

PROJECT 20-19864

FINAL REPORT, v2

The Development and Application of a Hybrid Domain Overlapping Coupling between SAM and NekRS/STAR-CCM+

Prepared by:

Aaron Huxford¹, Victor Coppo Leite²,
Elia Merzari², Ling Zou³,
Victor Petrov¹ and Annalisa Manera¹

¹University of Michigan

²Penn State University

³Argonne National Laboratory



August 2023

(This page intentionally left blank)

Table of Contents

1 INTRODUCTION	4
1.1 SAM	4
1.2 NekRS	4
1.3 Coupling Tool: Cardinal	5
2 METHODOLOGY AND EQUATIONS	5
2.1 CFD-informed Friction Factor	6
3 COUPLING VERIFICATION	7
3.1 Open-Loop Configurations	7
3.1.1 Momentum coupling tests	7
3.1.2 Energy coupling tests	12
4 STABILITY COMPARISONS	14
4.1 Pump-driven Closed Loop Configuration	15
4.2 Natural Circulation Closed Loop Configuration	16
5 COUPLING VALIDATION	18
5.1 Double T-Junction Experiment	18
5.1.1 Double T-junction setup	18
5.1.2 NekRS model details	18
5.1.3 SAM model details	19
5.1.4 Coupled SAM-NekRS concentration profiles	20
5.1.5 Coupled SAM-NekRS concentration integrals	21
5.2 TALL-3D STH/CFD Benchmarking Experiment	21
5.2.1 TALL-3D setup	22
5.2.2 SAM model details	23
5.2.3 STAR-CCM+ model details and validation	23
5.2.4 SAM-STAR-CCM+ coupling of the TG03.S301.04 transient	25
6 ADDITIONS TO CARDINAL AND SAM	29
6.1 Cardinal's Documentation and Test Suite	30
6.2 SAM's Code Modifications, Documentation and Testing	31
6.3 Continued Cardinal and SAM Work	32
7 CONCLUSIONS	33
References	34
Appendix A SAM-NekRS Coupling Documentation in Cardinal	36

1 INTRODUCTION

The NEUP-19864 project implemented the domain decomposition and hybrid domain overlapping coupling methods between the system analysis code SAM Hu et al. (2021) and the computational fluid dynamics (CFD) code NekRS Fischer et al. (2021a) using and adding to coupling tool Cardinal Merzari et al. (2021). All codes and software are under active development at Argonne National Laboratory (ANL). The coupling's goal is to extend SAM's range of applicability to scenarios when local momentum and energy transfer are important to transient analysis of accident scenarios.

One important problem in nuclear reactor safety analysis is how to compute the evolution of accident scenarios for accurately estimating reactor safety margins. Currently, government regulators accept results from System Thermal Hydraulics (STH) codes if the physical conditions of application are within the code's range of validation. STH codes utilize a simplified, 1D (one-dimensional) description of the nuclear power plant, and to augment the simplified models, they rely on a large number of closure correlations typically based on experimental data. When 3D (three-dimensional) flow effects play an important role in the system, the STH code's simplifying 1D assumptions face many challenges and can even become invalid.

In current and next-generation nuclear reactor systems, components such as the reactor pressure vessel, containment compartments and pools contain 3D flow and mixing effects that can play an important role in the evolution of accident scenarios. For example, during a loss of forced-cooling accident, a Modular High Temperature Gas-cooled Reactor's upper plenum can contain 3D mixing and thermal striping due to reversed flow within the reactor's core (Huning et al. (2021)). During a similar protected loss of forced-cooling accident, a pool-type Sodium-cooled Fast Reactor's sodium pool contains heat and mass transfer that produces thermal stratification, which is inherently a three-dimensional process (Wu et al. (2020)). Computational Fluid Dynamics (CFD) codes can provide more-accurate predictions of complex 3D flow phenomena than STH codes.

Over recent years, CFD codes have been increasingly applied to the simulation of single-phase flow in complex nuclear reactor systems due to the increased maturity of CFD models and increased availability of computational resources (Smith et al. (2008)). However, modeling an entire reactor system with CFD remains prohibitively computationally expensive. As a result, the coupling of CFD with STH codes to take full advantage of CFD's modeling accuracy and STH's computational efficiency is an important undertaking.

1.1 SAM

The present work couples the STH code SAM to the CFD code NekRS using Cardinal. SAM (System Analysis Module) is a modern STH code aimed at advanced non-LWR safety analysis. It aims to provide fast-running, whole-plant transient analyses capability with improved-fidelity for Sodium-cooled Fast Reactors, Lead-cooled Fast Reactors, and Molten Salt Reactors. SAM utilizes the open-source, NQA-1 compliant, object-oriented application framework MOOSE Permann et al. (2020) with its underlying meshing and finite-element library libMesh Kirk et al. (2006), and linear and non-linear solvers PETSc Balay et al. (2022).

1.2 NekRS

NekRS is an open-source, GPU-oriented CFD code based on the spectral element method. State-of-the-art multilevel preconditioners, efficient high-order time-splitting methods, and runtime-adaptive communication strategies are built on a fast OCCA-based kernel library, libParanumal Chalmers et al. (2022), to provide scalability and portability across the spectrum of current and future high-performance computing platforms. NekRS is suited to model a wide range of spatial scales from Direct Numerical Simulations and Large Eddy Simulations to Reynolds-Averaged Navier Stokes (RANS). As an example of NekRS's applicability within the nuclear energy research community, Fischer et al. (2021c) used NekRS to perform full-core Large Eddy Simulations of a pebble bed reactor with more than 350,000 pebbles. Also, Merzari et al. (2020) used NekRS to perform full-core Small Modular Reactor RANS simulations, showing NekRS's range of fidelity.

1.3 Coupling Tool: Cardinal

The in-memory data transfers between SAM and NekRS are handled by Cardinal. Cardinal is an open-source wrapping of NekRS within the MOOSE framework Permann et al. (2020) and is used for coupling NekRS to other MOOSE-native or MOOSE-wrapped applications. One clear advantage of using the MOOSE framework for coupling is MOOSE’s MultiApp system, Gaston et al. (2015) which facilitates code execution and data transfer timing. Cardinal uses a generic data transfer implementation that allows NekRS to be coupled to any MOOSE application, enabling a broad set of multiphysics and multiscale capabilities. Example uses of Cardinal include the following: Merzari et al. (2021) performed coupled Monte Carlo, thermal fluids and fuel performance simulations in pebble bed reactors; Novak et al. (2021) performed coupled conjugate heat transfer simulations for bypass flow modeling in pebble bed reactors; Novak et al. (2022a) performed coupled Monte Carlo and thermal hydraulics simulations of a prismatic gas reactor fuel assembly; and Novak et al. (2022b) performed coupled Monte Carlo transport and conjugate heat transfer simulations for wire-wrapped fuel bundles.

2 METHODOLOGY AND EQUATIONS

The present work utilizes an explicit temporal coupling implementation with both domain decomposition and hybrid domain overlapping methods to couple mass, momentum, energy, and passive scalar equations of single-phase incompressible flows (Huxford et al. (2022)). An explicit coupling method is utilized because it requires less computational resources than the semi-implicit coupling method, as well as far less user intervention into the CFD code. Diagrams of the domain methods are shown in Fig. 1, where \mathbf{U} is velocity, P is pressure, T is temperature and f_{CFD} is a CFD-informed friction factor. The main difference between methods is that the domain overlapping method allows the STH code to model the entire domain, including the CFD-modeled portion. The CFD domain’s solution is used to pass a CFD-informed friction factor to the overlapping STH component, which is expanded on in Section 2.1.

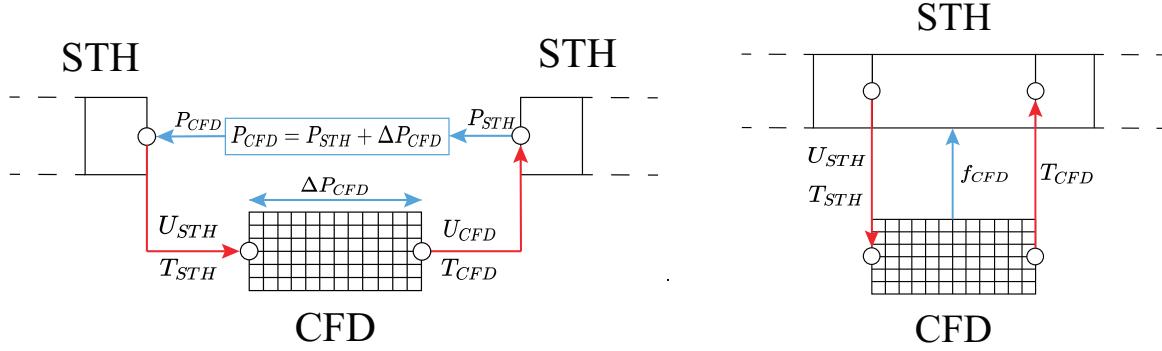


Figure 1: Coupling schematics for the domain decomposition (left) and hybrid domain overlapping (right) methods.

The domain overlapping method proposed by Grunloh and Manera (2016) was extended to coupling of the energy equation and passive scalar equations utilizing a domain decomposition-like approach. Unlike the strong coupling between velocity and pressure fields, temperature and passive scalars are less-coupled in space, such that a domain decomposition-like coupling is sufficient. That is, the present hybrid domain overlapping coupling method uses an overlapping methodology for momentum coupling and a domain decomposition methodology for energy and passive scalar coupling.

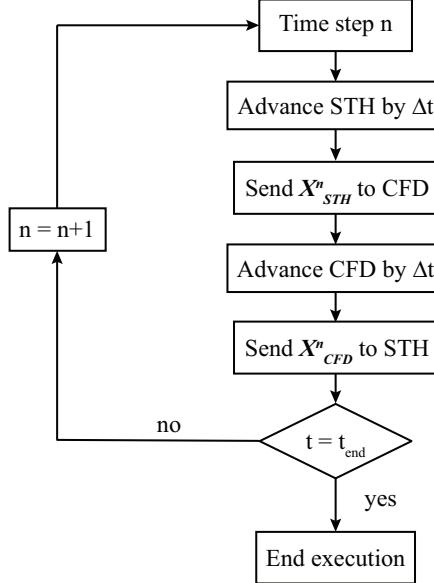


Figure 2: Explicit coupling scheme implemented.

The explicit coupling scheme implemented is shown in Fig. 2. The terms X_{STH}^n and X_{CFD}^n reflect data coming from the STH code and CFD code, respectively. Their description depends on the domain method, decomposition or overlapping, and are written in Table 1. Data coming from the CFD code are surface-averaged quantities over their corresponding boundary surface. For passive scalar coupling, additional scalars are transferred using the same method as temperatures T_{STH}^n and T_{CFD}^n .

Table 1: Data transfers for each coupling method.

Coupled	Coupling Method		
	Term	Decomposition	Overlapping
X_{STH}^n	U_{STH}^n, T_{STH}^n	U_{STH}^n, T_{STH}^n	
X_{CFD}^n	$U_{CFD}^n, P_{CFD}^n, T_{CFD}^n$	f_{CFD}^n, T_{CFD}^n	

2.1 CFD-informed Friction Factor

Care needs to be taken when implementing the domain overlapping's f_{CFD} because the STH code may utilize either the Fanning or Darcy friction factor, which differ by a factor of four. To derive f_{CFD} , the authors follow the same procedure as Grunloh and Manera (2016). First, the CFD, NekRS momentum conservation equation and the equivalent STH, SAM 1D equation are outlined in Eq. (1) and Eq. (2), respectively. These equations are also followed by the solenoidal condition $\nabla \cdot \mathbf{u} = 0$ stemming from incompressible mass conservation.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nabla \cdot \boldsymbol{\tau} - \mathbf{g} \quad (1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -\frac{1}{\rho} \frac{\partial P}{\partial x} - \frac{f}{D_e} \frac{u|u|}{2} - g \quad (2)$$

The domain overlapping coupling method uses a CFD-informed friction factor to ensure that given the same inlet velocity, the pressure gradient in SAM's overlapping component matches the NekRS solution. In order for the pressure gradients in Eq. (1) and Eq. (2) to match, the sum of SAM's convection, friction,

and gravity terms should match the sum of NekRS's convection, viscous, Reynolds-stress, and gravity terms. To do so, SAM's convection and gravity terms in Eq. (2) are first removed to allow NekRS to model these effects. Then, SAM's friction term is equated to NekRS's inertial (including gravity) and friction terms, and the friction factor is solved for as the following CFD-informed friction factor:

$$f_{CFD} = -\frac{2D_e}{\rho u |u|} \left[\nabla P + \rho \frac{\partial \mathbf{u}}{\partial t} \right]_{CFD}, \quad (3)$$

where $[\dots]_{CFD}$ contains information from the NekRS CFD solution. The friction-factor in Eq. (3) depends on SAM's velocity u and is thus implemented as an implicit term within SAM's internal solver. Also, the overlapped portion of SAM solves a modified version of the momentum equation since NekRS is already modeling the convective and gravitational terms. This is especially important when buoyancy effects are present, such as during coupled modeling of natural circulation loops.

