Science and Technology Facilities Council

Techniques and software relevant to the coupling of continuum (fluid) and particle models of plasma for NEPTUNE

Technical Report 2060049-TN-02
Deliverable 3.1

Hussam Al Daas* Tyrone Rees* Andrew Sunderland† Emre Sahin†
Sue Thorne‡

March 2022

1 Introduction

The aim of the "Fusion Modelling" use case within ExCALIBUR is the development of new algorithms, software and related infrastructure that will allow for the efficient use of current Petascale and future Exascale supercomputing hardware to draw insights for ITER, and guide and optimise the design of STEP, the UK demonstration nuclear fusion power plant, and related fusion technology. With existing legacy codes not scaling well and not containing all of the required latest physics, significant investment is needed. However, it is important that the different pieces of the modelling and simulation strategy must not only scale well in isolation, these pieces need to couple together and limit any losses in scalability.

1.1 The limitations of existing legacy codes

Figure 1 highlights the areas of the tokamak simulation that are being targeted by the NEPTUNE project. The lack of the latest physics in current legacy codes means that kinetic effects within the "burning plasma regine" are not included and the design of the "divertor" region of future fusion power plants will be impossible using an in-silico design approach. However, this is the only feasible approach that can be used to design the tokamak.

Within the UK, researchers currently make extensive use of fluid-based codes when modelling the edge-plasma region of fusion devices. However, the electrically charged nature of the plasma adds significant complications when using a fluid approach and add significant uncertainty to the models, particularly, when close to the wall. In these regions, a particle-based (kinetic) method such as particle-in-cell (PIC) is preferable. It is therefore necessary to couple together two different approaches for different regions of the plasma. With regards the NEPTUNE Project, it is important to note that it is intended that the same grid is used within both the fluid and particle simulations.

In Section 2, we review some of the methods available within the literature for performing this coupled approach. We then review some of the available code coupling libraries to see if there are any "off the shelf" solutions for these problems, Section 3.

*The authors are with the Scientific Computing Department, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK.

†The authors are with the Hartree Centre, STFC Daresbury Laboratory, Sci-Tech Daresbury, Keckwick, Daresbury, Warrington, WA4 4AD, UK.

‡Sue Thorne is with the Hartree Centre, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK. Email contact: sue.thorne@stfc.ac.uk

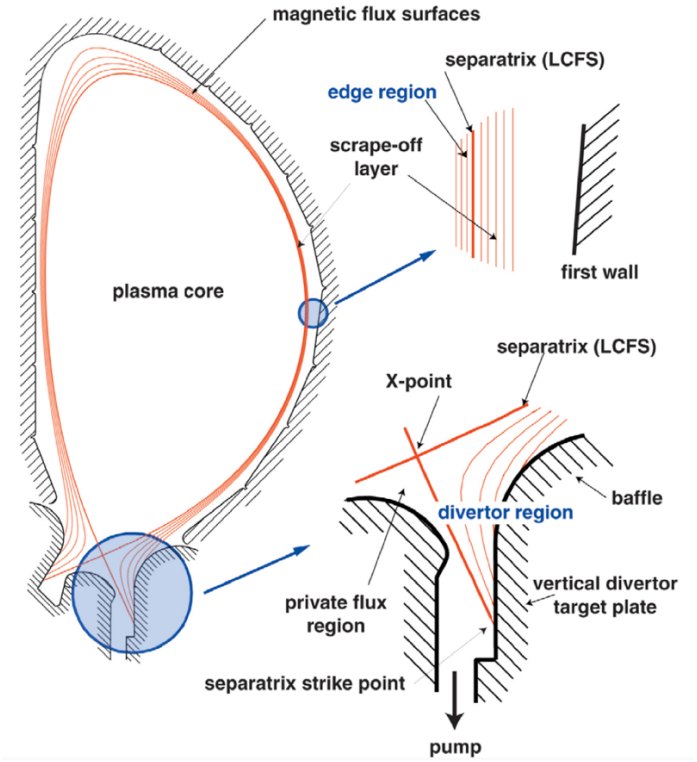

Figure 1: Schematic diagram of a generic tokamak "poloidal cross section". The shaded circles highlight the areas of plasma and first wall that are being targeted by the NEPTUNE project. Attribution: G. Federici et al. [CC BY 3.0 (`https://creativecommons.org/licenses/by/3.0`), minor modifications to figure].

