequently iterated until a desired stopping condition is reached. We present some qualitative results as a proof-of-concept demonstration of AURORA. We further demonstrate that there is no measurable degradation in shared-memory performance arising from wrapping OpenMC in a MOOSE application. While AURORA should provide utility as a standalone tool for coupled thermo-mechanical and neutronics analyses, we view it as a first step towards our ultimate goal of having a single suite capable of capturing non-trivial couplings between the many disciplines—encompassing in addition fluid dynamics, electromagnetism, materials science and chemistry—involved in the simulation of a tokamak.

Keywords: multi-physics, neutronics, HPC

(Some figures may appear in colour only in the online journal)

* Author to whom any correspondence should be addressed.

1. Introduction

Driven by the worldwide challenge to tackle climate change, the technology to develop commercial magnetic confinement fusion power plants is presently accelerating. Going beyond academic questions regarding the feasibility of energy production, the commercial industry is also concerned with the financial viability of electricity generation from such devices. One consequence of this is a move towards more compact designs, with the UK government program favouring a spherical (low aspect-ratio) tokamak [1], targeting delivery of a prototype in 2040. Despite the scientific case being informed by results from the MAST-U experiment [2], the spherical topology is less well-explored than conventional aspect-ratio tokamaks, such as ITER [3] and DEMO [4].

The enormity of the engineering challenge goes beyond matters of plasma stability. Even from a reductionist standpoint that treats the plasma as a source of heat and particles, one must simultaneously consider the transient thermal-mechanical response of the device in the presence of electromagnetic fields, turbulent fluid flows, and material degradation induced by effects, such as thermal-cycling, irradiation-induced dislocations, nuclear transmutations, and tritium embrittlement.

Indeed, the fusion environment presents some of the most extreme conditions to which materials may be subjected. For example, toroidal magnetic fields in ITER can reach values of 12 T and leading to circumferential tension of order 100 MN [3]. Meanwhile, plasma-facing components will experience neutron heat loads of $\gtrsim 0.5\,\mathrm{MW\,m^{-2}}$ with temperatures reaching as high as 650 °C. Taken in combination, such conditions will push traditional engineering simulation tools outside their validation regimes. Data generated by testing facilities such as HIVE [5] and CHIMERA [6] will to some extent alleviate this dearth of information. However, given the short time-frames over which the technology will be developed, it is likely that much engineering analysis will be need to done in-silico; there is therefore an urgent need to develop software in support of this effort.

Already, there exist tools, which by utilising both physical and empirical bounds, can produce optimised and self-consistent parametric computer-assisted design (CAD) models for a given a qualitative concept [7, 8]. These tools target high-throughput computing and the generated CAD models necessarily lack fidelity, with volumes constituting space allocation rather than resolving detailed components. Furthermore, physics models are integrated only weakly, so there is no opportunity to capture the effect of feedback. Although this class of software plays an essential role in quickly constraining the available design space and pinpointing principle parameters, it cannot provide a replacement for the type of high-fidelity engineering analysis, which must also be performed to fully characterise a given component's physical behaviour. It is incumbent upon fusion engineers to aspire towards fully-holistic modelling of the entire device surrounding a fusion plasma, in which all relevant physical domains are tightly-coupled such that the impact of feedback between them may be captured.

Specifically, what is required is a multi-physics and—given the diversity of physics, which must be represented—a multi-scale approach. Ultimately, this will constitute the creation of digital twins: virtual replica of components capable of describing the fully transient response to a set of initial conditions. Once further combined with uncertainty quantification, this will provide confidence in the predictive capabilities of engineering simulation, and assurance that a given design will be fit for purpose.

The digital twinning of an entire tokamak is commonly projected to be an exascale endeavour: it is not dissimilar to the challenge of modelling small modular reactors in fission where equivalent projections have already been established [9]. It is also generally accepted that traditional tools employed in computer-assisted engineering workflows do not scale well even beyond tens of compute cores, let alone the many thousands, which will surely be required. Thus, even if exascale resources were already commonly accessible, the currently available software would not be sufficiently mature to be capable attaining such levels of performance. It should be apparent then that there is a need to develop highly-scalable and multi-physics engineering simulation software tools that are capable of modelling highly complex fusion-relevant components. As a medium-term goal it is appropriate to target components such as breeder blankets, divertors and magnets. Not only are these systems critical to the functioning of a tokamak, but also encompass all areas of essential physics, namely neutronics, thermodynamics, solid mechanics and contact, electromagnetism, fluid dynamics, material damage, and tritium production[1].

Given the need for rapid development it is natural for any new code to take advantage of existing packages where these are available since this considerably reduces the human resource—and thus cost—of the process. In considering which dependencies are acceptable, it is necessary to consider that any software developed must adhere to the following requirements. First, it must be sufficiently user-friendly so as to be easily incorporated into existing workflows. Secondly, since we cannot anticipate exactly what supercomputing hardware will be available in future, the software must be portable between architectures. For example, it would not be appropriate to tailor performance to a specific compiler or chipset at the sacrifice of poor performance on a different system. Thirdly, the fusion industry will be a regulated environment; therefore, it will be necessary to comply to a certain standard of quality (including for example testing suites, continuous integration and documentation). Finally, since the development process will indubitably occur over many years, dependencies must be well-maintained and easily extensible. All these requirements strongly motivate taking an open-source approach.