To use f_{CFD} from Eq. (3), a single value is computed for the entire overlapped STH component. The bracketed term with 3D CFD information is derived as a volume-averaged quantity. First, the 3D CFD data is dotted into a 1D flow direction, which is labeled by the flow path vector field $\hat{\mathbf{n}}_{fp}$. Then it is volume averaged using $\langle \cdot \rangle = \frac{1}{V} \iiint_V (\cdot) dV$. Doing so for the pressure gradient and inertial term in Eq. (3), the full form of f_{CFD} becomes:

$$f_{CFD} = \frac{2D_e}{\rho u |u|} \left[\frac{\Delta P_{CFD}^n}{L} - \frac{\rho}{\Delta t} \left(\frac{1}{V} \iiint (\mathbf{u}^{n+1} \cdot \hat{\mathbf{n}}_{fp}) dV - \frac{1}{V} \iiint (\mathbf{u}^n \cdot \hat{\mathbf{n}}_{fp}) dV \right) \right], \quad (4)$$

where n corresponds to the n -th time step, L is the 1D component length and V is the volume of the component. The inertial term in Eq. 4 is time-averaged using a first-order scheme. Eq. 4 is only applicable to simple geometries where $\hat{\mathbf{n}}_{fp}$ and L are easily defined, and the extension of Eq. 4 to more sophisticated geometries requires further description.

3 COUPLING VERIFICATION

3.1 Open-Loop Configurations

The present SAM-NekRS coupling is first verified against canonical problems containing open-loop configurations. These problems are split into two categories: momentum coupling tests and energy coupling tests. Unless otherwise specified, the working fluid is water at atmospheric conditions with fluid properties obtained from NIST Linstrom and W.G. Mallard (2022). Furthermore, all round pipes have a nominal diameter of 0.1 meters.

3.1.1 Momentum coupling tests

To begin, the SAM-NekRS coupling is verified using canonical problems with isothermal flow conditions. A series of steady and transient test cases are carried out using SAM standalone, NekRS standalone, SAM-NekRS domain decomposition coupling, and SAM-NekRS domain overlapping coupling. All turbulent NekRS simulations use the wall-resolved RANS $k - \tau$ model (Benton et al. (1996); Shaver et al. (2020)). For means of comparison, the NekRS inlet velocity field is always set as a flat profile such that NekRS produces a larger predicted pressure drop than a SAM model, due to 3D entrance effects captured by the NekRS model. Otherwise, if NekRS employed a fully-developed profile, the NekRS component's pressure drop would just match the SAM model's, and this is not useful for testing purposes. Therefore, the following tests employ a flat inlet velocity such that the NekRS solution is different than the SAM solution for these tests.

Straight pipe with one interface

First, the simple straight pipe shown in Fig. 3 is employed and contains a single coupling interface. First a steady state comparison is made by varying the inlet Reynolds number between 30,000 and 50,000 and comparing the total pressure drop predicted over the entire straight pipe.

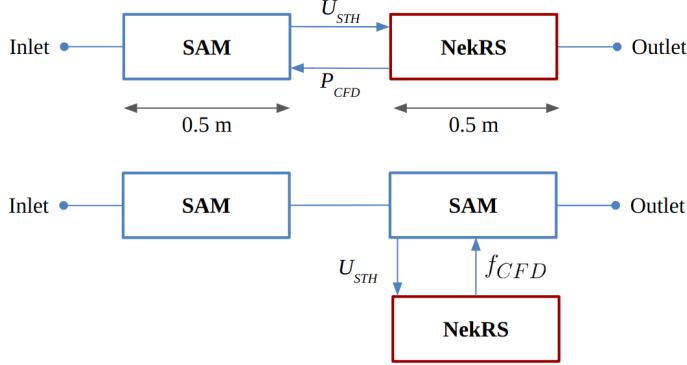


Figure 3: Straight pipe with one interface for domain decomposition (top) and domain overlapping (bottom).

The steady state results from each model are shown in Fig. 4, where the SAM standalone's pressure drop is always lower than the NekRS standalone or coupled SAM-NekRS results. As stated previously, this difference in the pressure drop is due to 3D entrance effects in the NekRS domain due to imposing a flat inlet velocity profile, and these effects are not modeled in SAM. The main takeaways from Fig. 4 are 1) the SAM-NekRS coupled results using the domain decomposition and domain overlapping methods are consistent and 2) the coupled results are between the SAM standalone and NekRS standalone results. The latter is the result of the coupled simulations' domain being split in half between SAM and NekRS models. These outcomes provide confidence in the implementation of both coupling methods for steady state isothermal simulations.

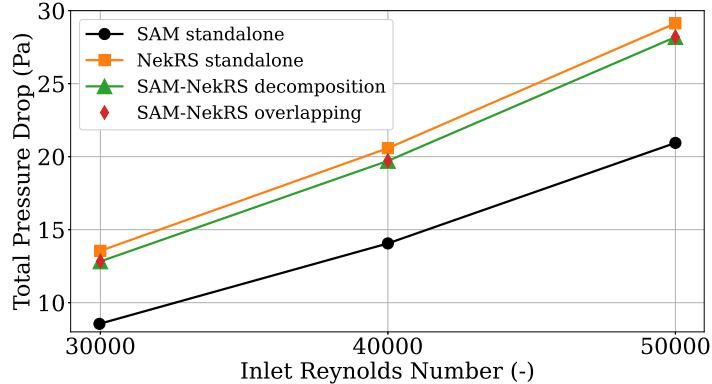


Figure 4: Steady state results for the straight pipe with one interface.

Now for a transient test case, the same setup in Fig. 3 is used while varying the inlet Reynolds number linearly over time following Eq. 5. Doing so adds inertial effects to the pressure drop being predicted by each standalone and coupled simulation. The transient pressure drop results are shown in Fig. 5. As was shown in the steady state test, the SAM standalone pressure drop is always lower than the NekRS standalone as well as the coupled SAM-NekRS results. Furthermore, the SAM-NekRS domain decomposition and domain overlapping results match during this transient as well. This gives confidence in the domain overlapping method's implementation of the inertial term when forming the CFD-informed friction factor.

$$Re_{\text{inlet}}(t) = \begin{cases} 30000, & t \leq 1.0 \text{ s}, \\ 30000 + 2000(t - 1.0), & 1.0 \text{ s} < t < 6.0 \text{ s}, \\ 40000, & t \geq 6.0 \text{ s}, \end{cases} \quad (5)$$

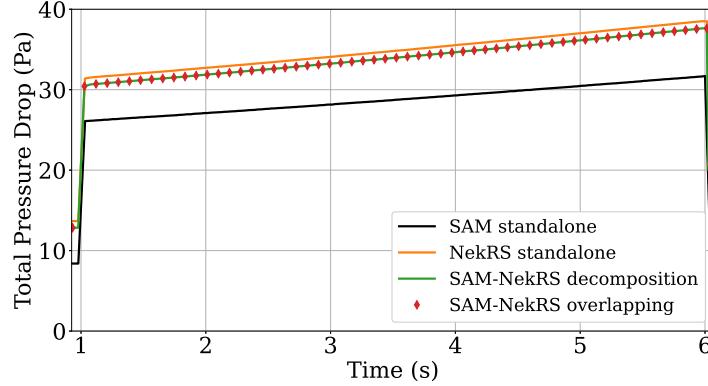


Figure 5: Transient results for the straight pipe with one interface.

To further test the coupling using this simple setup, another transient is utilized. Here, the flow is initially stagnant and the pipe is exposed to a sudden pressure drop over the entire pipe's domain such that flow develops within over time. In this case, flow is driven by the specified pressure gradient. Such a case is ill-posed for a CFD standalone simulation unless the inlet distributions for the pressure and RANS scalar fields. Instead, only SAM standalone and SAM-NekRS coupled simulations are utilized for this transient.

The sudden pressure drop is set such that the SAM standalone simulation's final steady state Reynolds number will be approximately 30,000. Results for this flow development transient are shown in Fig. 6 and contain two main takeaways. First, due to the NekRS portion's entrance effect modeling, the final coupled steady state Reynolds number is lower than the SAM standalone result, and as expected when using a flat inlet velocity for the NekRS domains. Second, the coupling domain decomposition and domain overlapping results are consistent. This shows that the domain overlapping coupling method is able to properly simulate flow that is driven by a pressure gradient, and such is the case for the pump-driven loop in Sec. 4.1.

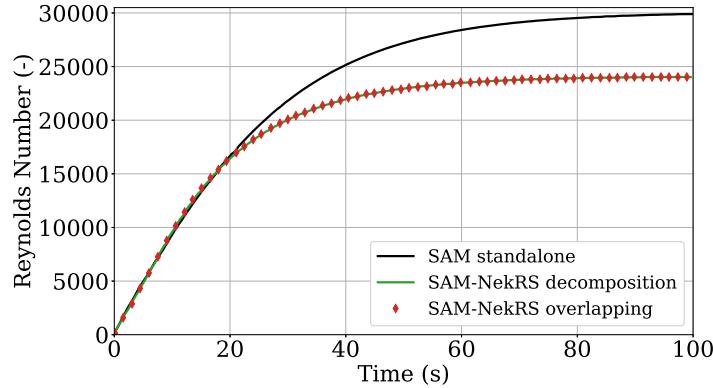


Figure 6: Transient flow development results for the straight pipe with one interface.

Straight pipe with two interfaces

Now, the straight pipe shown in Fig. 7 is employed and contains two coupling interfaces. Similar to the cases with one interface, a steady state comparison is made by varying the inlet Reynolds number between 30,000 and 50,000 and comparing the total pressure drop predicted over the entire pipe.

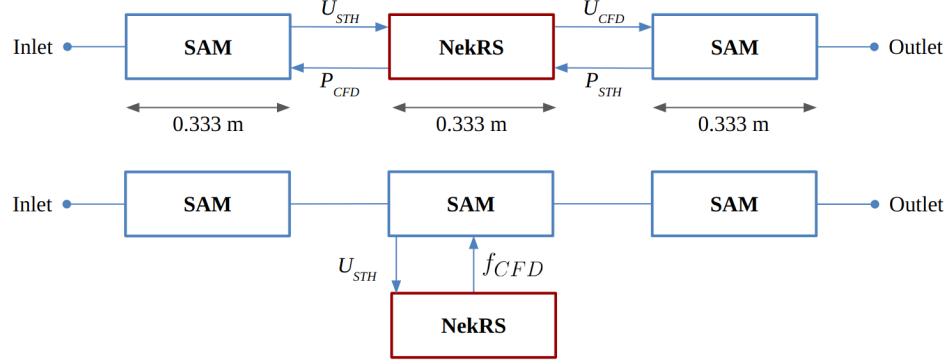


Figure 7: Straight pipe with two interfaces for domain decomposition (top) and domain overlapping (bottom).

The steady state results are shown in Fig. 8, where again the SAM standalone's pressure drop is always lower than the NekRS standalone or coupled SAM-NekRS results. Again, the coupled SAM-NekRS results are consistent, showing the methods are properly implemented when using two coupling interfaces.

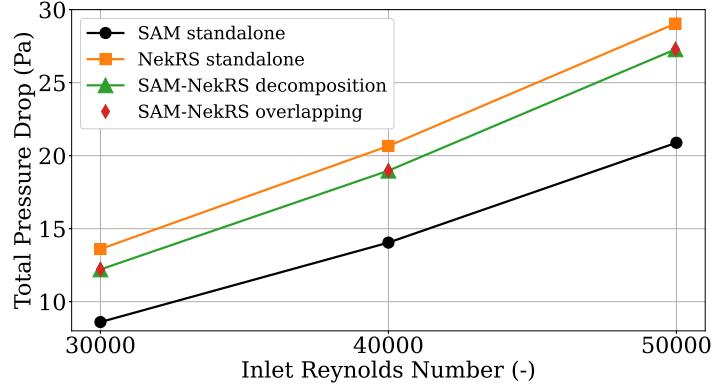


Figure 8: Steady state results for the straight pipe with two interfaces.

For a transient test case with the same setup in Fig. 7, the inlet Reynolds number increased linearly over time following Eq. 5 again. The transient pressure drop results are shown in Fig. 9. The same consistent results are observed as in the transient with one coupling interface. However, more interesting behavior is observed due to using the explicit temporal coupling method, where the SAM-NekRS domain decomposition's pressure drop in Fig. 9 lags behind the SAM-NekRS domain overlapping's pressure drop at the start and end of the transient. This is because when using the explicit coupling method implemented in Fig. 2 with multiple coupling interfaces, the downstream SAM component receives the updated U_{CFD} one timestep behind the updated U_{STH} value. On the other hand, the SAM-NekRS domain overlapping method does not have this issue because SAM doesn't receive any velocity information from the NekRS domain. The CFD-informed friction factor is able to handle this and as a result, the domain overlapping method doesn't lag behind at the start and end of the transient.