2 Methodologies for coupling the fluid and particle simulations

The authors of [1] have considered a number of methods for hybridising the fluid and particle (concentrating on the use of EIRENE [2]) approaches. The fluid approach is, in general, used to simulate plasma ions and electrons; kinetic (particle) or hybrid approaches are used for the neutrals including molecular species. The authors do review the possibility of using a purely fluid treatment of the neutral atoms (taking moments of the kinetic equation), which is known as the **advanced fluid neutral (AFN) model**: they conclude that whilst there are benefits in terms of computational costs, the modelling error can be quite significant, particularly in lower recycling regimes.

In the following, it will help to define $\mathbf{K_n}$, the Knudsen number. In fusion-relevant plasmas, let λ be the mean free path for neutrals and $\mathbf{L}$ be the gradient length, then the Knudsen number is defined as $\mathbf{K_n} = \lambda/\mathbf{L}$. For large values of $\mathbf{K_n}$, there is no accurate fluid closure available for the neutrals, which is why a particle approach is used. In high-collision regions, $\mathbf{K_n}$ becomes low and there is very strong coupling of the neutrals with the background plasma.

2.1 Spatial Hybridization (SPH) Approach

The regions where $\mathbf{K_n}$ is sufficiently small to allow the use of the fluid model typically occur in only a small part of a dense divertor. A number of *spatially hybrid* approaches have been proposed in literature and usually have distinguish between the particle and fluid regions, coupling the regions together by using proper boundary conditions at the interfaces: how this is done varies between the different methods.

The choice of interface between the two regions poses interesting questions. It reported that it must be such that the boundary conditions can be defined based on fluid-like distributions but it must also ensure that we do not lose too much computational speed. This poses an interesting question when considering Exascale simulations: how to spread the work for the particle and fluid approaches across an HPC system to optimally load balance the whole simulation? We discuss this further in Section 3.

2.2 Micro-Macro Hybrid (mMH) Approach

In [3], the authors propose splitting the (complete) neutral distribution function f into a fluid part f^f and a kinetic correction f^δ such that $f = f^f + f^\delta$ holds across the entire simulation domain. For the fluid-part, the approach leads to the AFN equations but the kinetic correction leads to additional corrections on source and transport terms. A modified kinetic simulation is used to obtain the kinetic correction: positive and negative correction particles are used with net zero density and momentum, which provide exact closure for the AFN model. This approach has benefits, namely, the method has been shown to be produce solutions that are equivalent to solving the fully kinetic equations and is independent of the recycling regime used. However, the method requires a significant amount of developmental effort and the fluid method will need to act across the whole of the domain instead of just a subdomain. The use of such an approach would require significant changes in the approach currently being used by some of the NEPTUNE projects but, given the interesting nature of this methodology, it is being explored by Felix Parra for NEPTUNE.

2.3 Kinetic-Diffusion Monte Carlo Method (KDMC)

The Kinetic-Diffusion Monte Carlo (KDMC) scheme is an asymptotic-preserving scheme that is fully Monte Carlo. Each Monte Carlo particle alternates between acting according to a fluid limit (moving via a random walk that incorporates a Monte Carlo descritization of the limiting fluid equation) and acting in a kinetic manner. Whilst this method has merit in that it does not require intricate couplings between fluid and kinetic neutral sub-domains, it has so far been only demonstrated on simplified problems involving single-species scattering or absorption. The decision to use of such a method would result a significant change of steer for the NEPTUNE project.