As a first step towards developing scalable, open-source, multi-physics software for fusion engineering analysis, we present a new code AURORA [11, 12]: 'A Unified Resource

[1] We do not anticipate performing a multiscale simulation of the plasma itself, which already constitutes an exascale problem [10], instead treating it rather as a source of heat and particles upon the surrounding structure.

for OpenMC (fusion) Reactor Applications'. This application has been developed to couple neutron transport calculations provided by OpenMC [13, 14] to the native heat conduction and tensor mechanics modules provided in the MOOSE finite-element framework [15, 16]. This coupling is motivated by the fact that neutron-nuclide interaction cross sections depend upon both the temperature and density of the material. Both of these quantities are however influenced by heat deposited by the neutrons themselves: a temperature increase is driven by conduction from the heat source, and density changes occur via thermal expansion. It is thus desirable to capture these effects within the neutron transport calculations.

The format of the paper is as follows. In section 2 we describe the implementation of AURORA, providing details on dependencies, algorithmic methodology and verification. In section 3 we first present some results, which demonstrate the current capabilities of AURORA in the case study of a helium-cooled pebble bed breeder blanket. In addition, we present an analysis of the shared-memory parallel performance of AURORA as compared to OpenMC. Finally, we summarise and provide an outlook in section 4.

2. Implementation of AURORA

2.1. Dependencies

As already noted, AURORA is built upon two primary dependencies: MOOSE for finite element analysis and OpenMC for neutron transport. MOOSE itself is implemented using libMesh [17] for the encapsulation of mesh and finite element data and PETSC [18, 19] for the solution of discretised partial differential equations. It is highly performant, with parallel scalability having been demonstrated to in excess of 32 000 cores [15].

There are numerous alternative open source libraries available for finite element analysis; these include deal.ii [20], DOLFIN [21], DUNE [22], MFEM [23, 24] and Nektar++ [25]. These options were evaluated in [26] on numerous criteria including performance, portability, extensibility, and support. MOOSE was found to demonstrate competitive, although not necessarily maximal, performance with respect to both memory usage and parallel scalability. However, in the context of all criteria it was found to be the best all-round contender. To be more specific, MOOSE provides numerous native physics modules, and the syntax is such that implementing new kernels is essentially trivial. These features significantly reduce the time cost to set up and validate a problem. Extensive documentation and example usage is available, and community support is actively provided. Furthermore, MOOSE employs extensive testing and continuous integration to conform to the American Society of Mechanical Engineers' software quality standard NQA-1 [27].

OpenMC is a modern, open source Monte Carlo particle transport code. Other well-established Monte Carlo transport codes exist (such as MCNP [28], Tripoli-4 [29], and Serpent [30]) but these require licenses for use and therefore do not meet our requirements. Like MOOSE, OpenMC has also demonstrated parallel scalability, and provides documentation and support. OpenMC implements a stochastic solution to the neutron transport equation using generation of successive batches of particle histories; particles are tracked through geometries represented in terms of cells (volumes) enclosed by surfaces. Quantities of interest (such as flux and heat deposition) may be estimated via tallies by scoring tracks or collisions globally or in a local region. Filters are used to specify the domain over which particles are scored, with possibilities including material, energy or mesh element. Unstructured mesh-filter tallies are supported either via the library MOAB [31, 32] or more recently, libMesh.

Geometries in OpenMC may be represented in one of two ways. Natively, OpenMC supports constructive solid geometry, where cells are the negative or positive space generated from the intersection of one or more first- or second-order analytic surfaces. Particle transport upon more complex geometries that may be represented only using CAD may be supported using DAGMC [33], whereby surfaces are faceted into a triangle surface mesh using MOAB, and particle tracking through the enclosed volumes is performed using ray-tracing algorithms.

2.2. Methodology

AURORA is organised into a driver application, intended to perform finite element analysis, and a sub-application in which calls to the OpenMC C++ API are directly invoked. This structure, with the addition of key data objects and the interactions between them is depicted in figure 1. It is possible to run the sub-application as a standalone, which is equivalent to performing a single run of OpenMC; this feature is also useful for verification purposes, and we take advantage of it in section 3.2 for performance comparisons.

For normal usage, a user should specify a transient system of equations for the variables of interest (anticipated to be temperature and displacements) via MOOSE's `Variable` and `Kernel` systems to solved by the driving application. For the case of heat conduction, the relevant partial differential equation is simply:

$$-\nabla\cdot(k\nabla T)+\rho c_p\frac{\partial T}{\partial t}-\dot{q}=0 \tag{1}$$

where T is the temperature, k is the thermal conductivity, ρ is the density, c_p is the specific heat capacity, and $\dot{q}$ is a volumetric heat source, anticipated to arise from neutron energy depositions. In this case the user might use the MOOSE kernels `ADHeatConduction`, `ADHeatConductionTime`, and `ADMatHeatSource`. However, the formulation in MOOSE is completely flexible and the user must determine what physics is of interest.

The starting geometry upon which the variables are solved is represented by an unstructured tetrahedral mesh, that is in read from file in ExodusII format [34] (as is standard for MOOSE applications). This must be accompanied by a DAGMC surface mesh file of the same geometry in HDF5