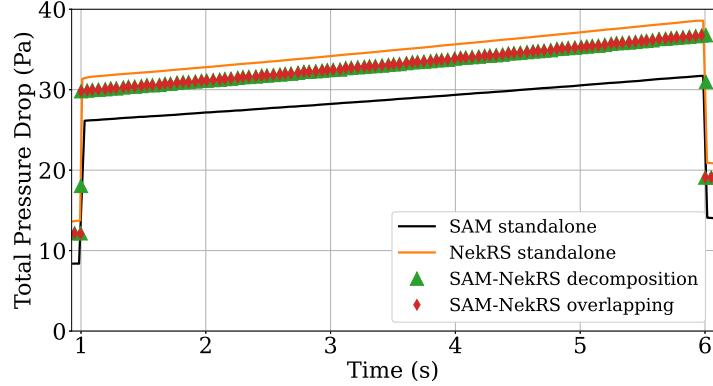


Figure 9: Transient results for the straight pipe with two interfaces.

Bent pipe for domain overlapping coupling

The bent pipe in Fig. 10 is now used to specifically test the domain overlapping coupling implementation on a non-straight pipe. Here, the pipe diameter and bend radii are set to 0.06 m and 0.07 m, respectively. First a steady state test is carried out while varying the inlet Reynolds number from 10,000 to 30,000. Then, a similar transient test is carried out by varying the inlet Reynolds number over time.

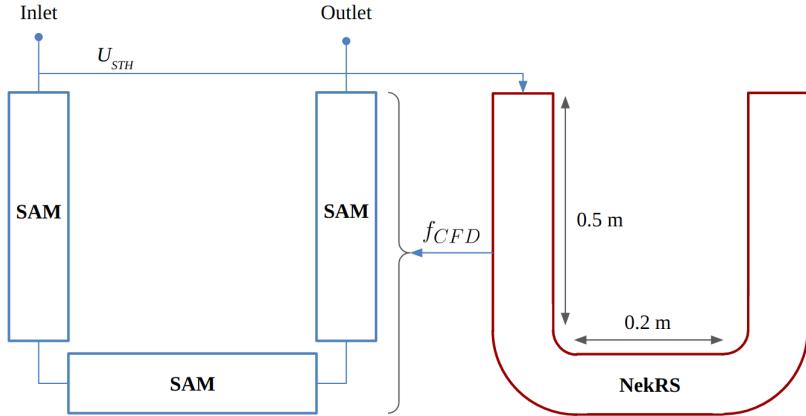


Figure 10: Full bent pipe for the domain overlapping coupling.

The steady state results are shown in Fig. 11, where the only coupling utilized is the SAM-NekRS domain overlapping coupling. In Fig. 11, the SAM-NekRS coupled results match the NekRS standalone results because the full bent pipe is being coupled. These values are consistent and, as previously seen, again have a higher predicted pressure drop than the SAM standalone simulation due to NekRS modeling entrance effects.

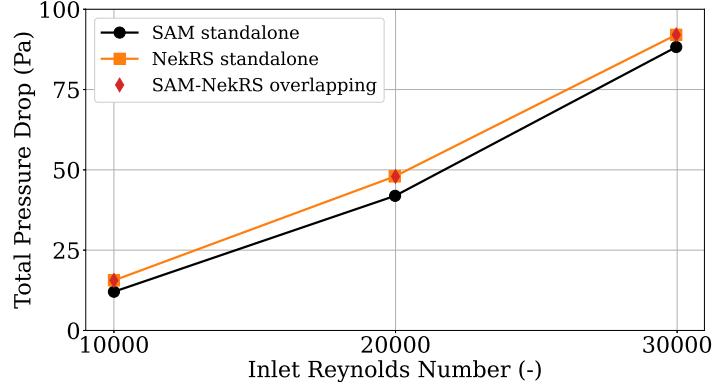


Figure 11: Steady state results for the bent pipe.

For a transient test case with the same setup, the inlet Reynolds number increased linearly over time following Eq. 6 again. The transient pressure drop results are shown in Fig. 12. Here, the coupled SAM-NekRS domain overlapping results match the NekRS standalone results over the entire transient. Therefore, the domain overlapping method's implementations of the CFD-informed friction factor is correct for this non-straight pipe. This gives further confidence in the present SAM-NekRS coupled implementation.

$$Re_{\text{inlet}}(t) = \begin{cases} 10000, & t \leq 1.0 \text{ s}, \\ 10000 + 2000(t - 1.0), & 1.0 \text{ s} < t < 6.0 \text{ s}, \\ 20000, & t \geq 6.0 \text{ s}, \end{cases} \quad (6)$$

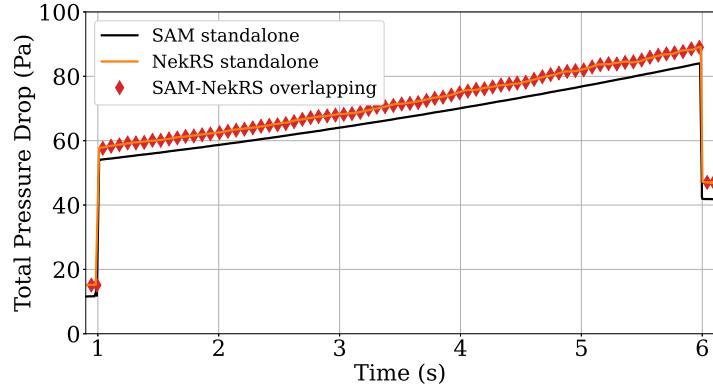


Figure 12: Transient results for the bent pipe.

3.1.2 Energy coupling tests

Next, the SAM-NekRS coupling is verified in terms of its energy coupling, where each canonical problem now has nonisothermal flow conditions. These tests are mainly to test the hybrid domain overlapping coupling method's ability to provide consistent results with the domain decomposition method in terms of pressure drop in the presence of buoyancy effects as well as transient temperature transport.

Steady full pipe against gravity

The first test case contains the full straight pipe in Fig. 13 that contains a volumetric heating source with flow moving against gravity. This test's flow contains buoyancy effects due to temperature change within

the pipe that must be accounted for in the SAM-NekRS coupling. To simplify this test, the inlet Reynolds number is laminar such that NekRS does not require any RANS turbulence modeling. Furthermore, the fluid properties and heat source are set using a ρ of 1.0 kg/m^3 , μ of $1\text{-}3 \text{ Pa-s}$, C_p of 1.0 J/(kg-K) , k of $1\text{-}3 \text{ W/m-K}$, β of 0.1 1/K and q''' of 1.0 W/m^3 . All NekRS models utilize the Boussinesq approximation with the provided thermal expansion coefficient β in its momentum equation to account for buoyancy effects due to temperature variation. Furthermore, the velocity U_{STH} passed to NekRS is employed using a fully-developed, parabolic profile for pipe flow, but the temperature T_{STH} is passed as a flat inlet temperature profile. As a result, any difference in predicted pressure drop between SAM and NekRS is due only to buoyancy effects.

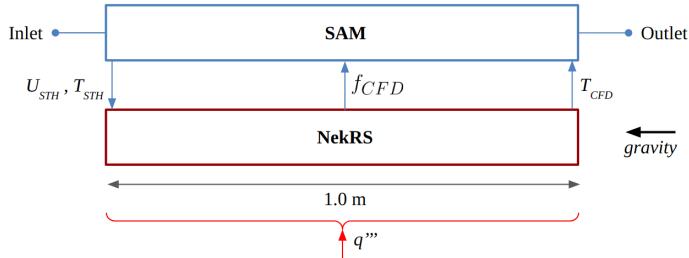


Figure 13: Heated straight pipe test case for hybrid domain overlapping coupling.

The steady state results are shown in Fig. 14, where the steady state pressure drop is compared between models. Here, the NekRS standalone model's steady state pressure drop is lower than the SAM standalone results due to NekRS employing a flat inlet temperature profile. Similar to the entrance effects shown previously when using a flat inlet velocity profile, using a flat temperature profile produces entrance effects due to buoyancy. Also, the coupled SAM-NekRS results in Fig. 14 match the NekRS standalone results. This shows that the hybrid domain overlapping coupling's CFD-informed friction factor properly takes buoyancy effects into account for the steady state, nonisothermal flow.

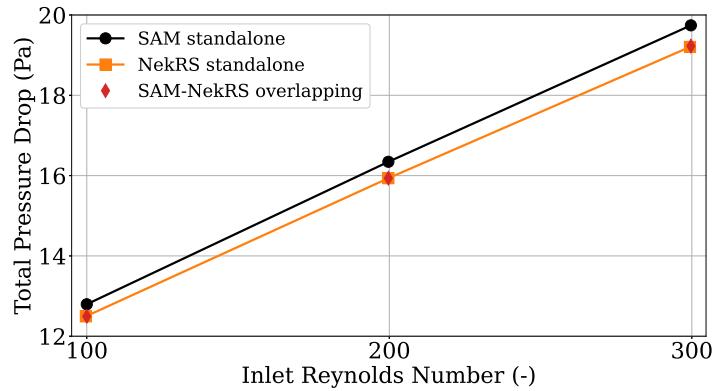


Figure 14: Steady state results for the heated straight pipe.

Transient partial straight pipe

For a transient test, the partial straight pipe in Fig. 15 is employed in order to ensure the domain decomposition and hybrid domain overlapping methods' results are consistent. Here, the inlet Reynolds number is held constant and the initial temperature within the domain is 0.0 K . The inlet temperature over time follows the step function in Eq. 7. This inlet temperature is then transported over time from the pipe's inlet to its outlet.

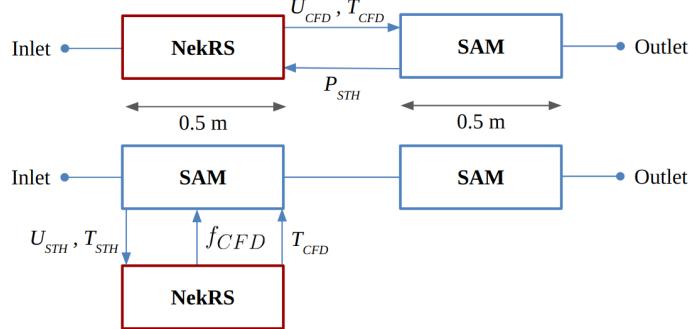


Figure 15: Straight pipe with one interface for domain decomposition (top) and domain overlapping (bottom).

$$T_{\text{inlet}}(t) = \begin{cases} 0.0, & t \leq 0 \text{ s}, \\ 1.0, & t > 0 \text{ s}, \end{cases} \quad (7)$$

The transient results are shown in Fig. 16, where the outlet temperature is monitored overtime and compared. The transported temperature reaches the outlet in the NekRS standalone model before the SAM standalone model due to the 3D flow field within the NekRS simulation. Explicitly, the NekRS model contains a parabolic velocity profile which contains a maximum velocity corresponding to $U_{\max} = 2 \cdot U_{\text{STH}}$, whereas the SAM model only contains a single velocity U_{STH} . Since the temperature is transported mainly via advection effects, the NekRS standalone outlet temperature rises approximately twice as quickly as the SAM standalone temperature.

Furthermore, in theory the SAM standalone plot in Fig. 16 should be a perfect step function increasing from 0 to 1 K exactly at time of 1.0 seconds. However, the implicit euler time stepping scheme is utilized in the SAM model, which adds a small amount of temporal diffusion and slightly smooths the step function. Also, the both coupling SAM-NekRS results in Fig. 16 are between the standalone simulations, which is consistent as the coupling models' spatial domain is split in half between NekRs and SAM domain. Lastly, the domain overlapping transient results match the domain decomposition results, showing that the methods are consistent when performing energy coupling.

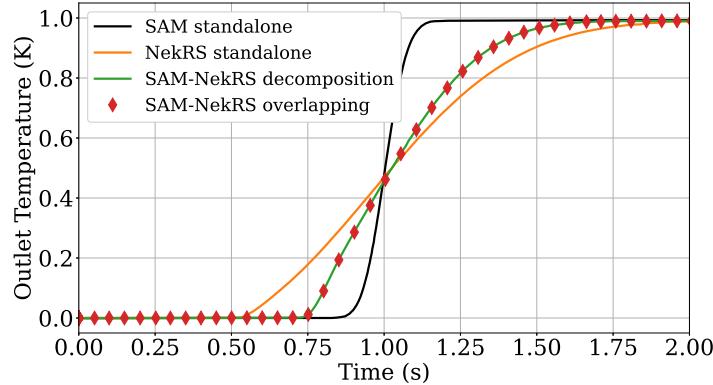


Figure 16: Transient results for the energy coupling test.

