# Scalable Tightly-Coupled Multi-physics for Fusion Reactors with AURORA

H. M. Brooks1, A. Davis1

1United Kingdom Atomic Energy Authority, Culham, Abingdon, United Kingdom

Email contact of corresponding author: helen.brooks@ukaea.uk

1. **Introduction: Tightly-Coupled Multiphysics for Fusion Reactors**

With the advent of numerous planned magnetic confinement fusion plants [1-3] having significant political clout [4] the question of fusion’s feasibility to provide a sustainable clean energy source is no longer the sole remit of academia [5]. While the concern of plasma stability remains an important and active domain of research, beyond this there exists the significant challenge of modelling potential fusion reactors as a single cohesive system. The goal to create a digital twin [6] of a tokamak or even a subset of its critical components is motivated by the need to predict, evaluate, and ultimately optimise, the performance of competing reactor designs.

 To assist the design process there already exist software packages such as PROCESS [7] and BLUEPRINT [8]. The former is a systems code which identifies and optimises self-consistent parameter sets given physical constraints implemented as consistency equations and inequalities. The latter is a tool to automate the construction of a reactor design configuration; initially leveraging PROCESS to perform a low-fidelity analysis, subsequently constructing a two-dimensional reactor profile, and finally defining a complete three-dimensional CAD geometry. Such tools are invaluable in constraining the available parameter space and configuring design points. However, such static tools lack temporal resolution and will ultimately fail to account for feedback loops between its loosely-coupled physics modules. It is therefore necessary to develop a tool capable of capturing non-trivial couplings between multiple disparate physics models, ranging across disciplines such as neutronics, thermodynamics, fluid dynamics, tensor mechanics, materials science and chemistry. As the first step in the development of such a tool, we introduce a new open-source code, AURORA [9]: A Unified Resource for OpenMC (fusion) Reactor Applications.

 AURORA is built upon two open-source packages: the main application is built using the MOOSE finite element framework [10]; this application wraps an interface to OpenMC [11] for neutronics calculations. In the following we discuss some of the considerations for the selection of such tools. Subsequently we present the method employed for capturing the effect of time-dependent temperature and density fields upon particle transport. Finally we review the current status and consider the outlook for future developments.

1. **Consideration of Open Source Dependencies**

There is understandably a cost to the high degree of fidelity required for multiphysics simulations for fusion reactors.

The first cost is human, in the sense that there is a conceptual barrier in realising such complex systems programmatically. It is therefore necessary to reduce the cognitive burden and accelerate the path to development by utilising existing specialist tools which have already refined the implementation of a given numerical method or physics module. Naturally the combination of multiple tools and the passing of data between them minimally requires (1) common file formats and metadata conventions, and (2) exposed programming interfaces.

The second - and perhaps more obvious - cost is computational; interleaving codes and performing numerous iterations while retaining a fine resolution of complex geometries is unsurprisingly expensive in both CPU time and memory. Indeed, it is anticipated that simulation of a tokamak will require exascale levels of computation. Thus the ability of a library to leverage parallelism and demonstrate a high degree of scalability with processes and threads is a central requirement to its selection. Furthermore, since the application will ultimately require deployment on cutting-edge supercomputers, the target architecture is as yet unknown and as such there is a requirement for the software to have a high degree of portability. This necessitates flexibility and the software cannot be limited to proprietary operating systems.

All of these considerations - reusability, accessibility, scalability, portability, and flexibility - necessitate development within an open-source environment.

As previously stipulated, simulating the fusion reactor environment requires modelling of numerous physical systems. As a starting point, it is essential to calculate the heat deposited from neutrons produced in fusion reactions throughout the entire reactor geometry. We rely upon the modern transport code, OpenMC [11] to perform such calculations. (Although other well-established transport codes exist such as MCNP [12], however in this case distribution of the source code is restricted.) OpenMC in turn depends upon DAGMC [13] for acceleration of particle transport upon CAD geometries via surface ray tracing and MOAB [14] for meshing.

Secondly, in order to model the effects of the deposited neutron heat upon the reactor via heat conduction, advection, and thermal expansion we employ the finite element framework MOOSE, which itself depends upon LibMesh [15] for meshing and data structures relevant to finite element analysis (FEA), and PETSc [16] for linear algebra and preconditioning. There exist numerous tools for FEA, of which a thorough comparison was performed in [17]. MOOSE was selected due to its extensive available physics modules, extensibility and parallel scalability.