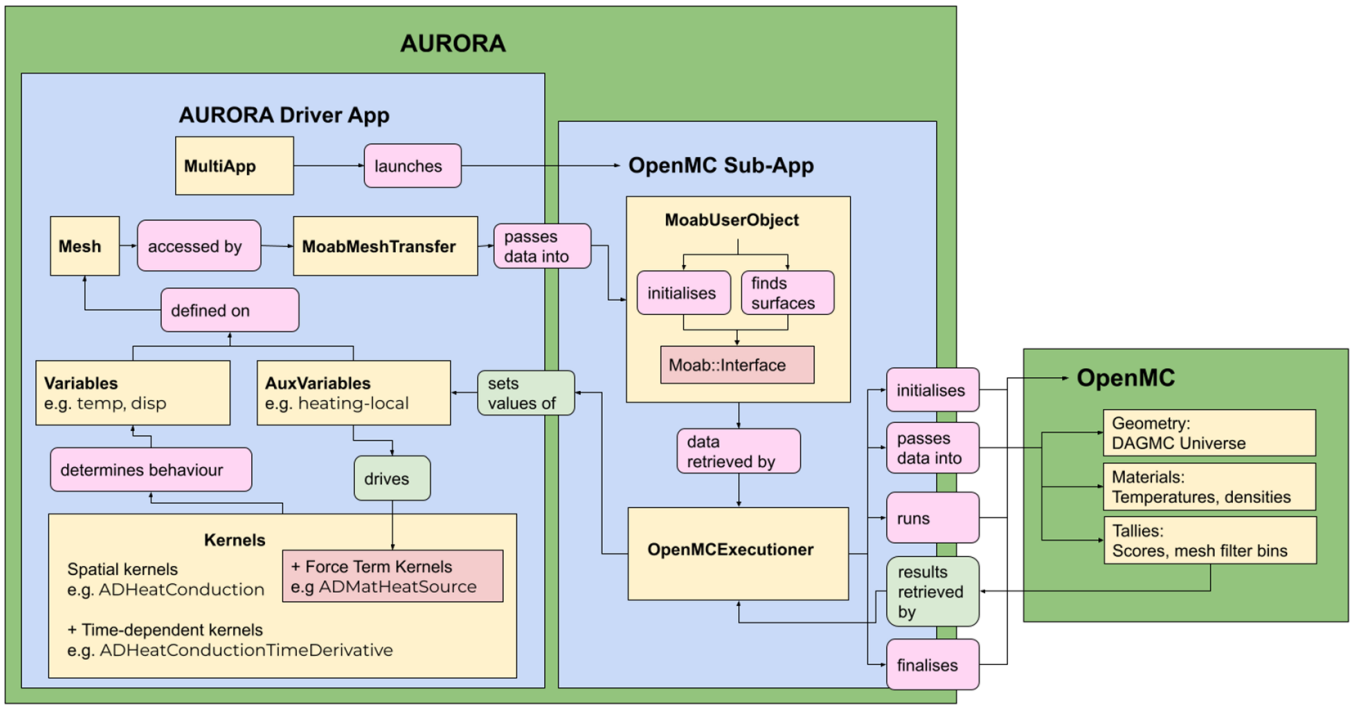


Figure 1. Flow chart depicting the organisation and flow of data between AURORA and OpenMC. Top level-applications are shown in dark green; important data objects are shown in yellow; interactions between objects are shown in pink (for information flow within or from AURORA) and light green (for information flow from OpenMC). Arrows indicate the directionality of information.

format for the initial set-up of OpenMC[2]. Within the sub-application, the mesh data is converted into a MOAB representation and used to initialise a mesh-filter tally in OpenMC. In this conversion there is a one-to-one mapping of nodes and elements[3].

For every score the user wishes to tally upon the mesh in OpenMC (such as neutron heating), they will specify an `AuxVariable` in the MOOSE input card into which they wish the data to be stored. Once initialised, OpenMC is subsequently run and the requested scores are evaluated. Since the mesh-filter tally has a direct correspondence with the MOOSE mesh, each score may be directly retrieved and stored in the corresponding `AuxVariable` in the driver application. These auxiliary variables may now be used to interact with other MOOSE objectsin particular, the heat source in equation (1) may be defined.

After each time step in MOOSE, having solved for temperature and optionally displacements (and consequently, relative density), mesh elements are sorted by their results into user-defined bins. Each material region may thus be split into local regions of approximately fixed temperature and density (and hence constant nuclide cross sections). The outer surfaces of these local regions is obtained through MOAB's so-called 'skinning' operation. A new DAGMC instance is initialised with the resulting surfaces and volumes, with the updated geometry and material properties being passed into OpenMC. In addition, where the mesh has been displaced, the unstructured mesh tally is also updated. OpenMC is re-run and the scores are obtained for the updated materials and geometry.

This procedure may be iterated for as many time-steps as is desirable or until a stopping criterion is reached. Again, since MOOSE is a flexible tool, such a criterion should be specified by the user according to the analysis they wish to perform. For example, one might wish to stop when a component reaches a certain maximum temperature, or one might wish to run until steady state is detected. We refer the reader to the MOOSE documentation (which may be found at https://mooseframework.inl.gov/) for more details.

2.3. *Verification*

Both MOOSE and OpenMC are already independently verified and validated. It is beyond the scope of this work to perform a comprehensive review of the literature here, but for example, the validation of the fuel-performance MOOSE-application BISON, which makes extensive use of the heat conduction and tensor mechanics modules, is described in [35]. Validation of OpenMC was described in [14], and an up-to-date list of benchmarks are maintained in the documentation (https://docs.openmc.org/en/stable/publications.html).

What is primarily needed in addition to this is the *verification* that the wrapping of OpenMC and the identification of new materials surfaces is correct implemented. This has been ensured via a suite of unit and integration tests which are automatically run as part of continuous integration. At the time of

[2] The need for this initial surface mesh file is somewhat redundant as it could be generated from the ExodusII file; for improved usability and consistency, we intend to eliminate this step in future.

[3] We intend to support the more-recently added libMesh unstructured mesh-filter tallies in future, which will avoid duplication of the mesh.

writing, there are 55 tests in the suite, so it is not possible to provide details of every single test, however we review some of the most important checks.