3 Available software libraries for code coupling

If an approach such as SPH is used, then there are a number of code coupling libraries already available at it is unlikely that the NEPTUNE project would have to develop their own library. In this section, we review some of the available libraries and note that the Software Outlook Project [4] performed an extensive review of a number of code coupling libraries [5] and refer the reader to performance comparisons within their report for further details.

3.1 MUI

The Multiscale Universal Interface (MUI) is an open source code-coupling library, which was originally developed by Brown University [6]. Today, MUI's primary developers and maintainers are UKRI-STFC [7]. The MUI project aims to create " . . . a light weight plugin library that can glue together essentially all numerical methods including, but not limited to, Finite Difference, Finite Volume, Finite Element, Spectral Method, Spectral Element Method, Lattice Boltzmann Method, Molecular Dynamics, Dissipative Particle Dynamics and Smoothed Particle Hydrodynamics" [6].

MUI provides a small set of programming interfaces to conduct send and receive messages between domains. As such, it does not put restrictions on the multi-physics solvers themselves; "MUI also follows PLE's philosophy of not prescribing how the physics of a problem should be coupled, only that coupling should be achieved by passing data through an interface. Assuming the data in question is associated with a point structure then the only requirement is that the data can be associated with a single point in space" [8]. A header-based approach is used for the entire library and the only external library is the Message Passing Interface (MPI). As such, it can be used in the same way any other C++ standard library would be used, without the need for pre-compilation. However, it has the bonus advantage of not interfering with pre-existing MPI communications within the pre-existing solver.

3.2 OpenPALM

OpenPALM (Projet d'Assimilation par Logiciel Multimethodes) is another coupling library that enables a user to execute components of code concurrently with communication between them. There are two parts to OpenPALM: PrePALM and PALM. PrePALM is the graphical user interface, and PALM is the driver of coupling framework itself. PALM uses MPI communication to handle the exchange of data for the coupling algorithm.

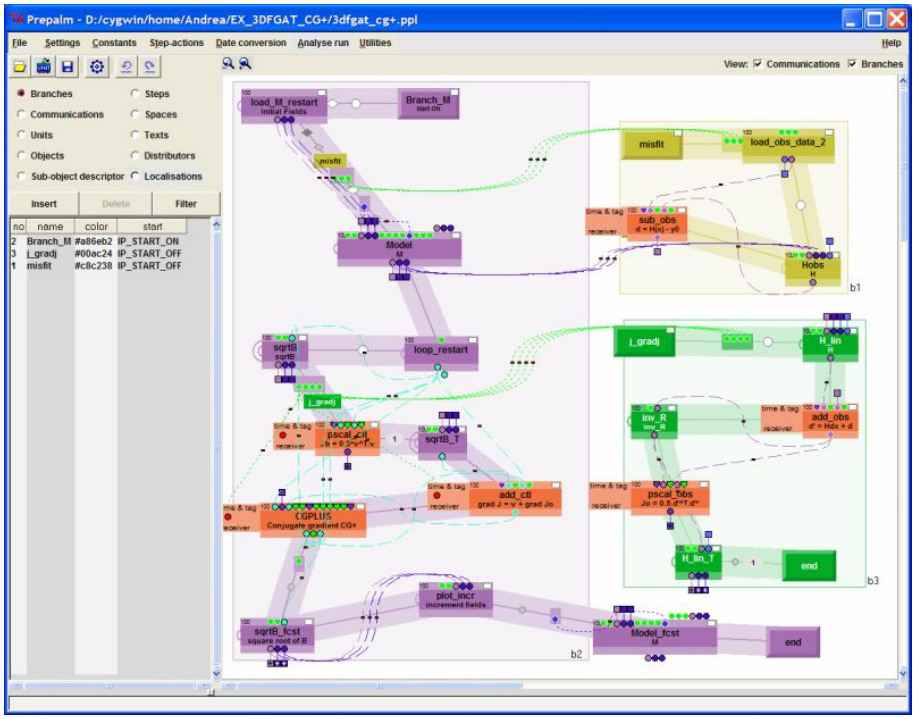

Figure 2: The PrePALM Interface. This example was sourced from OpenPALM's website [9].