4 STABILITY COMPARISONS

The present SAM-NekRS coupling's domain decomposition and domain overlapping implementations are now compared in terms of stability. This is carried out using two closed-loop configurations: a pump-driven loop and a natural circulation loop.

4.1 Pump-driven Closed Loop Configuration

The pump-driven loop in Fig. 17 is implemented with isothermal flow conditions. The set pump head produces a pressure gradient in the loop, resulting in a forced flow rate and a tightly-coupled system between the pressure and velocity field. The working fluid is water at atmospheric conditions with fluid properties obtained from NIST Linstrom and W.G. Mallard (2022).

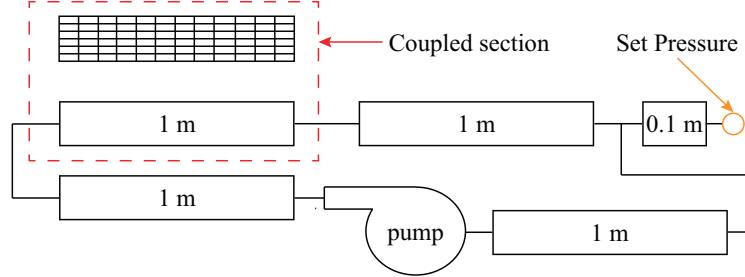


Figure 17: Pump-driven loop's setup with lengths in meters.

All straight components in Fig. 17 are round pipes with a diameter of 0.1 meters, and NekRS is used to model the coupled section of the loop. The pump head is set to 25 Pa such that the steady-state Reynolds number predicted by a SAM standalone simulation is approximately 25,000. NekRS RANS modeling is employed using the $k - \tau$ model Benton et al. (1996); Shaver et al. (2020). This model is implemented in NekRS similarly to what is described in Kok (2000).

To compare the stability of domain decomposition and domain overlapping coupling methods, the change in loop velocity U_{STH} between time steps n and $n - 1$ is considered relative to the steady loop velocity U_{steady} . The relative stability criteria is explicitly written in Eq. (8). Once this criteria is met, the coupled SAM-NekRS simulation is deemed numerically stable.

$$\left| \left(\frac{dU_{STH}}{dt} \right)^n - \left(\frac{dU_{STH}}{dt} \right)^{n-1} \right| < 1e-3 \cdot U_{steady}. \quad (8)$$

To obtain a good initial guess for the loop, a coupled simulation is first preformed to obtain a “correct” steady flow rate in the loop. This steady flow rate is then used to set the initial velocity field within the NekRS domain, and the initial velocity in the SAM domain is deviated from this steady value to test the coupling’s ability to return to a stable solution. For means of comparison, the total number of NekRS pressure solver iterations required to reach the stability criteria in Eq. (8) is compared while deviating the initial SAM domain’s velocity. The total NekRS pressure solver iterations is used as the figure of merit because, as Fischer et al. (2021b) showed, the majority of NekRS’s computational effort is used for its pressure solver.

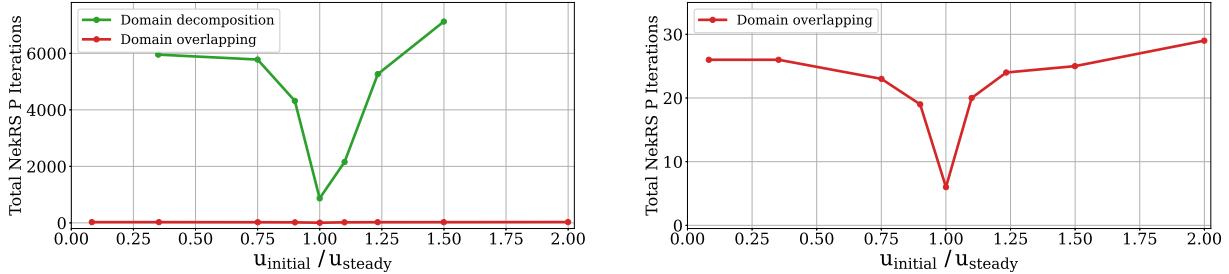


Figure 18: Pump-driven loop stability comparison between both methods (left) with a zoomed-in view (right).

The stability comparison between coupling methods is shown in Fig. 18, where a lower number of NekRS

pressure iterations equates to a more efficient coupled simulation. In Fig. 18, the lowest number of pressure iterations is required when the initial loop velocity is the same as the steady state velocity ($u_{initial}/u_{steady} = 1$). Given the same initial velocity, the domain overlapping method requires far fewer NekRS pressure iterations to reach stability than the domain decomposition method. Also, the domain decomposition method is able to reach stability using an initial velocity 33-150% the steady velocity, whereas the domain overlapping can converge with an initial velocity 5-200% the steady velocity. Therefore, the domain overlapping method is able to handle a wider range of conditions than domain decomposition.

4.2 Natural Circulation Closed Loop Configuration

A natural circulation loop presents a highly-coupled system where pressure losses are balanced between buoyancy (temperature) and friction (velocity) effects. The single-phase, natural circulation loop from Grunloh (2016)'s STH/CFD coupling is used and is shown in Fig. 19. For a complete description, authors refer to Grunloh (2016), but a brief description is given here. The heat q imposed and removed in Fig. 19 is set such that the steady state Reynolds number predicted is approximately 100,000 and thus sufficiently turbulent to utilize NekRS's RANS $k-\tau$ model. Water properties from NIST Linstrom and W.G. Mallard (2022) at a reference pressure and temperature of 15 MPa and 550 K, respectively, are used for prescribing the fluid's density, viscosity, and thermal expansion coefficient.

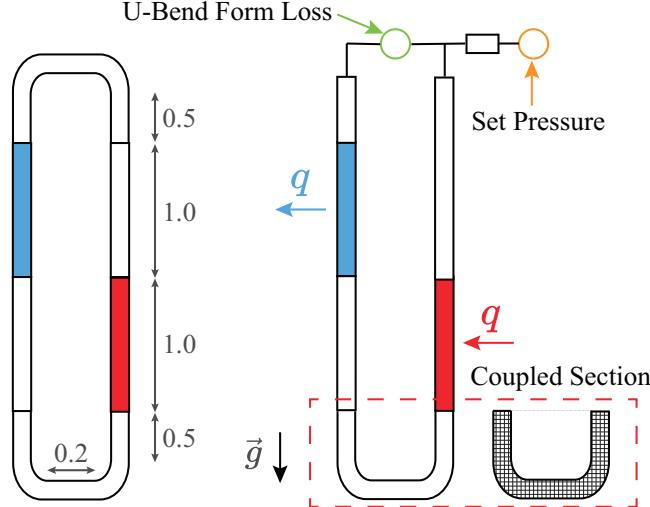


Figure 19: Natural circulation loop's setup with lengths in meters (left) and SAM-NekRS setup (right).

For the coupling setup in Fig. 19, NekRS models the bottom U-bend where 3D effects are most present, and SAM models the rest of the loop. The NekRS model's all-hex mesh is generated using Gmsh Geuzaine and Remacle (2009) while ensuring the $y+$ off the wall is below 1. Fig. 20 shows the computational mesh utilized and the velocity field predicted by NekRS. Boundary conditions set in the NekRS model are listed in Table 2. Furthermore, the NekRS RANS $k-\tau$ model is employed using the Boussinesq approximation in its momentum equation to account for buoyancy effects due to temperature variation. For the SAM model, to best account for friction effects in the top U-bend a U-bend form-loss coefficient from Idelchik (1986) is employed.

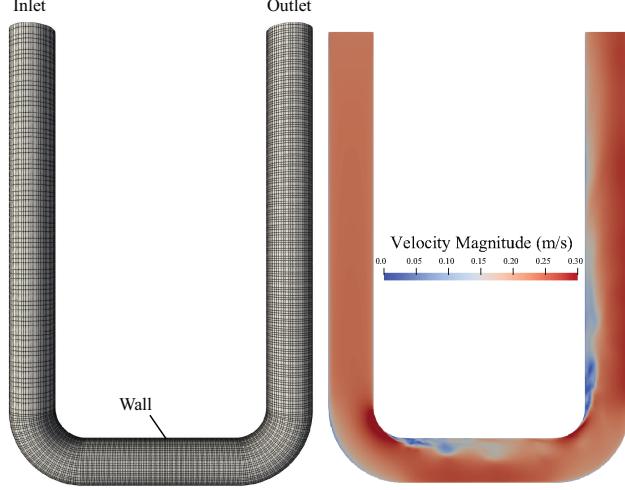


Figure 20: NekRS computational mesh with labeled boundaries (left) and NekRS RANS $k - \tau$ predicted velocity field (right).

Table 2: Boundary conditions set for the U-bend's NekRS model.

Boundary	Pressure	Velocity	RANS k and τ	Temperature (T)
Inlet	-	Developed w/ U_{STH}	Developed	Flat w/ T_{STH}
Outlet	0	$\nabla U = 0$	$\nabla k = \nabla \tau = 0$	$\nabla T = 0$
Wall	-	0	0	$\nabla T = 0$

To compare stability again, the change in loop velocity between time steps is considered relative to the steady loop velocity. The same relative criteria as the pump-driven loop is utilized, as written previously in Eq. (8). To obtain a good initial guess for the loop, a coupled simulation is performed to obtain a “correct” steady flow rate in the loop, which is considered the ideal initial conditions for the coupled simulation. The steady flow rate is then used to set the initial velocity field within the NekRS domain, and the initial velocity in the SAM domain is deviated from this steady value. The temperature field in the loop is initialized using a SAM checkpoint file.

For means of comparison, the total number of NekRS pressure solver iterations required to reach the stability criteria in Eq. (8) is compared while deviating the initial SAM domain’s velocity. The results for the domain overlapping coupling are shown in Fig. 21, where the method was able to reach the stability criteria while starting with an initial velocity deviated nearly 10% from the steady velocity.

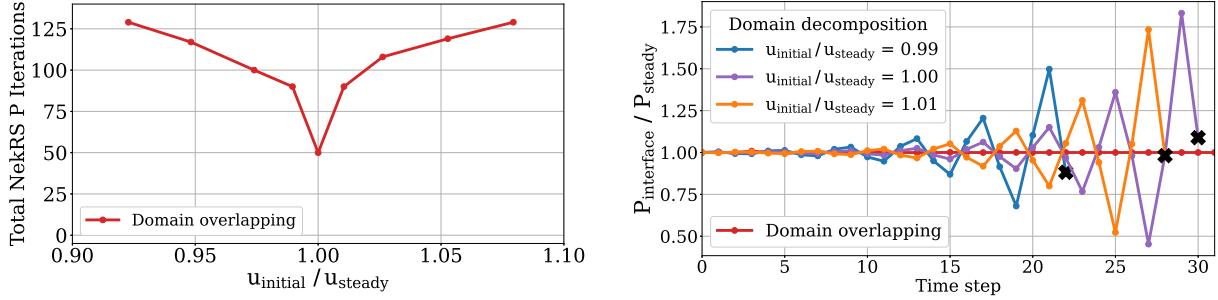


Figure 21: Natural circulation loop stability comparison between methods. Black X’s denote when the coupled domain decomposition simulation fails to converge.

The domain decomposition coupled simulation breaks at or before the 30th time step, which is due to the large pressure feedback effects in the loop. The normalized interface pressure at the inlet of the NekRS domain is shown in Fig. 21 for various initial flow conditions. Large pressure oscillations are seen and continue to propagate until the simulation fails to converge, whereas the domain overlapping method stays stable. This is due to the natural circulation loop's tight coupling between velocity, pressure and temperature fields, which domain decomposition with explicit coupling can not stabilize. As a whole then, Fig. 21 shows the domain overlapping method is again more stable than the domain decomposition method.

5 COUPLING VALIDATION

5.1 Double T-Junction Experiment

To validate the hybrid domain overlapping coupling implementation, the double T-junction experiment from Bertolotto et al. (2009) is employed. Comparisons are made between experiment data, a SAM standalone simulations, a SAM-NekRS coupled simulation, and Bertolotto et al. (2009).’s coupled TRACE-CFX simulation.

5.1.1 Double T-junction setup

For a complete description of the experiment, the authors direct readers to Bertolotto et al. (2009) and Bertolotto (2011), but a brief description is given here. The double T-junction’s setup is shown in Fig. 22, and it contains two loops joined by a double T-junction where one loop serves as the main line and one the recirculation line. The main line enforces a set flow rate into the double T-junction, and the recirculation line has a pump forcing flow back into the double T-junction. The system is operated at atmospheric conditions with tap water as the working fluid. A tracer (deionized water) is injected within the side loop, and its concentration is measured at three different locations using wire mesh sensors (WM). The flow rate is controlled in each loop to maintain the same Reynolds number of 23,750.