The instantiation of mesh is verified by comparing the number of nodes, elements, and topological features such as volumes and surfaces and the relationships between them to known values. We verify that for known (parametric) spatial distributions of temperature and density, the number of new DAGMC surfaces and volumes constructed, and the metadata associated with them, is correct. We ascertain that prior to normalisation the results for scores obtained in every mesh element are identical to that in OpenMC run alone, by comparing the relative absolute difference to a numerical tolerance of 1×10^{-9}. This is checked for a range of scores, upon both first and second order meshes. We ensure that when the mesh has been deformed through thermal expansion, the change in volume of mesh elements is consistent with that reported by MOOSE postprocessors.

There remains an outstanding task to *validate* how accurately the coupling algorithm described in section 2.2 will describe the thermo-mechanical response of a component subject to the fusion levels of neutron irradiation. However, the identification of a suitable experimental dataset for such a validation is in itself a challenge since as noted by the UK Fusion Materials Roadmap [36] there is no global source of neutrons which will replicate the conditions anticipated in future fusion reactors. Until a suitable neutron source becomes available, it will be necessary to employ creativity in finding alternative sources of data. We therefore postpone such a study to future work; this is reflected the versioning of AURORA using the convention `MAJOR:MINOR:PATCH` and incrementing the major version to exceed 0 only when such a validation has been completed.

3. Results

3.1. Demonstration: helium-cooled pebble bed breeder blanket

As a proof-of-concept demonstration we consider a section of a DEMO-like Helium Cooled Pebble Bed (HCPB) breeder blanket [37, 38]. The geometry comprises tritium breeder pins surrounded by a neutron-multiplier material in a hexagonal arrangement, and a steel support structure. The breeder pin material is taken to be Lithium Orthosilicate (Li_4SiO_4) and the neutron-multiplier material is take to be lead (Pb). The physical dimensions of the model are $1.4 \times 0.8 \times 0.1\text{m}^3$; the tetrahedral mesh used for finite element analysis has 1.6×10^5 degrees of freedom.

The initial temperature is taken to be 293.15 K; we employ heat flux boundary conditions on the surfaces to cooling channels to emulate fluid modelling. The heat transfer coefficients are taken to be 309 and 7.8 $\text{W m}^{-2}\text{K}^{-1}$ on the innermost and outermost cooling channel surface boundaries respectively. The neutron source was distributed uniformly in the $x-z$ plane at $y = 0.88$ m between $x = \pm 0.7$ m and $z = \pm 0.07$ m, having a fixed strength of 10^{17} s^{-1}; the energy distribution spectrum was unchanged relative to default settings, and is given by a Watt spectrum.

A transient analysis for the model was performed with AURORA across 1000 s in time-steps of 30 s. To attain reasonable Monte Carlo convergence, 200 batches of 5×10^6 particle histories were used. In this preliminary study, we did not consider displacements of the mesh, therefore the density of materials is constant. For simplicity, we have also run with photon transport turned off, and the heating was tallied using the `heating-local` score in OpenMC which assumes the energy from photons is deposited locally [39].

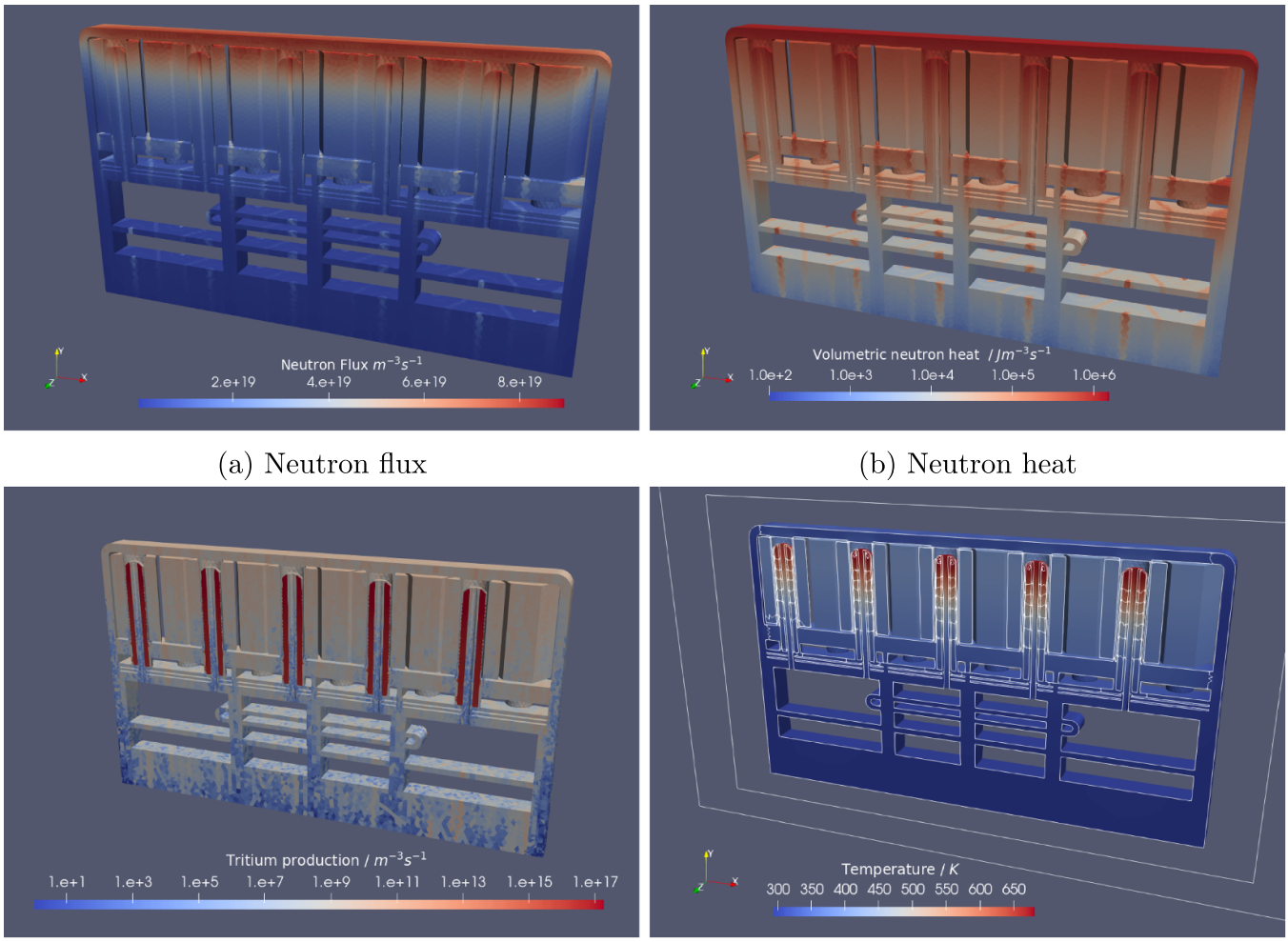
In figure 2 we show a visualisation of some representative results corresponding to the final time-step. The neutron flux, volumetric neutron heat, tritium production, scored by OpenMC and normalised by the source strength in AURORA are shown in figures 2(a), (b) and (d) respectively. The relative error in the OpenMC scores varies across the geometry, ranging from 0.1%–13% for the neutron flux and 0.2%–48% for the heating (with the highest uncertainties corresponding to the innermost corner of the geometry with the lowest flux); the relative uncertainty for tritium production is at a level of 1% in the breeder pin, but as high as 100% in the surrounding material, and thus should be interpreted as an order of magnitude only. No variance reduction techniques were employed here, which would be necessary to achieve better convergence. We note that these results are intended to be simply a demonstration of the software, and should not be interpreted as a realistic physics analysis; we therefore refrain from any quantitative post-processing of the results. However, we can make some qualitative comments.

