Science and Technology Facilities Council

Implementation and scalability analysis of coupling continuum (fluid) and particle models of plasma for NEPTUNE

Technical Report 2068625-TN-06
Deliverable D3.3

Josh Williams* Sue Thorne†

March 2023

1 Introduction

Modelling nuclear fusion is a multiscale problem [2, 6], where the hot plasma core is treated as a continuum coupled to a kinetic simulation in the edge region due to the larger mean free path between molecules. These two regions share a common interface that can be modelled through coupling appropriate boundary conditions. The location of the interface is dependent on the Knudsen number, $K_n = \lambda / L$ where λ is the mean free path of the neutrals and L is a characteristic length scale. In low K_n regions ($K_n < 0.001$), the plasma can be treated with a continuum approach. When $K_n > 10$ (known as free-molecular flow), the plasma is rarefied and no longer fluid-like [9]. Then, the particle motion must then be modelled in a kinetic approach and properties such as the electron number density can be gathered into a grid format (to couple with the continuum) using a particle-in-cell approach [4]. To maximise the potential of exascale HPC architectures, the parallel performance of continuum-kinetic coupling should be investigated and optimised. In our previous report [5], we reviewed relevant coupling libraries to arrive at the decision to use CWIPI [14] for coupling due to its lightweight nature and it is already implemented within Nektar++. Coupling simulations for CWIPI have been reported for coupling turbulent flow simulations to acoustic solvers [8, 10]. In this report, we aimed to (i) implement a continuum-kinetic simulation coupling with CWIPI, and (ii) evaluate the scalability of our implementation.

2 Methods

2.1 Problem setup

We used Nektar++ version 5.3.0 [3, 11] to solve the incompressible Navier-Stokes equations in a 2D periodic domain. Our simulation is based on a case provided by UKAEA (Hermes-3 calculation [7]), where the equivalent of a ball of hot gas is initialised in the centre of the domain and rises upwards based on the Grashof number, $Gr = 5 \times 10^9$. The source term for velocity is $F_u = Gr\,T$, where T is the temperature. In our NESO-Particles [13] simulation, we initialised $N_p = 10^7$ where N_p is the number of particles in the domain. A visualisation of the 'u' field after 100 timesteps is shown in **??**. The NESO-Particles solver advects the particles with a random velocity ($\mathcal{N}(0,1)$), and calculates the number of particles per cell. This is transferred to Nektar++ as a source term for temperature, since this is analogous to electron density (when a plasma solver is used for the continuum phase).

In Nektar++, we discretised the domain using 10^6 square elements and 6th order polynomial to solve the fluid transport. We discretised the NESO-Particles domain into 256 coarse cells, which were subdivided twice, giving 4096 cells for calculating the particle properties. The timestep size was $\Delta t = 2 \times 10^{-8}$ s for the fluid and particle simulations. For our scalability analysis, we integrated the particles and fluid for 10 timesteps, with coupling performed at each timestep. The default settings for linear solvers were used, which uses a diagonal preconditioner.

*Josh Williams is with the Hartree Centre, Edinburgh Royal Observatory, Blackford Hill, Edinburgh, EH9 3HJ, 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

2.2 Implementation

The NESO-Particles example, provided by UKAEA, was coupled to Nektar++ by calling the `coupling_cwipi.hpp` header file [12]. This header file contains all of the necessary CWIPI calls to couple two simulations, such as initialising the coupling and sending or receiving data. CWIPI also required information on the NESO-Particles mesh connectivity. Therefore, we added a function to create a mesh with connectivity information to the `main.cpp` file. This connectivity information could also be included in the core NESO-Particles release in future iterations. Our implementation is available on GitHub [12].

NESO-Particles was compiled with hipSYCL v0.9.4 [1] and clang++ version 14.0.6. Nektar++ was compiled with GCC 7.2.0. The OpenMPI library (version 4.1.4) was used for both codes.