1. **Algorithm to Couple Time-dependent Fields to OpenMC using MOOSE**

Nuclear cross sections for neutron interactions depend upon material density and temperature. In OpenMC, these fields, once defined for a given cell in the geometry, are deemed to be fixed. However, the deposited heat from neutrons drives an increase in temperature and subsequently local density changes from thermal expansion. It is thus necessary to repeat the calculation of deposited heat within OpenMC with updated temperature and density fields. It is precisely this feedback loop that we encapsulate in AURORA. In what follows, we elaborate on the method to implement this feedback.

1. We start with an initial unstructured tetrahedral mesh of a given geometry in Exodus II [18] format, which is read into MOOSE and data is contained in a libMesh object.
2. The mesh is converted in memory into a MOAB data structure; an initial surface mesh is also instantiated.
3. OpenMC is initialised as follows:
	* DAGMC is initialised upon the surface mesh, whose surfaces define the boundaries between regions (cells) of fixed temperature and material density in OpenMC.
	* The elements of the tetrahedral mesh define regions upon which to score tallies of interest, such as heat flux.
4. OpenMC is run, and results for tallies such as heat flux are collated.
5. The results for heat flux are converted into a MOOSE variable object; this can be done with ease since there is a one-to-one representation in terms of degrees of freedom of the tallying mesh in OpenMC and the “finite element” variable in MOOSE.
6. MOOSE is run to calculate the change in temperature and density using the variable set in step (5) as a heat source.
7. Given some predefined binning in temperature and density, elements in the mesh are organised into local regions where these variables may be considered fixed.
8. The surfaces of the local regions found in step (7) are found.
9. The unstructured mesh (which may have deformed from thermal expansion), along with the DAGMC surfaces and cells constructed in step (8) are updated in OpenMC.
10. The process is repeated from (4) for as many iterations (time steps) as are desired, or required by the problem at hand.

We note that in this approach there is an unavoidable duplication in the representation of the mesh in memory. There is no theoretical reason that this need to be the case; it is simply a by-product of the fact that MOOSE depends upon libMesh and DAGMC depends upon MOAB. There is in principle scope for providing libMesh support in DAGMC (though this would represent a significant update); this step may become essential if calculations become memory-limited.

1. **Status and Outlook**

AURORA currently exists as a proof-of-concept, and is available for download from GitHub [9]. The most significant outgoing challenge in the development of such complex tools is likely to lie with VVUQ: verification, validation, and uncertainty quantification; the utility of multiphysics software is irrelevant if the simulations they produce are not trustworthy. The first of these challenges, verification, may be alleviated by using best-practice methods of software development procedures, including: use of unit, integration and regression tests; use of containers for stable dependency sets; continuous integration; documentation; frequent cycles of tagged releases (using a convention such as MAJOR:MINOR:PATCH) to aid reproducibility.

The remaining challenges of validation and uncertainty quantification are themselves significant tasks. The former depends on comparison against and tuning to experimental data, which in the context of fusion reactors (outside of plasma research) is somewhat limited. A degree of resourcefulness in identifying suitable experiments for comparisons will be needed.

The latter task pertains to the inherent uncertainties present on empirical data used for input, and propagating these through simulations in the form of parameter ranges and scans. Propagating uncertainties analytically is a theoretical problem, while performing large parameter scans becomes computationally expensive. It may be necessary to develop surrogate models to make the latter approach a viable one; this avenue is one that will benefit from leveraging existing open-source libraries since this is a common problem in data science.

Here we have restricted ourselves to a discussion on the interplay between neutron transport and the impact of heating. However, numerous of other tools are currently in development within the MOOSE framework, intended to incorporate the effects of electromagnetism (APOLLO [19]), fast ion fluxes (PHAETON), tritium trapping and diffusion, and turbulent fluid flow through Reynolds-averaged Navier Stokes. By using a consistent and demonstrably performant open-source framework, and in particular one which by design permits a trivial coupling of different applications [20], we aim to combine these tools to ultimately provide sophisticated multiphysics modelling for fusion reactor devices.