The neutrons do not propagate far into the component, primarily depositing heat on the innermost volumes, with both scores falling off rapidly with depth. By design, tritium is produced predominantly in the breeder pins, falling by 8 orders of magnitude crossing into the adjacent material. The temperature, superimposed with a slice through the DAGMC material surfaces to reveal the temperature contours, is shown in figure 2(d). The steel support structures having high thermal conductivity and being closest to the cooling surfaces remain at a relatively constant temperature, while the pins heat up, displaying a fairly linear temperature gradient. Correspondingly several new 'material' boundaries are created in the pins.

The precise impact of the temperature feedback upon the neutron transport calculation depends on numerous factors: the choice of nuclear data selected; the method of temperature treatment within OpenMC (interpolation or windowed multipole); the nuclide composition of selected materials; the incident neutron energy spectrum. Enabling a systematic study of these effects is precisely the purpose of AURORA, but was beyond the scope of this initial work; however this will be a priority for future work.

3.2. Shared-memory parallel performance

OpenMC and AURORA support both task-based (MPI) and shared-memory (OpenMP) parallelism. OpenMC scales better with threads, a statement we justify later, so in this



(a) Neutron flux (b) Neutron heat

(c) Tritium production (d) Temperature

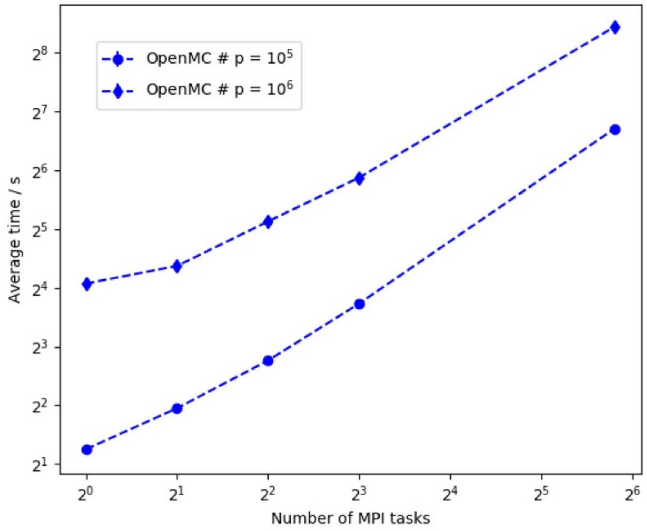
Figure 2. Representative AURORA results for the HCPB breeder blanket model at $t = 1000$ s visualised with Paraview [40]. Figures (b)–(d) are reproduced with permission from [12]. Copyright (2022) by the American Nuclear Society, La Grange Park, Illinois.

section we investigate the shared-memory parallel performance attained by AURORA as compared to OpenMC. To obtain a more direct comparison, only the sub-application wrapping OpenMC was run; this nevertheless includes conversion of mesh data from libMesh to MOAB, plus both the construction and results extraction of the mesh-filter tally, therefore constituting a non-trivial comparison.