We performed scalability analysis on Hartree Centre's Scafell Pike supercomputer. The simulations were performed on compute nodes composed of 2x Intel Xeon Gold E5-6142 v5 (also known as Skylake) 16 core 2.6GHz (up to 3.7GHz), 192 GB RAM. Each core has two threads, allowing 64 processes to be run per node. Nodes are connected with Infiniband.

We note that when running parallel simulations for the coupled analyses, we observed a spurious NaN in the fluid solver in the timestep immediately after the fluid results were written. This was not a deterministic error, but usually arose after data was written at initialisation. As the number of MPI ranks increased, we found the error arose more frequently. As a temporary mitigation, we commented the code related to the `Checkpoint_Output` function in `EquationSystem.cpp`, and set the checkpoint interval in the XML configuration file to be longer than the number of steps the simulation would run for (removing writing the data). This error should be looked into, potentially with the aid of the Nektar++ developers.

3 Results

To understand the parallel performance of the Nektar++/NESO-Particles coupling, we evaluated the speedup and parallel efficiency for simulations on a single compute node, as well as multiple nodes. We calculated the speedup as T_b/T_N, where T_b is the execution time for a baseline simulation and T_N is the execution time for a simulation with N processes. For the single-node simulations, we used the execution time of a simulation with 8 processes as our baseline simulation ($b = 8$). For the multi-node simulations, we used the execution time of a simulation on one node as our baseline ($b = 1$). We calculated the parallel efficiency as

$$e = \frac{T_b}{T_N}\frac{b}{N}. \tag{1}$$

3.1 Single-node results

Figure 1 shows the speedup and efficiency scaling analysis results for a single node. We show this for the coupled simulations, as well as a simulation of only Nektar++ to evaluate the loss in performance due to coupling. At $N = 32$ to 64, the performance difference is negligible. At $N = 16$, coupling appears to reduce the efficiency by 10%. The overall performance is poor for $N > 32$, as the efficiency drops from 90% to 60% for the Nektar++ simulations. Similarly, the efficiency of the coupled simulations dropped from 84% to 62%. The scaling performance of the coupled simulation is in close agreement with the fluid only simulations (Nektar++), showing that, for this example, coupling with the CWIPI library does not influence parallel efficiency significantly.

Figure 2 shows the scalability performance for various parts of the Nektar++ incompressible Navier-Stokes solver. We observe the most significant performance reduction in applying pressure boundary conditions and solving the pressure equation. Applying forcing to and solving the viscous terms also showed poor scalability. Performance in these parts of the code may potentially be improved by optimised preconditioners and linear equation solvers.

3.2 Multi-node results

Similar to the single-node analysis, Figure 3 shows the speedup and efficiency scaling analysis results for simulations ran on multiple nodes. At $N = 128$ (two nodes) the performance difference is negligible for the Nektar++ simulation (95% efficiency, Figure 3b). For two nodes, the coupled simulation had a 22% reduction in parallel efficiency. For four and eight nodes, the scaling performance of the coupled simulation is in close agreement with the Nektar++ only simulation and, again, shows that coupling with the CWIPI library does not influence parallel efficiency significantly on highly parallel tasks. Similar to the single-node analysis, the overall performance is poor as the efficiency is 41% and 50% for the Nektar++ only and coupled simulations on eight nodes, respectively.