On the OpenPALM website, it states they have had simulations of 130,000 cores on Titan, 130,000 cores on Turing and 12,000 cores on Curie [9]. Keyes et al. [10] includ OpenPALM as one of the successes in multi-physics software. OpenPALM does not appear to be restricted to particular scientific fields and it has a lengthy user guide [11]. Therefore, OpenPALM maybe of interest to the NEPTUNE Programme.

The OpenPALM team is joint between CERFACS and ONERA. Cerfacs have also created another coupler called OASIS [12] but it is dedicated to geophysical applications. In contrast to OASIS, OpenPALM provides a more generic interpolation framework based on an unstructured mesh formalism [13], provided by the CWIPI library [14].

OpenPALM applications are implemented via a user interface called PrePALM. The user designs their coupling algorithm into sequential and parallel sections, loops, conditional executions, and communication between components. The designed algorithm is presented to the user clearly in a window as seen in Figure 2. OpenPALM stress you can create parallel code "...without anything to know about MPI, only by drawing! It is one of the PALM features: one can make parallel computing without any further specific knowledge" [11].

3.3 preCICE

The Precise Code Interaction Coupling Environment (preCICE) [15] is another available open source coupling library that aims to couple existing solvers together, creating what they call 'partitioned' simulations. In this way, they have similar motivation to MUI (Section 3.1) and wante the highest flexibility possible in reusing existing components. Their team have particular interests in fluid-structure interaction and conjugate heat transfer simulations, but they stress that they are not limited to such fields. The development of preCICE has been done by doctoral candidates from the Technical University of Munich and the University of Stuttgart.

To couple your code with preCICE, an input file must be developed. preCICE is configured with XML file: a reference guide is available [16]. One particular differnce between preCICE and the other coupling libraries that we discuss is that in this XML file, you can establish an `m2n` communication channel (i.e. from `m` processes of `Solver1` to `n` processes of the `Solver2`) based on TCP/IP sockets. It is possible to configure preCICE to use MPI ports if you wish, but the preCICE team recommend using `sockets`. Once this XML configuration is complete, the user can then insert calls to the preCICE API in original solvers and, at this stage, preCICE is very similar to MUI.

3.4 PLE

The Point Location Exchange library, PLE, is part of Code_Saturne [17]: an open-source CFD software released by EDF. PLE was specifically designed to simplify parallel couplings using a minimalist API and and minimal dependencies. The library allows location of points on meshes and the team compare themselves to MUI and preCICE [18]. It is used to couple Code_Saturne to other EDF codebases, such as EDF's thermal software SYRTHES [19]. Code_Saturne is available on GitHub [20], with PLE included in this repository [21]. PLE featured in CIUK in 2019 [22] in a presentation about Exascale CFD code and it has Doxygen documentation [23].

The PLE team have developed a framework through which data can be passed between different solvers. To do this, PLE creates an interface between two solvers and an MPI communicator is created for each interface, thus allowing pre-existing MPI functionality within the original solvers. Data is transferred between solvers at specific spatial points [8]. In this way, it works in a very similar way to MUI in Section 3.1. We note that within the NEPTUNE Programme, the multiple domains are planned to use the same underlying mesh structure at the interfaces.

Unfortunately, apart from the Doxygen documentation, PLE does not have any easily obtainable user guide to help new users to learn how to use PLE. Regretfully, because of the lack of support in this way, it is difficult to justify recommending PLE as the coupling library that a team should use if they have never used it before. While it does appear to be a current and popular choice, we would recommend using MUI over PML: the two libraries work in a very similar way and MUI has more user support.

4 Conclusions

To conclude, of the three algorithmic approaches considered within Section 2, the spacial hybridization (SPH) and micro-macro hybridization (mMH) approaches are of the most interest. The latter approach with involve significantly larger software development time within the NEPTUNE Programme, with the current set-up of NEPTUNE more closely aligning to the SPH approach. Additionally, the open source libraries considered in Section 3, align with the SPH approach, which should also reduce the software development efforts required so long as the fluid and kinetic/particle projects are scoped in such a manner that enables the use of one of these libraries.