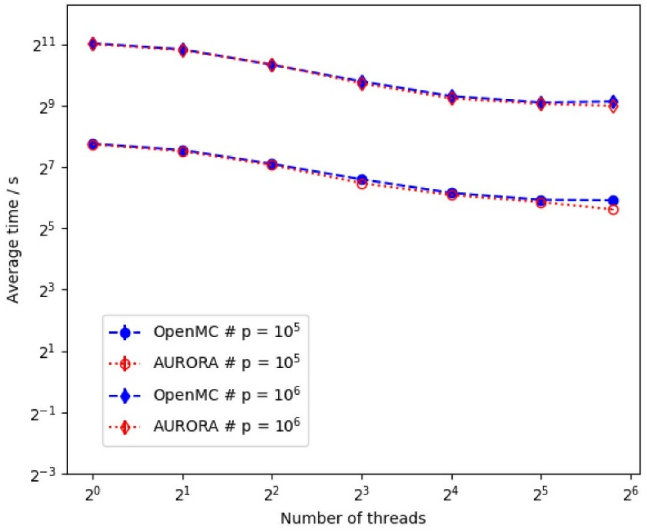
Results were obtained by running on a single Cascade Lake node of the Peta4 supercomputer at the Cambridge Service for Data Driven Discovery (CSD3), each node having 56 CPU. Use of a single node ensures isolation from the impact of inter-node communication. Irrespective of MPI tasks and OpenMP threads used, the entire node was reserved to achieve stable timings. The C++ compiler used was gcc 7.2; the MPI provider was OpenMPI 3.1.3. MPI tasks were launched with the option `--bind-to none`, and the following OpenMP environment variables were set: `OMP_PROC_PLACES = cores` and `OMP_PROC_BIND = close`. For these initial studies, OpenMC transport was performed upon a toy geometry that was quite small, consisting of only 2 non-void volumes; the volume mesh has only $\sim 10^5$ tetrahedral mesh elements and the surface mesh has $\sim$2000 facets. In all cases, timing results were obtained by averaging over ten identical runs. In each run there were ten batches of either 10^5 or 10^6 particle histories.

Each particle history in a given batch is independent, so particle numbers can be split equally between the available processes and threads. Therefore, OpenMC should in principle be embarrassingly parallel. In practice, there are operations which cannot be parallelised such as initialisation and communication, and given the smallness of problem size we are especially sensitive to impact of this. It is certainly notable from figure 3(a) that for a *fixed* total number of processes (i.e. $\{\text{MPI tasks}\} \times \{\text{threads per task}\} = 56$)[4] the total application run time increases with number MPI tasks, presumably arising from increased communication. We subsequently now restrict our considerations to a pure shared-memory parallelism.

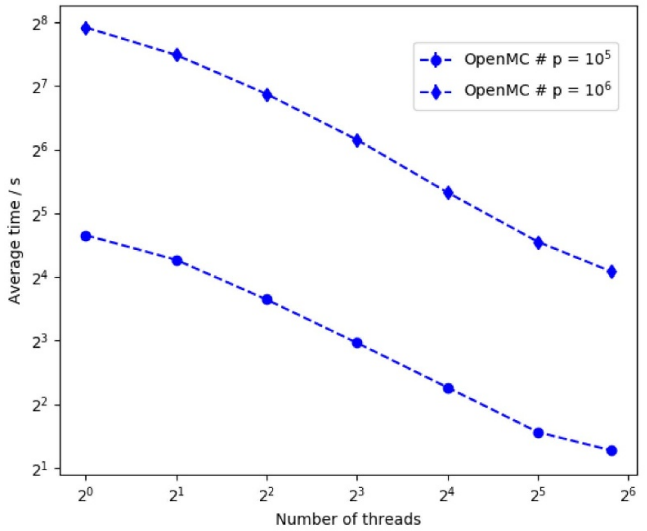
[4] For example, $\#\text{MPI} = 1 \Rightarrow \#\text{threads} = 56$; $\#\text{MPI} = 2 \Rightarrow \#\text{threads} = 28$ etc.



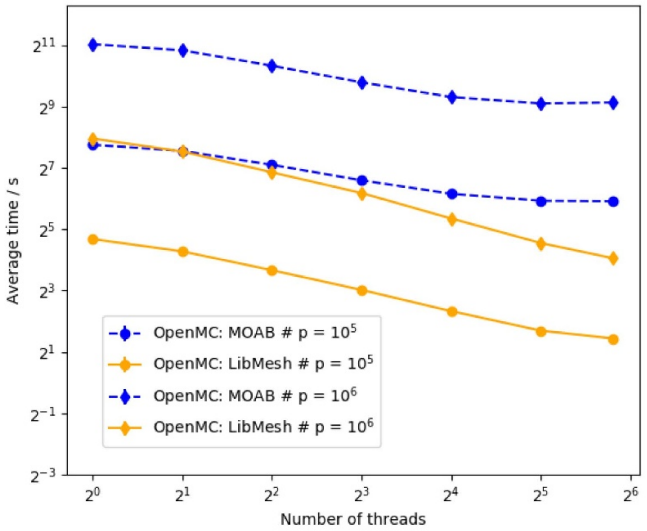
(a) OpenMC total application run-times without mesh tallying for fixed total number of processes equalling 56 (MPI tasks multiplied by threads per task).



(b) Comparison of total application run-times between OpenMC and AURORA sub-application.



(c) OpenMC total application run-times without mesh tallying with pure shared-memory parallelism.



(d) Comparison of OpenMC total application run-times using libMesh and MOAB for mesh tallying.

Figure 3. Plots of total application run time for OpenMC and the AURORA sub-app for increasing numbers of MPI tasks (a) and threads (c)–(b).

From figure 3(b) we conclude that AURORA can reproduce the application times of OpenMC across numerous threads i.e. the wrapping of OpenMC within a MOOSE application does not introduce any significant overheads. This provides confidence that any future acceleration of OpenMC may be leveraged directly in AURORA. It is notable however, that the scaling demonstrated in figure 3(b) is far from ideal. We can in part account for this however. In figure 3(c) total application times for pure particle-transport runs, namely in the absence of scoring any mesh tallies, are shown as a function of threads. Although still not ideal, there is a vast improvement in scalability.

Therefore it is the scoring of an unstructured mesh tally using MOAB that dramatically increases runtimes and reduces scalability. However as can be observed in figure 3(d), when instead the meshing library used is libMesh, run times are more reasonable, resembling much more closely the scaling seen in the purely particle transport results of figure 3(c).