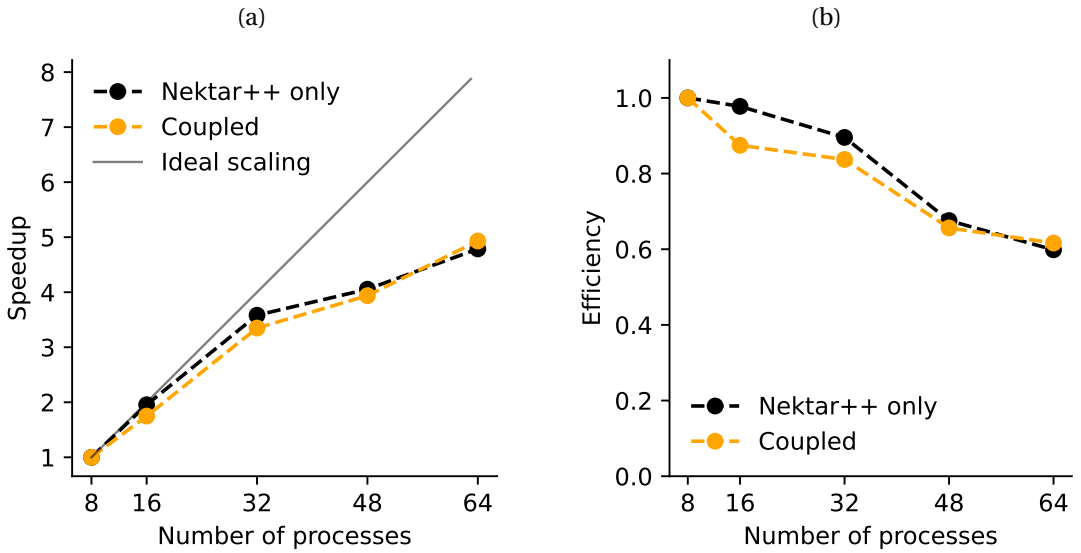

Figure 1: Strong scalability analysis showing performance for thermal plume simulation on a single node. We compare the performance for a coupled simulation with a pure fluid simulation. Each node has 64 slots. Panels show (a) speedup and (b) efficiency.

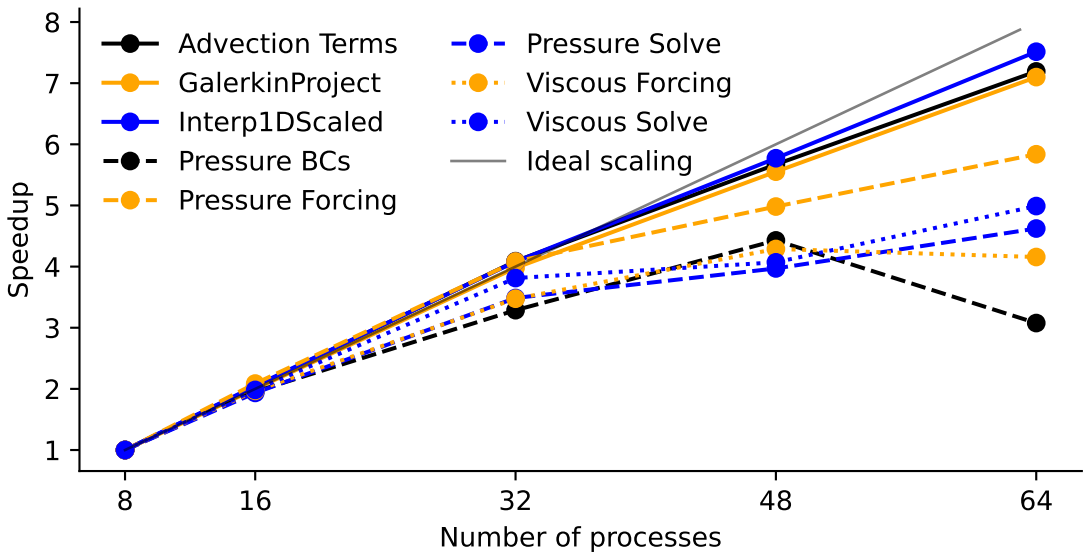

Figure 2: Profiling Nektar++ with a strong scalability analysis on the fluid only (uncoupled) simulation. Scaling results shown for a single node.