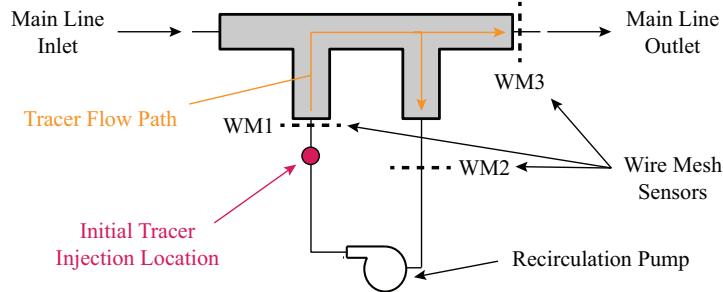


Figure 22: Schematic of the double T-junction experiment, adapted from Bertolotto et al. (2009).

5.1.2 NekRS model details

NekRS is used to model the double T-junction where 3D mixing effects are most present, and the NekRS model employs a unsteady-RANS simulation with its $k-\tau$ model along with an additional passive scalar for modeling the tracer’s transport. An all-hex mesh is generated using Gmsh Geuzaine and Remacle (2009) while ensuring the wall $y+$ is below 1, and Fig. 23 shows the NekRS mesh used and the NekRS predicted flow field with boundaries labeled. Boundary conditions set in the NekRS model are listed in Table 3. The NekRS model’s domain was extended at both outlets to ensure there’s no reversed flow at these boundaries. The term “Developed” in Table 3 refers to a fully-developed profile coming from a periodic simulation for the same operational Reynolds number as the experiment.

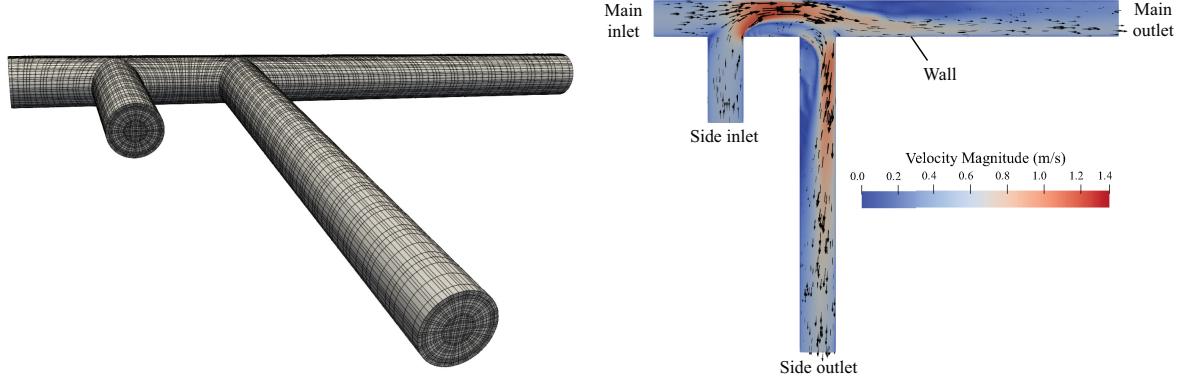


Figure 23: NekRS mesh (left) and predicted velocity field along the domain’s centerline, with boundaries labeled (right).

In Table 3’s NekRS boundary conditions, the relative pressure at the Side outlet is set to -365 Pa such that NekRS reproduces the 1:1 flow ratio used in the experiment. In Bertolotto et al. (2009)’s CFX simulation, the authors utilized a split-flow boundary condition. For the present NekRS model, setting the relative Side outlet pressure achieves the same desired result.

For the velocity field’s boundary conditions in Table 3, a developed profile from a periodic simulation is used for both inlets. However, the Side inlet’s velocity profile is scaled such that its average velocity matches the velocity U_{STH} coming from SAM. For the tracer’s Side inlet boundary condition, Bertolotto et al. (2009) found that the distribution injected into the CFD domain has little influence on tracer concentrations extracted at wire mesh sensor locations. Therefore, a flat inlet distribution is used, corresponding to the value S_{STH} from SAM, where S denotes that the tracer is a passive scalar.

Table 3: Boundary conditions set for the double T-junction’s NekRS model.

Boundary	Pressure [Pa]	Velocity	RANS k and τ	Tracer (S)
Main inlet	-	Developed	Developed	0
Side inlet	-	Developed w/ U_{STH}	Developed	Flat w/ S_{STH}
Main outlet	0	$\nabla U = 0$	$\nabla k = \nabla \tau = 0$	$\nabla S = 0$
Side outlet	-365	$\nabla U = 0$	$\nabla k = \nabla \tau = 0$	$\nabla S = 0$
Wall	-	0	0	$\nabla S = 0$

The initial tracer injection is simulated as a prescribed, time-dependent profile at NekRS’s Side inlet boundary, and the injection follows experimental wire mesh sensor 1 data. The tracer is modeled as a passive scalar with zero molecular diffusivity but includes turbulent diffusion using a turbulent Schmidt of 0.9, the same methodology used in Bertolotto et al. (2009).

5.1.3 SAM model details

While NekRS models the double T-junction, SAM is used to model the entire recirculation loop, including the portion of the double T-junction that closes the recirculation loop. To enforce the 1:1 flow ratio between the main and recirculation loop, Bertolotto et al. (2009) implemented a TRACE component to set the flow rate in the recirculation loop. To do the same in SAM, the “desired mass flow rate” parameter is set for SAM’s pump component, such that a 1:1 flow ratio is maintained. The SAM model employs a passive scalar to model the tracer’s transport with a diffusion coefficient of 7.6e-3 m²/s, which acts to model 3D mixing effects in the recirculation loop. Bertolotto (2011) used a similar value for TRACE’s modeling of the same system.

5.1.4 Coupled SAM-NekRS concentration profiles

For all present simulations, a time step size of 2.5e-4 seconds is used. The simulations were repeated using a NekRS polynomial order of 3, 4, and 5. Results using a polynomial order of 4 and 5 did not differ significantly, so to limit computational expenses an order of 4 is used. For all comparisons, experiment and previous TRACE-CFX coupled results are compared to SAM standalone and SAM-NekRS coupled simulation results, using the present hybrid domain overlapping method.

The cross-section averaged concentration over time for the first flow through of tracer is shown in Fig. 24. WM1's concentration profile in all simulations matches the experiment exactly because this is the initial tracer injection used for all simulations. SAM-NekRS coupled results for WM2 and WM3 match the experiment better than the SAM standalone simulation while producing similar results to previous TRACE-CFX coupling. The SAM standalone simulation under predicts tracer recirculating in the recirculation loop through WM2 and over predicts tracer leaving the system through WM3. This is due to the 3D mixing effects present within the double T-junction, which SAM doesn't take into account when using its 1D components.

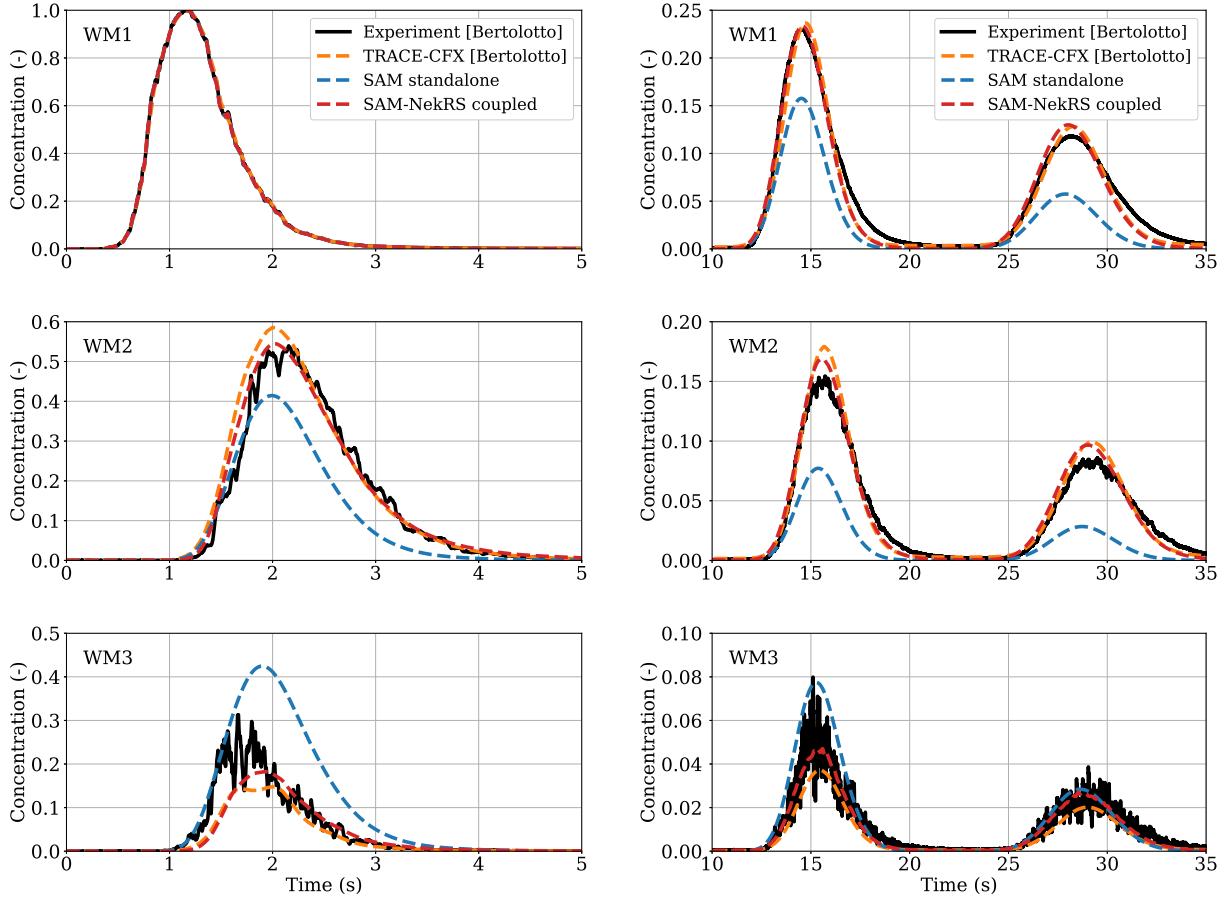


Figure 24: Cross-section averaged tracer concentration plots for the first tracer flow through (left) and second and third flow throughs (right).

After the first flow through, the tracer is recirculated through the side recirculation loop and re-enters the double T-Junction at WM1. The concentration over time for the second and third flow throughs are shown in Fig. 24. The SAM standalone simulation continues to under predict tracer recirculation through WM2 and over predict tracer leaving the system through WM3. SAM-NekRS coupled results are similar to previous TRACE-CFX results, but for WM2 and WM3, the present SAM-NekRS results match more closely with the experiment. This is likely due to a combination of effects stemming from differences in the codes

and models used between the SAM-NekRS and TRACE-CFX coupling. SAM utilizes higher-order numerical methods than TRACE, resulting in a low amount of numerical diffusion that would greatly affect passive scalar transport, both spatially and temporally. As for the CFD models, NekRS uses the RANS $k-\tau$ model whereas CFX used the RANS SST (Shear Stress Transport) model. Different RANS models are expected to predict different flow fields, so for the double T-junction the NekRS RANS $k-\tau$ model likely predicts a flow field more similar to the experiment compared to the CFX RANS SST model.

5.1.5 Coupled SAM-NekRS concentration integrals

Next, concentration integrals are used to compare the total concentration of tracer passing through each wire mesh sensor over time. The concentration integrals are shown in Fig. 25, where each integral increases three times, one for each flow through of tracer. In Fig. 25 it is even more apparent that the SAM standalone simulation under predicts tracer recirculation (through WM2) and over predicts tracer leaving the system (through WM3). Furthermore, SAM-NekRS coupled results match experiment more closely than the previous TRACE-CFX coupling, with the largest difference being in the WM3 concentration integral plot. This is likely due to differences in the predicted velocity field between NekRS's RANS $k-\tau$ model and CFX's RANS SST model used by Bertolotto et al. (2009).

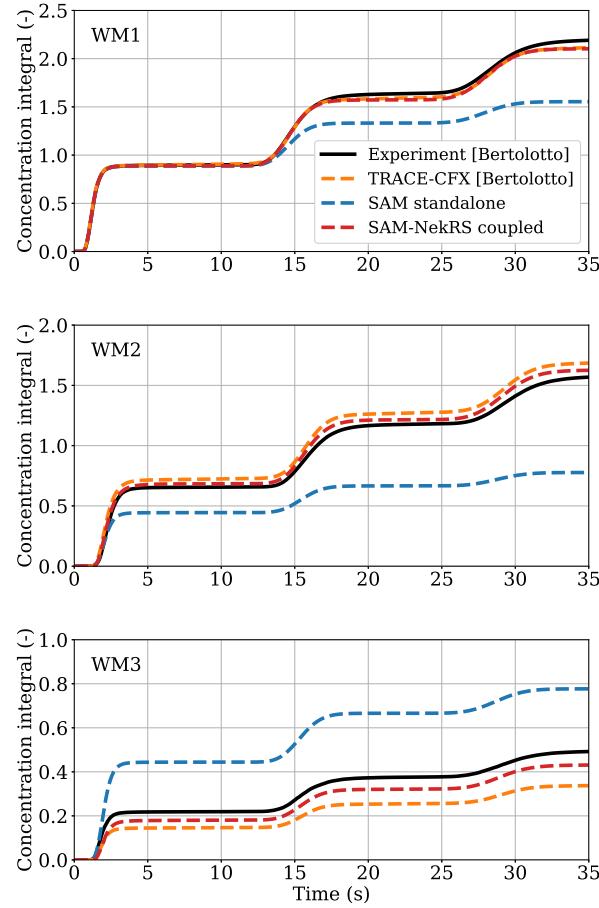


Figure 25: Concentration integrals through each wire mesh sensor location.