AURORA does not yet support libMesh mesh-tallying since it was not available at the time of development, however there is clearly strong motivation to support this in future, since it would likely result in an immediate performance improvement. Furthermore, this would reduce, although not completely remove, dependence upon MOAB: it would still be required for skinning and as a DAGMC dependency until such a time as alternatives for these become available. With this in place it will be possible to tackle larger problems which require splitting across multiple nodes. Finally, we note that OpenMC has been shown to demonstrate weak scaling to many thousands of cores [14] for the Hoogenboom benchmark [41]; while admittedly this involved a CSG geometry of a fission reactor core, we do not anticipate significant issues in tackling more realistic problems for fusion.

4. Summary and outlook

To support engineering design and analysis of fusion reactors there is an urgent need to develop scalable software tools for multi-physics and multi-scale simulation. Our new code AURORA represents a first step towards this goal, focusing on tightly-coupled neutronics and thermomechanics. To facilitate a rapid development that is also portable, extensible and maintainable, we take an open-source approach. We have selected dependencies that are user-friendly, community-driven and maintained to high software quality standards: the MOOSE framework for finite element analysis and application organisation, and OpenMC with particle-tracking on generic CAD models enabled by DAGMC. We have shown some proof-of-concept results for a HCPB breeder blanket model to demonstrate current capabilities. Furthermore we have demonstrated that no significant overheads result from the wrapping of OpenMC in a MOOSE application.

Already AURORA constitutes a powerful standalone tool that can be used to investigate the impact of temperature and density upon neutron transport and vice versa. Indeed, a more involved study on this topic will be a priority for future work. However, also currently under development at https://github.com/aurora-multiphysics are a suite of other tools, currently encompassing tritium transport (Achlys), electromagnetism (Apollo, Hephaestus), turbulent fluid dynamics (Proteus) and fast ions (Phaeton).

The advantage of retaining a modular organisation permits each tool to be first validated independently. By developing these tools within a common framework it should be possible to ultimately combine them into a unified multi-physics simulation environment for fusion reactor modelling. Through its use of multi-apps, MOOSE makes the nesting of applications trivial. In some cases it may be necessary to offload to an external solver such as MFEM [23, 24] as in the case of Hephaestus, or invoke external models for provision of additional sources of heat and particles, such as ASCOT [42] in Phaeton; however, such procedures will always occur within the context of a Picard iteration.

In the near-term future, studies of interest may include a coupling to fluid dynamics to investigate the impact of neutronics on cooling (and vice versa). In addition, scores of tritium production can already be digested by Achlys to perform transport and trapping modelling, thereby enabling an investigation on the impact of temperature on tritium breeding.

In pursuit of high performance, an immediate improvement is likely possible by the addition of support for the recently implemented libMesh unstructured mesh tallies in OpenMC. To attain more nuanced perform improvements it will first be necessary to perform systematic benchmarking with larger problems having more mesh elements and simulating greater numbers of particles. For models of extreme size these will need to be distributed across multiple compute nodes and employ hybrid parallelism.

Going forwards, one of the biggest challenges faced in the development of multi-physics software for fusion will be the identification of validation scenarios that probe the extreme regimes where tight coupling is necessary. Facilities such as HIVE and CHIMERA, and will provide invaluable sources of data, and digital twinning of these devices will constitute an important milestone. The validation of scenarios involving neutronics tightly-coupled to other physics is especially challenging due to the lack of neutron data at fusion energies and fluxes; it will be necessary to identify alternative sources of data, such as those in high-energy physics and nuclear fission.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

This work has been funded by STEP, a UKAEA programme to design and build a prototype fusion energy plant and a path to commercial fusion. We would like to acknowledge Sergio Sádaba, Sara Haouala, Rodrigo Herrero (IDOM Nuclear Services) for providing the CAD model of the HCPB blanket used to generate results.

ORCID iD

H Brooks https://orcid.org/0000-0002-7690-9857

References

[1] UK Government, Department for Business, Energy and Industrial Strategy 2021 Towards fusion energy: the UK fusion strategy (available at: https://www.gov.uk/government/publications/towards-fusion-energy-the-uk-fusion-strategy)

[2] UK Government 2021 First results from UK experiment point to a solution to one of fusion's hottest problems (available at: https://www.gov.uk/government/news/first-results-from-uk-experiment-point-to-a-solution-to-one-of-fusions-hottest-problems)

[3] IAEA 2001 *Summary of the ITER Final Design Report* (*ITER EDA Documentation Series No. 22*) (Vienna: International Atomic Energy Agency) (available at: https://www.iaea.org/publications/6442/summary-of-the-iter-final-design-report)

[4] EUROfusion 2018 European research roadmap to the realisation of fusion energy (available at: https://www.euro-fusion.org/fileadmin/user_upload/EUROfusion/Documents/2018_Research_roadmap_long_version_01.pdf)

[5] Hancock D, Homfray D, Porton M, Iain Todd B W, Bamber R, Flinders K, Jepson P, Lewtas H and Robinson H 2018 Testing advanced divertor concepts for fusion power plants using a small high heat flux facility (UKAEA-CCFE-PR(18)33) (available at: https://scientific-publications.ukaea.uk/wp-content/uploads/Preprints/UKAEA-CCFE-PR1833.pdf)

[6] Barrett T R, Carrelli C, Grant T, Kovari M, Mantel N, Muir A and Surrey E 2020 *IEEE Trans. Plasma Sci.* **48** 1432–8

[7] Kovari M, Kemp R, Lux H, Knight P, Morris J and Ward D 2014 *Fusion Eng. Des.* **89** 3054–69