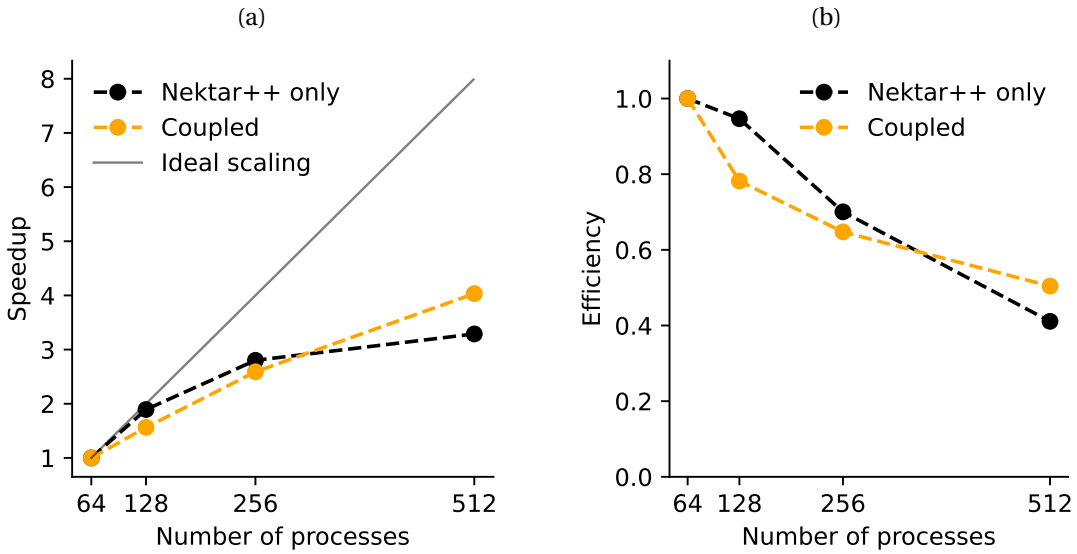

Figure 3: Strong scalability analysis showing performance for thermal plume simulation on multiple nodes. We compare the performance for a coupled simulation with a pure fluid simulation. Each node has 64 slots. Panels show (a) speedup and (b) efficiency.

Table 1: Duration of each step in Nektar++ solver, relative to the total simulation duration. Results shown for a simulation with 512 MPI processes.

Solver step	Relative duration [%]
Advection Terms	1.506
GalerkinProject	0.355
Interp1DScaled	0.522
Pressure BCs	0.006
Pressure Forcing	0.108
Pressure Solve	85.547
Viscous Forcing	0.133
Viscous Solve	11.823

The pressure solver drained the performance significantly in the multi-node analysis, achieving an efficiency of only 37% at $N = 512$ (Figure 4). The scalability for the pressure boundary conditions and the viscous solver cause some moderate loss in performance (66.6% and 65% efficiency, respectively). In Table 1, we also observe that the pressure and viscous solvers have the longest duration in clocktime (as well as poor scaling shown in Figure 4). From this analysis, it is apparent that efficient and scalable solvers for pressure and viscous terms are key to maximising the performance of HPC.

4 Conclusions

This report described our coupling implementation in CWIPI and provided a test-case for scalability benchmarking. The parallel overhead created by coupling compared to a uncoupled (fluid only) simulation was minimal, showing that CWIPI is well-suited to coupling plasma simulations on exascale computing clusters. The overall parallel performance was shown to be poor, as the parallel efficiency on 8 nodes was 50% for the coupled simulation and 41% for the fluid only simulation. From profiling the fluid solver, we observed that the main decrease in performance was due to the scalability of the linear equation solvers for the pressure equation and viscous terms in the velocity equation, which would likely be resolved with improved preconditioners. Future efforts should be made to establish a suitable test case for quantitative validation of the coupling.