**Acknowledgements**

This work was funded by the Spherical Tokamak for Energy Production (STEP) project, which is an initiative of the United Kingdom Atomic Energy Authority (UKAEA) effort to design and construct a prototype fusion energy plant, and demonstrate commercial viability of fusion.

We acknowledge support from authors of OpenMC, in particular Patrick Shriwise, Paul Romano and April Novak in facilitating code changes needed to develop our interface to OpenMC and general advice. We also acknowledge the MOOSE authors for providing user support and responding to issues.

**References**

1. INTERNATIONAL ATOMIC ENERGY AGENCY, ITER Technical Basis, ITER EDA Documentation Series No. 24, IAEA, Vienna (2002).
2. FEDERICI, G. *et al*, “Overview of the DEMO Design-Staged Approach in Europe”, Nuclear Fusion, 59(6):066013, (2019).
3. “UK hatches plan to build world's first fusion power plant”, Nature, doi: https://doi.org/10.1038/d41586-019-03039-9
4. RHS Report on Fusion Energy, (2021); https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\_data/file/990445/rhc-fusion-report.pdf
5. “Fusion power is attracting private-sector interest”, The Economist, (2019) https://www.economist.com/science-and-technology/2019/05/04/fusion-power-is-attracting-private-sector-interest
6. “Cheat sheet: What is Digital Twin?”, IBM, (2020); https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/
7. KOVARI, M., KEMP, R., LUX, H., KNIGHT, P., MORRIS, J., WARD, D., “PROCESS: A systems code for fusion power plants—Part 1: Physics”, Fusion Engineering and Design, 89: 3054–3069 (2014); KOVARI, M., FOX, F., HARRINGTON, C., KEMBLETON, R., KNIGHT, P., LUX, H., MORRIS, J., “PROCESS: A systems code for fusion power plants – Part 2: Engineering”, Fusion Engineering and Design, 104: 9–20 (2016)
8. COLEMAN, M., MCINTOSH, S., “BLUEPRINT: A novel approach to fusion reactor design”, Fusion Engineering and Design, 139: 26-38 (2019)
9. BROOKS, H., DAVIS, A., <https://github.com/aurora-multiphysics/aurora>
10. PERMANN, C., GASTON, D., ANDRS, D, CARLSEN, R., KONG, F., LINDSAY, A., MILLER, J., PETERSON, J., SLAUGHTER, A., STOGNER, R., MARTINEAU, R., “MOOSE: Enabling massively parallel multiphysics simulation”, SoftwareX, 11:100430, (2020)
11. ROMANO, P.K., HORELIK, N.E., HERMAN, B.R., NELSON, A.G., FORGET, B., SMITH, K. OpenMC: A state-of-the-art Monte Carlo code for research and development. Ann. Nucl. Energy 82:90-97 (2015)
12. WERNER, C. , "MCNP Users Manual - Code Version 6.2", LA-UR-17-29981 (2017).
13. https://github.com/svalinn/DAGMC
14. https://sigma.mcs.anl.gov/moab-library/
15. KIRK, B., PETERSON, J., STOGNER, R.,CAREY, G., “libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations”, Engineering with Computers, 22(3--4):237--254, (2006)
16. BALAY, S., GROPP, W., CURFMAN MCINNES, L., SMITH, B., “Efficient Management of Parallelism in Object Oriented Numerical Software Libraries”, Modern Software Tools in Scientific Computing, Birkhauser Press (1997); https://www.mcs.anl.gov/petsc
17. DUBAS, A., “Evaluating Exascale FEA Backends for Fusion Digital Twins”, 1st Spanish HPC Fusion Workshop, Barcelona, November 2020.
18. SCHOOF, A., YARBERRY, V., “Exodus II: A Finite Element Data Model”, SAND92-2137 (1995); https://github.com/gsjaardema/seacas#exodus
19. BLAIR, A., DAVIS, A. https://github.com/aurora-multiphysics/apollo
20. GASTON, D., PERMANN, C., PETERSON, J. SLAUGHTER, A., ANDRS, D., WANG, Y., SHORT, M., PEREZ, D., TONKS. M., ORTENSI, J., ZOU, L., MARTINEAU, R., “Physics-based multiscale coupling for full core nuclear reactor simulation”, Annals of Nuclear Energy, 84:45–54 (2015)