References

[1] Dmitriy V. Borodin, Friedrich Schluck, et al. "Fluid, kinetic and hybrid approaches for neutral and trace ion edge transport modelling in fusion devices". In: *Nuclear Fusion* (2021).

[2] D. Reiter, M. Baelmans, et al. "The EIRENE and B2-EIRENE Codes". In: *Fusion Science and Technology* 47.2 (2005), pp. 172–186.

[3] Niels Horsten, Giovanni Samaey, et al. "A hybrid fluid-kinetic model for hydrogenic atoms in the plasma edge of tokamaks based on a micro-macro decomposition of the kinetic equation". In: *Journal of Computational Physics* 409 (2020), p. 109308.

[4] *Software Outlook Website*. URL: `https://www.softwareoutlook.ac.uk/`.

[5] Philippa Rubin. *Comparison of Code Coupling Libraries for High Performance Multi-Physics Simulation*. Tech. rep. ADD. Science and Technology Facilities Council, 2022.

[6] Yu-Hang Tang, Shuhei Kudo, et al. "Multiscale Universal Interface: A concurrent framework for coupling heterogeneous solvers". In: *Journal of Computational Physics* 297 (Sept. 15, 2015), pp. 13–31.

[7] *Hartree Centre - Multiphysics and Multiphase Code Coupler | Scalable and flexible code coupling*. URL: `https://www.hartree.stfc.ac.uk/Pages/scalable-and-flexible-code-coupling-.aspx`.

[8] Stephen Longshaw, Alex Skillen, et al. "Code Coupling At Scale: Towards The Digital Product". In: May 30, 2017, p. 28. ISBN: 978-1-874672-72-2.

[9] *OpenPALM Website*. URL: `https://www.cerfacs.fr/globc/PALM_WEB/index.html`.

[10] David E Keyes, Lois C McInnes, et al. "Multiphysics simulations: Challenges and opportunities". In: *The International Journal of High Performance Computing Applications* 27.1 (Feb. 1, 2013), pp. 4–83.

[11] Thierry Morel, Florent Duchaine, et al. *OpenPALM coupler version 4.3.0 User guide and training manual*. Apr. 2019. URL: https://www.cerfacs.fr/globc/PALM_WEB/pdfs/user_guide.pdf.

[12] *OASIS Website*. URL: https://portal.enes.org/oasis (visited on 02/27/2021).

[13] *CERFACS Research Areas: Code Coupling and User Interface*. URL: https://cerfacs.fr/en/code-coupling-and-user-interface/.

[14] *The CWIPI coupling library*. URL: https://w3.onera.fr/cwipi/bibliotheque-couplage-cwipi.

[15] *preCICE Website*. URL: https://precice.org/index.html.

[16] *preCICE XML reference*. URL: https://precice.org/configuration-xml-reference.html.

[17] *Code_Saturne Website*. URL: https://www.code-saturne.org/cms/.

[18] Yvan Fournier. "Massively Parallel Location and Exchange Tools for Unstructured Meshes". In: *International Journal of Computational Fluid Dynamics* 34.7 (Sept. 13, 2020), pp. 549–568.

[19] *EDF R&D Simulation Software Webpage*. EDF France. URL: https://www.edf.fr/en/the-edf-group/world-s-largest-power-company/activities/research-and-development/scientific-communities/simulation-softwares.

[20] *Code Saturne GitHub Repository*. URL: https://github.com/code-saturne/code_saturne.

[21] *PLE GitHub Repository*. URL: https://github.com/code-saturne/code_saturne/tree/master/libple.

[22] *SCD CIUK 2019 Presentations*. URL: https://www.scd.stfc.ac.uk/Pages/CIUK-2019-Presentations.aspx.

[23] *PLE (Parallel Location and Exchange) documentation*. URL: https://www.code-saturne.org/cms/sites/default/files/docs/ple-2.0/html/index.html.