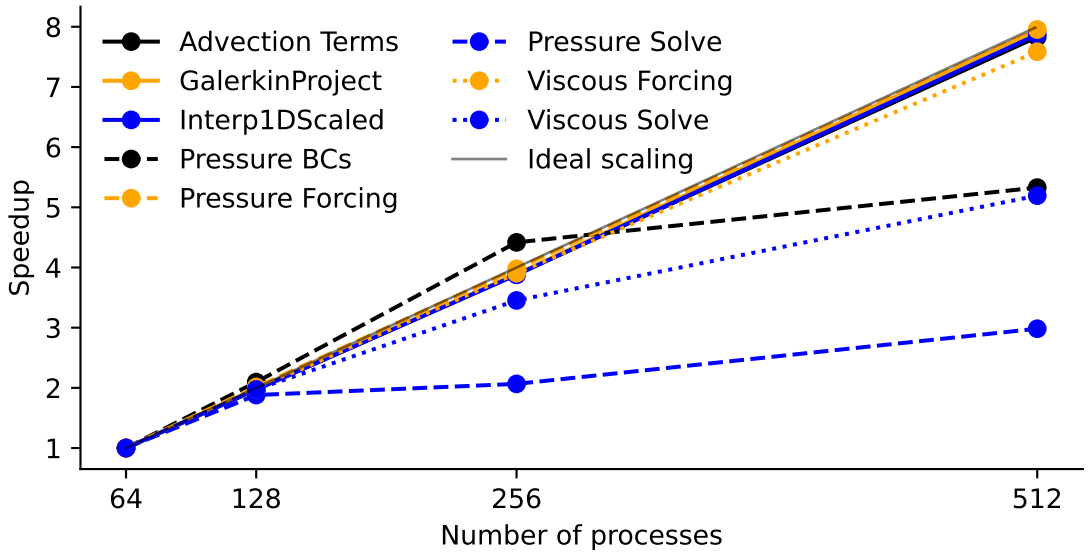

Figure 4: Profiling Nektar++ with a strong scalability analysis on the fluid only (uncoupled) simulation. Results from the multi-node scaling analyses.

References

1. Alpay, A. & Heuveline, V. *SYCL beyond OpenCL: The architecture, current state and future direction of hipSYCL* in *Proceedings of the International Workshop on OpenCL* (2020), 1–1.
2. Barnes, M., Abel, I., *et al.* Direct multiscale coupling of a transport code to gyrokinetic turbulence codes. *Physics of Plasmas* **17,** 056109 (2010).
3. Cantwell, C. D., Moxey, D., *et al.* Nektar++: An open-source spectral/hp element framework. *Computer physics communications* **192,** 205–219 (2015).
4. Cheng, J., Dominski, J., *et al.* Spatial core-edge coupling of the particle-in-cell gyrokinetic codes GEM and XGC. *Physics of Plasmas* **27,** 122510 (2020).
5. Daas, H. A., Bootland, N., *et al. Techniques and software relevant to the coupling of continuum (fluid) and particle models of plasma for NEPTUNE* tech. rep. 2068625-TN-05 (UKAEA, 2023).
6. Hasenbeck, F., Reiser, D., *et al.* Multiscale modeling approach for radial particle transport in large-scale simulations of the tokamak plasma edge. *Procedia computer science* **51,** 1128–1137 (2015).
7. *Hermes plasma edge simulation model: Hermes-3, a hot ion multifluid drift-reduced model.* `https://github.com/bendudson/hermes-3` (2023).
8. Langenais, A., Vuillot, F., *et al.* Assessment of a Two-Way Coupling Methodology Between a Flow and a High-Order Nonlinear Acoustic Unstructured Solvers. *Flow, Turbulence and Combustion* **101,** 681–703. `https://doi.org/10.1007/s10494-018-9928-0` (Oct. 2018).
9. Li, J., Jiang, D., *et al.* Kinetic comparative study on aerodynamic characteristics of hypersonic reentry vehicle from near-continuous flow to free molecular flow. *Advances in Aerodynamics* **3,** 10. `https://doi.org/10.1186/s42774-021-00063-0` (May 2021).
10. Moratilla-Vega, M. A., Angelino, M., *et al.* An open-source coupled method for aeroacoustics modelling. *Computer Physics Communications* **278,** 108420 (2022).
11. Moxey, D., Cantwell, C. D., *et al.* Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. *Computer Physics Communications* **249,** 107110 (2020).
12. *Nektar++-NESO-Particles coupling implementation by STFC* `https://github.com/ExCALIBUR-NEPTUNE/NumericalAnalysis/tree/feature-coupling/Coupling/nektar-cwipi-neso-particles` (2023).
13. Saunders, W. & Cook, J. *NESO-Particles* online repository. Jan. 31, 2023. `https://github.com/ExCALIBUR-NEPTUNE/NESO-Particles` (2023).
14. *The CWIPI coupling library* `https://w3.onera.fr/cwipi/bibliotheque-couplage-cwipi`.