[8] Coleman M and McIntosh S 2019 *Fusion Eng. Des.* **139** 26–38

[9] Merzari E, Fang J, Shaver D, Lan Y H, Min M, Fischer P, Rahaman R and Romano P 2020 Initial full core SMR simulations with NekRS *Technical Report* ANL-20/72 (Argonne National Laboratory)

[10] Suchyta E *et al* 2022 *Int. J. High Perform. Comput. Appl.* **36** 106–28

[11] Brooks H and Davis A (available at: https://github.com/aurora-multiphysics/aurora)

[12] Brooks H, Dixon S and Davis A 2022 Towards multiphysics simulations of fusion breeder blankets *Int. Conf. on Physics of Reactors 2022 (PHYSOR 2022) (Pittsburgh, PA, 15–20 May)* (American Nuclear Society) pp 2480–9

[13] Romano P K, Horelik N E, Herman B R, Nelson A G, Forget B and Smith K 2015 *Ann. Nucl. Energy* **82** 90–97

[14] Romano P K and Forget B 2013 *Ann. Nucl. Energy* **51** 274–81

[15] Permann C J *et al* 2020 *SoftwareX* **11** 100430

[16] Gaston D R *et al* 2015 *Ann. Nucl. Energy* **84** 45–54

[17] Kirk B S, Peterson J W, Stogner R H and Carey G F 2006 *Eng. Comput.* **22** 237–54

[18] Balay S, Gropp W D, McInnes L C and Smith B F 1997 Efficient management of parallelism in object oriented numerical software libraries *Modern Software Tools in Scientific Computing* ed E Arge, A M Bruaset and H P Langtangen (Boston, MA: Birkhäuser Press) pp 163–202

[19] Balay S *et al* 2022 PETSc web page (available at: https://petsc.org/)

[20] Arndt D *et al* 2021 *J. Numer. Math.* **29** 171–86

[21] Logg A and Wells G N 2010 *ACM Trans. Math. Softw.* **37** 1–28

[22] Blatt M *et al* 2016 *Arch. Numer. Softw.* **4** 13–29

[23] Anderson R *et al* 2021 *Comput. Math. Appl.* **81** 42–74

[24] MFEM: modular finite element methods (available at: https://mfem.org/)

[25] Moxey D *et al* 2020 *Comput. Phys. Commun.* **249** 107110

[26] Dubas A 2020 Evaluating exascale FEA backends for fusion digital twins *1st Spanish Fusion HPC Workshop* (available at: https://hpcfusion2020.bsc.es/sites/default/files/abstracts/Aleksander_Dubas_Evaluating_Exascale_FEA_Backends_for_Fusion_Digital_Twins.pdf)

[27] Slaughter A E, Permann C J, Miller J M, Alger B K and Novascone S R 2021 *Nucl. Technol.* **207** 923–30

[28] Werner C J *et al* 2021 MCNP version 6.2 release notes (available at: https://www.osti.gov/biblio/1419730)

[29] Diop C, Petit O, Dumonteil E, Hugot F X, Lee Y K, Mazzolo A and Trama J C 2007 Tripoli-4: a 3D continuous-energy Monte Carlo transport code *Trans. Am. Nucl. Soc.* **95** 661

[30] Leppänen J, Pusa M, Viitanen T, Valtavirta V and Kaltiaisenaho T 2015 The Serpent Monte Carlo code: Status, development and applications in 2013 *Ann. Nucl. Energy* **82** 142–50

[31] Tautges T J, Meyers R, Merkley K, Stimpson C and Ernst C 2004 MOAB: a mesh-oriented database SAND2004-1592 *Sandia National Laboratories Report* (Sandia National Laboratories)

[32] Mahadevan V, Grindeanu I, Jain R, Shriwise P and Wilson P 2020 MOAB v5.2.1 (Zenodo) (https://doi.org/10.5281/zenodo.2584862)

[33] Davis A, Shriwise P and Zhang X 2020 DAG-OpenMC: CAD-based geometry in OpenMC pp 395–8 (https://doi.org/10.13182/T32104)

[34] Schoof L A and Yarberry V R 1994 *EXODUS II: A finite element data model* (https://doi.org/10.2172/10102115) (available at: www.osti.gov/biblio/10102115)

[35] Williamson R L *et al* 2021 *Nucl. Technol.* **207** 954–80

[36] UK Fusion Materials Roadmap 2021 *Technical Report* (UKAEA and Henry Royce Institute) (UKAEA and Henry Royce Institute) (available at: https://www.royce.ac.uk/content/uploads/2021/09/UK_Fusion_M aterials_Roadmap_Interactive.pdf)

[37] Hernández F A, Pereslavtsev P, Zhou G, Neuberger H, Rey J, Kang Q, Boccaccini L V, Bubelis E, Moscato I and Dongiovanni D 2019 *Fusion Eng. Des.* **146** 1186–91

[38] Hernández F A *et al* 2020 *Fusion Eng. Des.* **157** 111614

[39] Abdou M, Maynard C and Wright R 1973 MACK: computer program to calculate neutron energy release parameters (fluence-to-kerma factors) and multigroup neutron reaction cross sections from nuclear data in ENDF format *Technical Report* ORNL-TM–3994 (Oak Ridge National Laboratory)

[40] Ayachit U 2015 *The Paraview Guide: A Parallel Visualization Application* (Clifton Park, NY: Kitware, Inc.)

[41] Hoogenboom J E, Martin W R and Petrovic B 2011 The Monte Carlo performance benchmark test—aim, specifications and first results *M and C 2011: Int. Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (Brazil)*

[42] Hirvijoki E, Asunta O, Koskela T, Kurki-Suonio T, Miettunen J, Sipilä S, Snicker A and Äkäslompolo S 2014 *Comput. Phys. Commun.* **185** 1310–21