5.2 TALL-3D STH/CFD Benchmarking Experiment

With the coupling method validated on a small system with passive scalar coupling, the method requires further validation against experimental data of large systems containing mass, momentum and energy coupling.

The TALL-3D STH/CFD benchmark facility from Grishchenko et al. (2015) was chosen as it fulfills these requirements. TALL-3D is a Lead-Bismuth Eutectic (LBE) loop built for validating STH/CFD coupling efforts for advanced nuclear reactor systems. The facility's benchmark test TG03.S301.04 contains a transient from forced-circulation to natural circulation, emulating a protected loss of flow accident scenario in a pool-type, liquid metal cooled fast reactor. Therefore, validating the present coupling against this facility is of great interest. For a complete description of the facility, readers are directed to Grishchenko et al. (2015).

5.2.1 TALL-3D setup

The TALL-3D facility contains a primary LBE loop and secondary oil (Dowtherm RP) loop that is used for cooling the LBE loop. A schematic of the facility's primary loop is shown in Fig. 26. The primary loop contains three vertical legs, a main heater (MH) leg, test section (TS) leg, and heat exchanger (HX) leg. The HX leg contains an electro-magnetic pump that is used to obtain the initial forced circulation flow condition in the loop. Once the pump is shut off/ran down, the loop undergoes a transient until a final natural circulation flow condition is reached within the primary loop.

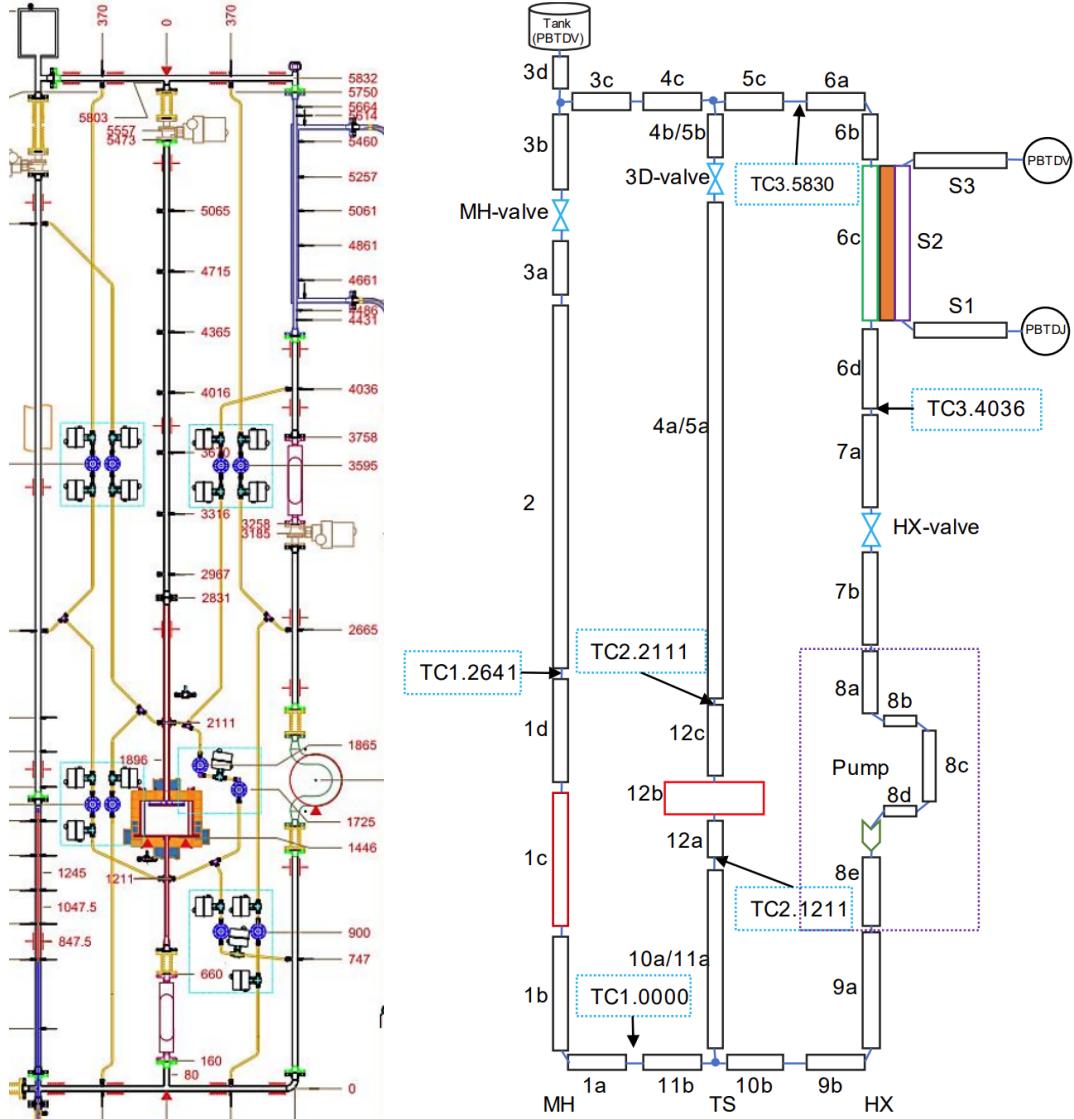


Figure 26: A schematic of the TALL-3D facility (left) and its SAM standalone model (right), from Kudinov and Grishchenko (2019a) and De Kock et al. (2022) respectively.

5.2.2 SAM model details

The SAM standalone model of the TALL-3D facility is shown in Fig. 26. All experimental conditions are taken from the TG03.S301.04 transient as provided by Kudinov and Grishchenko (2019b). Such conditions include the main heater's power (for component 1c), the test section heater's power (for component 12b), the additional heating to the LBE fluid when the pump is running initially (for component 8c), and the flow rate and inlet oil temperature on the secondary side of the heat exchanger (component PBTDJ connected to S1). All valves are modeled as components with form loss coefficients, which were calibrated to the system's final natural circulation state.

When adapting the SAM standalone model for domain-overlapping coupling, the only portion of Fig. 26 that changes is the test section's components of 12a, 12b and 12c. This portion of the TS leg is changed to a single PBOneDFluidComponent in SAM, such that the domain overlapping coupling can be employed properly.

5.2.3 STAR-CCM+ model details and validation

STAR-CCM+ is used to model the large enclosure test section instead of NekRS here. The main reason for this change is that STAR-CCM+ can model 2D axisymmetric flow whereas NekRS cannot and would instead require a 1/4 symmetric model. As a result, STAR-CCM+ is utilized to greatly reduce computational costs. The final, converged STAR-CCM+ model is shown in Fig. 27 with truncated inlet and outlet regions for brevity. The mesh shown partially in Fig. 27 contains a total of 105,000 quadrilateral elements, and the LBE fluid elements have a high least square quality of 0.996.

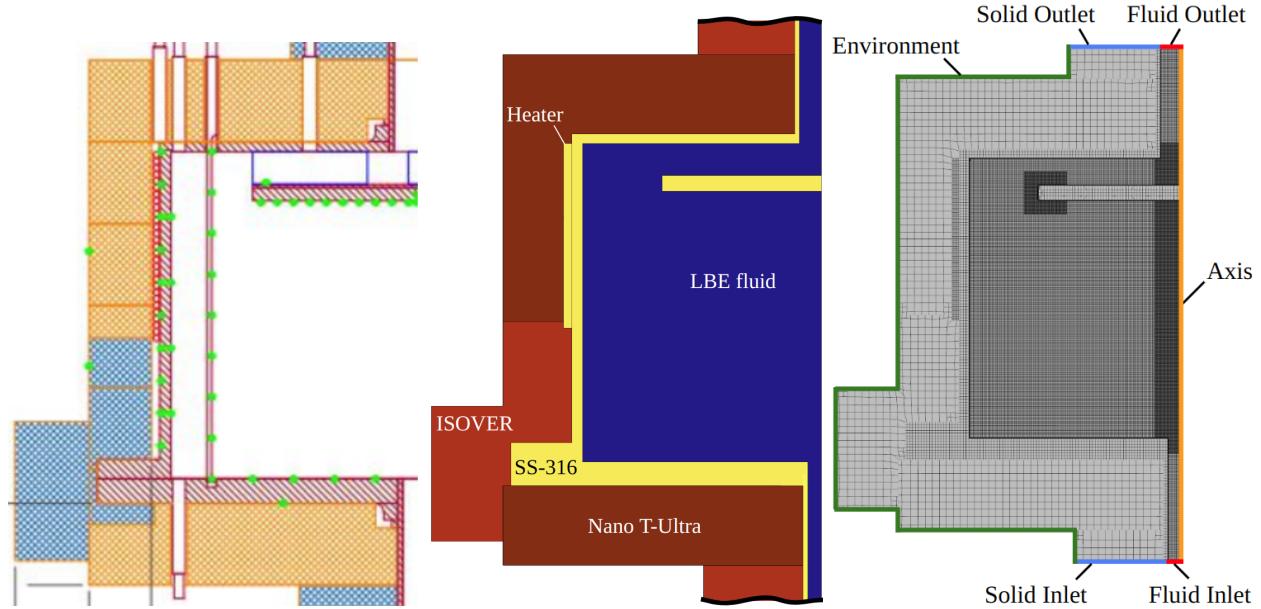


Figure 27: Half cross section of TALL-3D test section (left) STAR-CCM+ 2D, axisymmetric model of the test section with different fluid and solid regions (middle), and computational mesh (right) with external boundaries annotated.

The following relevant physics models were utilized for the LBE fluid region of the STAR-CCM+ model:

- Axisymmetric
- Segregated Flow
- Segregated Fluid Temperature
- Reynolds-Averaged Navier Stokes

- SST (Menter) K-Omega
- All $y+$ Wall Treatment
- Gravity
- Steady (used in STAR-CCM+ model validation) or Implicit Unsteady (used in coupled transient)
- User Defined EOS

The model implements LBE fluid properties from OECD (2007), Nano T-Ultra and ISOVER insulation solid properties from Kudinov and Grishchenko (2019c), and SS-316 solid properties from STAR-CCM+'s material library. Table 4 contains the boundary conditions used for each boundary in Fig. 27 and what coupled quantity is passed between SAM and STAR-CCM+. RANS turbulence modeling scalars are prescribed using a turbulence intensity and turbulent viscosity ratio of 1% and 10, respectively. The Environment boundary is set with ambient temperature of 29 °C and a heat transfer coefficient of 3 W/m²°C. During reversed flow, the Fluid Outlet boundary acts as an inlet and therefore requires a specified temperature T_{out} coming from SAM.

Table 4: External boundary conditions set for the STAR-CCM+ test section model.

ID	Type	Coupled Quantity
Fluid Inlet	Mass Flow Inlet	\dot{m}_{in}, T_{in}
Fluid Outlet	Pressure Outlet	T_{out}
Solid Inlet	Adiabatic	-
Solid Outlet	Adiabatic	-
Environment	Convection	-
Axis	Axis	-

To validate the STAR-CCM+ model of the test section, two standalone simulations are ran following the initial forced circulation and final natural circulation flow conditions of TG03.S301.04. Each steady state's conditions are shown in Table 5, where the inlet flow rate and temperature are set, and the model's predicted outlet temperature is compared to the experiment. Table 5 shows that the STAR-CCM+ model can predict the overall heat balance of the test section well for both flow conditions.

Table 5: Validation steady states for STAR-CCM+ modeling of TG03.S301.04

Parameter	Measurement Uncertainty	Forced Circulation		Natural Circulation	
		Experiment	Model	Experiment	Model
Inlet Flow Rate (kg/s)	3%	1.3	1.3	0.3	0.3
Inlet Temperature (°C)	2.2 °C	235	235	202	202
Outlet Temperature (°C)	2.2 °C	254	255.6	301	297.2

Fig. 28 and Fig. 29 show the STAR-CCM+ model's velocity and temperature fields during the forced circulation and natural circulation states, respectively. STAR-CCM+'s RANS $k-\omega$ SST model predicts the expected flow and temperature fields during both states. Fig. 28 shows jet impingement at the inner disc while Fig. 29 shows jet suppression with a large amount of thermal stratification within the test section.

Furthermore, Fig. 29 shows four line probes where the model's temperature field is now validated against experimental TC data, as shown in Fig. 30. The STAR-CCM+ model's prediction of thermal stratification is in excellent agreement with experimental data, although there is slight overestimation of temperature within the Bottom Plate's temperature plot. However, this difference is within just a few degrees of TC uncertainty bars and is therefore negligible. With the STAR-CCM+ model validated against the initial and final steady states, the model is used for the coupled transient simulation.

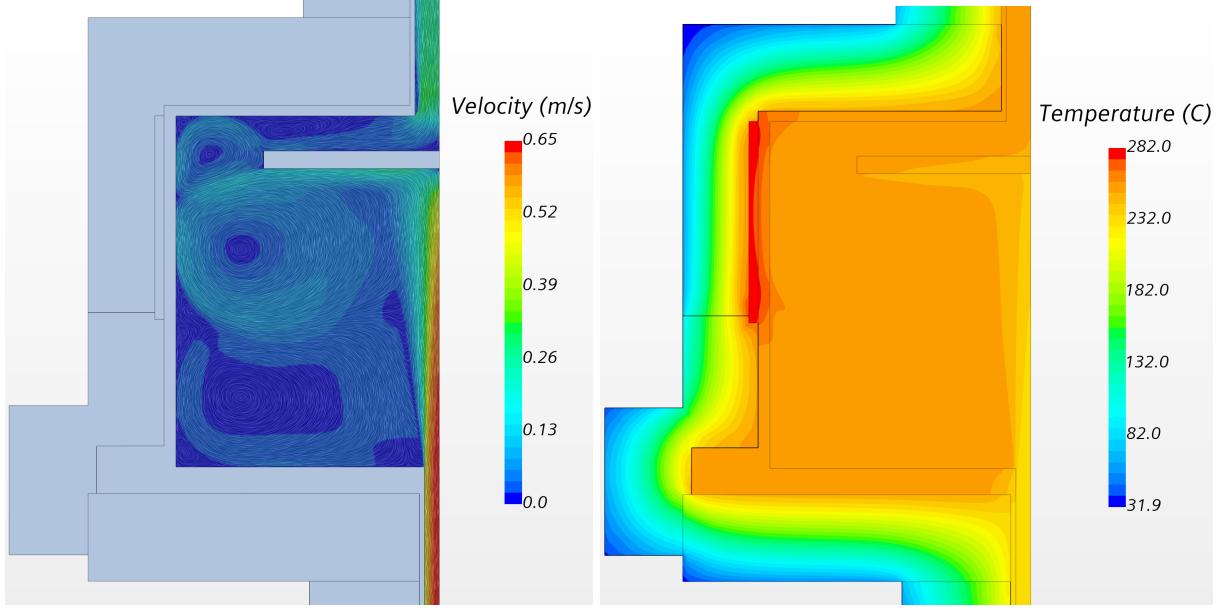


Figure 28: STAR-CCM+ model's initial forced circulation state velocity field (left) and temperature field (right) for TG03.S301.04.

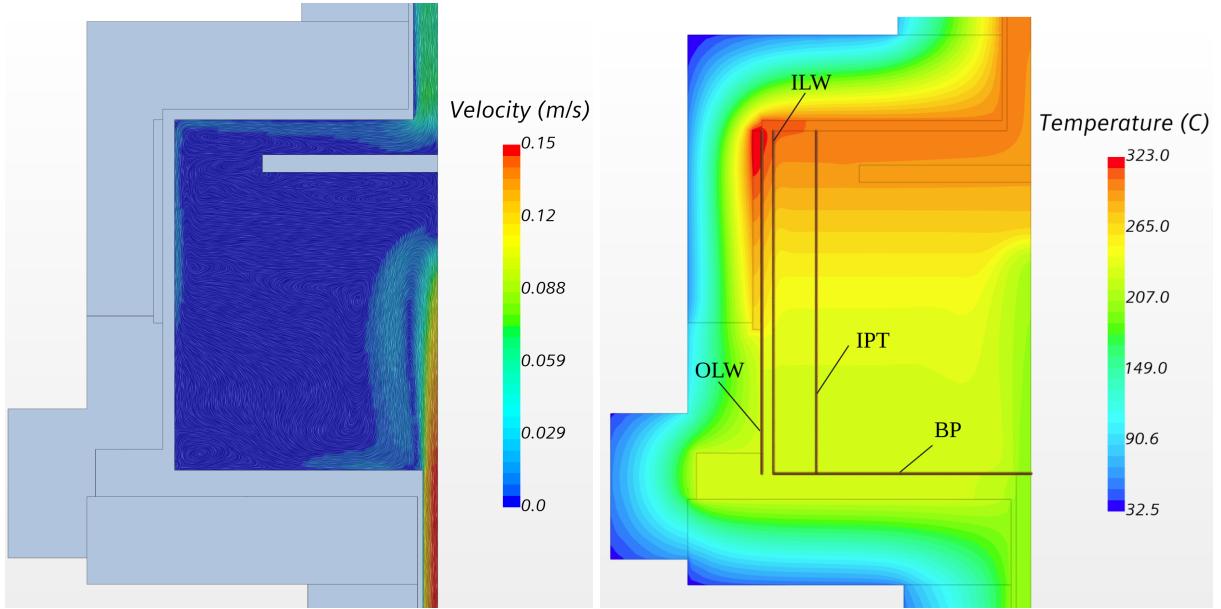


Figure 29: STAR-CCM+ model's final natural circulation state velocity field (left) and temperature field (right), annotated with line probes for comparing against experiment data for TG03.S301.04.

5.2.4 SAM-STAR-CCM+ coupling of the TG03.S301.04 transient

For the TG03.S301.04 transient, the SAM standalone simulation uses a variable timestep size between 0.1 and 5.0 seconds. The SAM-STAR-CCM+ coupled simulation uses a smaller variable timestep size between 0.01 and 0.1 seconds, with 10 inner iterations to ensure the all residuals drop well below 1E-4 between timesteps. Kudinov and Grishchenko (2019b) provide the following criteria for the successful modeling of the TG03.S301.04 transient, where a coupled STH/CFD model is expected to properly predict:

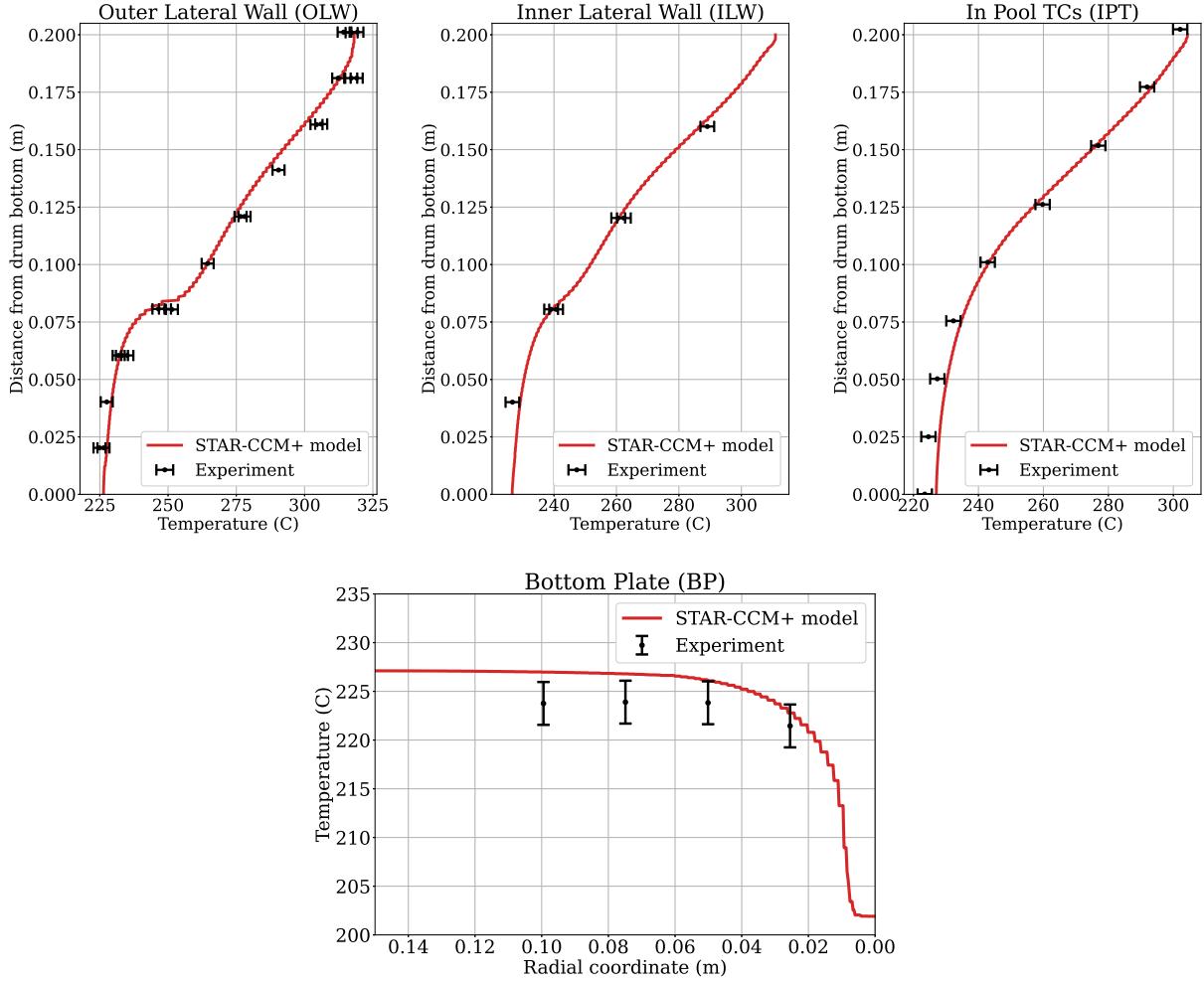


Figure 30: Comparison of STAR-CCM+ model's test section temperatures against experiment during the final natural circulation state.

Criterion 1 – the natural circulation steady state

Criterion 2 – flow oscillation frequency and decay rate following the flow reversal in the test section leg

Criterion 3 – transient temperature peak in the inlet to the test section during the flow reversal

With these three criteria in mind, Fig. 31 shows each leg's flow rate and inlet/outlet temperatures, comparing SAM-STAR-CCM+ coupled results, SAM Standalone results and experiment data. First, SAM standalone results are be discussed followed by the SAM-STAR-CCM+ coupled results. For brevity, main heater, test section, and heat exchanger will be abbreviated to MH, TS, and HX, respectively.

Starting with the flow rates in Fig. 31, the SAM Standalone model is able to predict flow reversal in the TS leg around 135 seconds, with proper MH and TS flow rates at the same time. However, the TS leg's first flow recovery peaks around 450 seconds, about 100 seconds sooner than the experiment's peak, and the MH leg's flow reverses more than what is seen in the experiment. Due to the early peak in flow recovery, the subsequent flow oscillations in all three legs also occur earlier than seen in the experiment. That being said, the frequency and decay rates are comparable to the experiments and thus satisfies **Criterion 2** fairly well. Also, the SAM standalone model is able to predict a proper final natural circulation state, with the largest flow rate discrepancy being 5% in the HX leg and satisfies **Criterion 1**. For leg temperatures in Fig. 31, similar trends are seen as in the flow rates oscillations with early temperature peaks during the first flow

recovery and subsequent oscillations. The TS leg's outlet temperature peak is overestimated compared to the experiment, while the inlet temperature peak during flow reversal is not captured by the SAM Standalone model, not satisfying **Criterion 3**. The final natural circulation temperatures match the experiment well, which is expected since the final flow rates do the same.

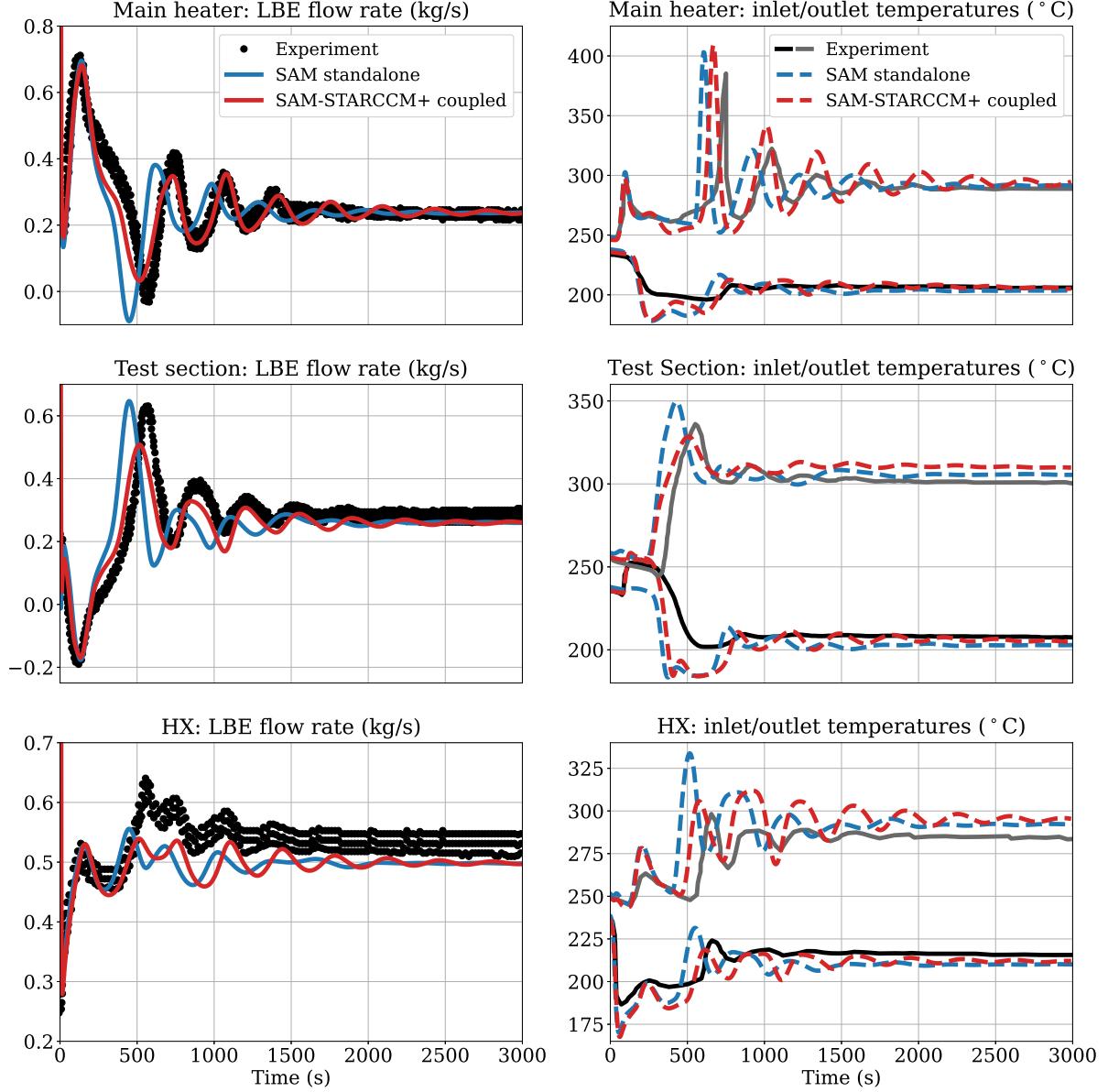


Figure 31: TALL-3D flow rates and temperatures during the TG03.S301.04 transient.

As for the SAM-STAR-CCM+ coupled results in Fig. 31, the MH, TS, and HX legs' flow rate oscillation frequencies and decay rates match experiment better than SAM Standalone, although the coupled model underestimates the TS leg's peak flow recovery around 500 seconds. That being said, the rest of the flow oscillations in the MH and TS legs match the experiment well, thus satisfying **Criterion 2** better than the SAM Standalone model. The coupled model also predicts the proper final natural circulation state with the largest flow rate discrepancy being 5% in the HX leg and therefore satisfies **Criterion 1**. The SAM-STAR-CCM+ coupled temperatures in Fig. 31 show similar results to SAM Standalone. However, the timing of

temperature oscillations matches the experiment better, which is expected as the flow rates does the same. Looking at the TS leg's temperatures, the SAM-STAR-CCM+ coupled inlet temperature peak during flow reversal matches the experiment well and satisfies **Criterion 3** unlike the SAM Standalone model. This ability shows a clear advantage of using CFD modeling during such a reversed flow transient.

For additional visualization of the coupled system, the SAM-STAR-CCM+ coupling's spatial results are shown using the open-source visualization tool ParaView. A video is used to visualize the coupled time-dependent behavior of the facility during the TG03.S301.04 transient. A full video is available upon request, but two screenshots are shown in Fig. 32 and 33. These figures show the coupled temperature field alongside LBE flow rate plots for each leg, where the overlapping SAM component is omitted for ease of visualization. Here, SAM provides the 1D solution from most of the system while STAR-CCM+ provides the 2D solution from only the test section portion of the TALL-3D facility. Furthermore, the figures show a mirrored STAR-CCM+ solution solely for better visualization, and mirroring the solution is applicable because the solution comes from a 2D axisymmetric model. Also for reference, the STAR-CCM+ modeled test section is 0.9 meters tall, or about 3 feet.

Fig. 32 shows the initial forced circulation state, where the STAR-CCM+ modeled test section's temperature field follows what is expected for an imposed jet of cool fluid inside a heated large enclosure. This also shows jet impingement at the inner disc within the test section, which enhances mixing. Fig. 33 shows the final natural circulation state, where large temperature gradients can be observed both within the SAM domain and also the STAR-CCM+ domain. Furthermore, the STAR-CCM+ model predicts a large amount of thermal stratification within the test section, so much that the inlet jet is suppressed well below the location of the inner disc.

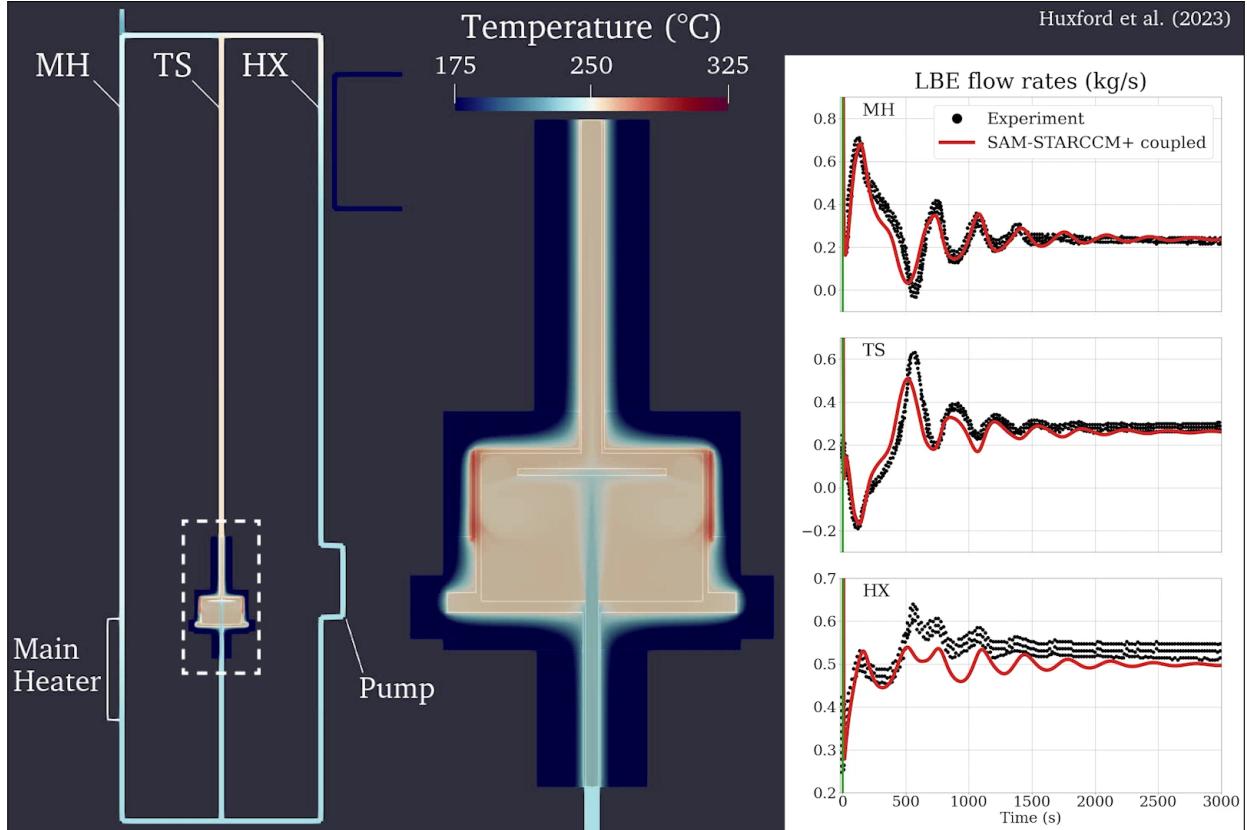


Figure 32: SAM-STAR-CCM+ coupled spatial solution at the intial forced circulation state.

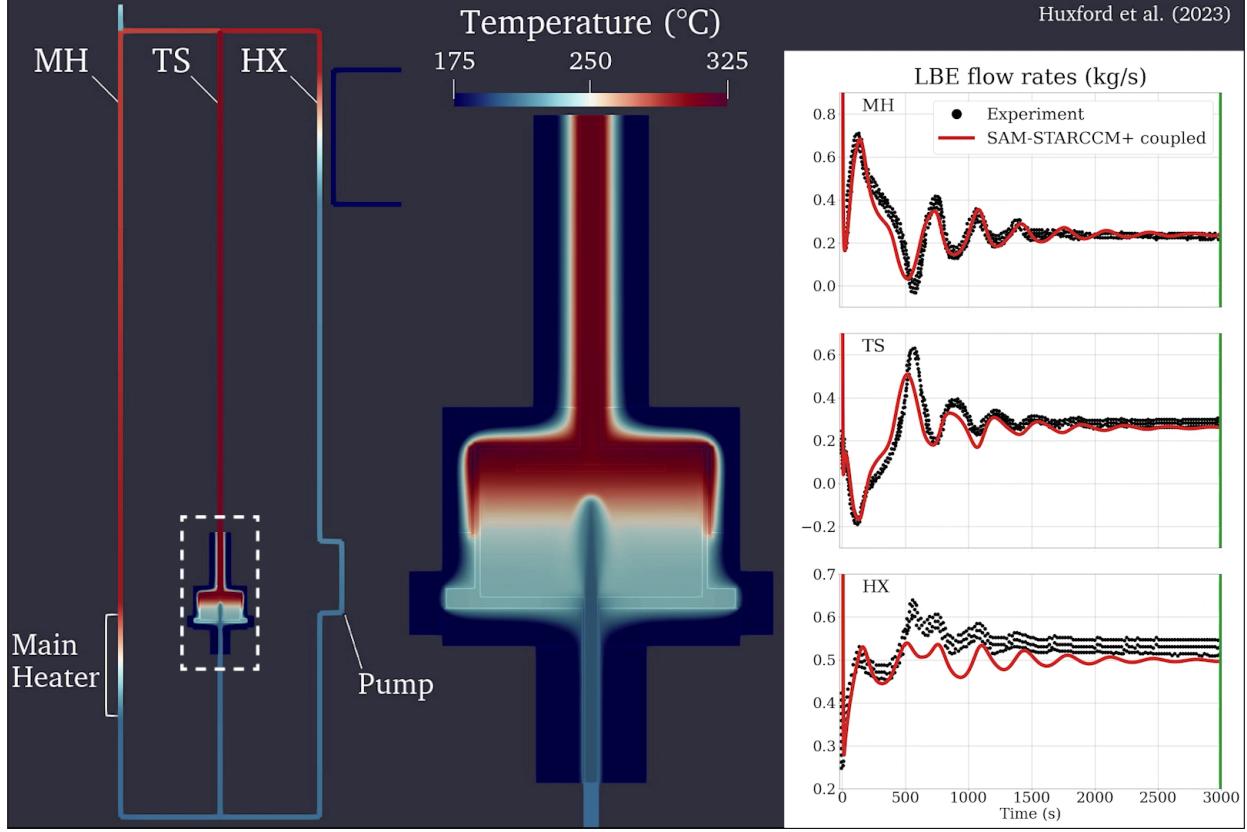


Figure 33: SAM-STAR-CCM+ coupled spatial solution at the final natural circulation state.

6 ADDITIONS TO CARDINAL AND SAM

The present SAM-NekRS coupling has been implemented into the open-source Cardinal repository located here: <https://github.com/neams-th-coe/cardinal>. Users can utilize the `NekRSSeparateDomainProblem` problem class in order to implement the domain decomposition coupling method. This class automates several NekRS-based Postprocessors needed for the domain decomposition method. Namely, Postprocessors needed for the coupling of NekRS to SAM in Fig. 1 are created to obtain U_{CFD} , ΔP_{CFD} and T_{CFD} without user intervention. The class allows for additional passive scalar coupling based on user input. Also, Cardinal’s front page contains highlighted implementations of Cardinal for multi-scale or multi-physics coupling. A snapshot of the front page is shown in Fig. 34 and highlights the coupled SAM-NekRS tracer transport within the double T-junction.

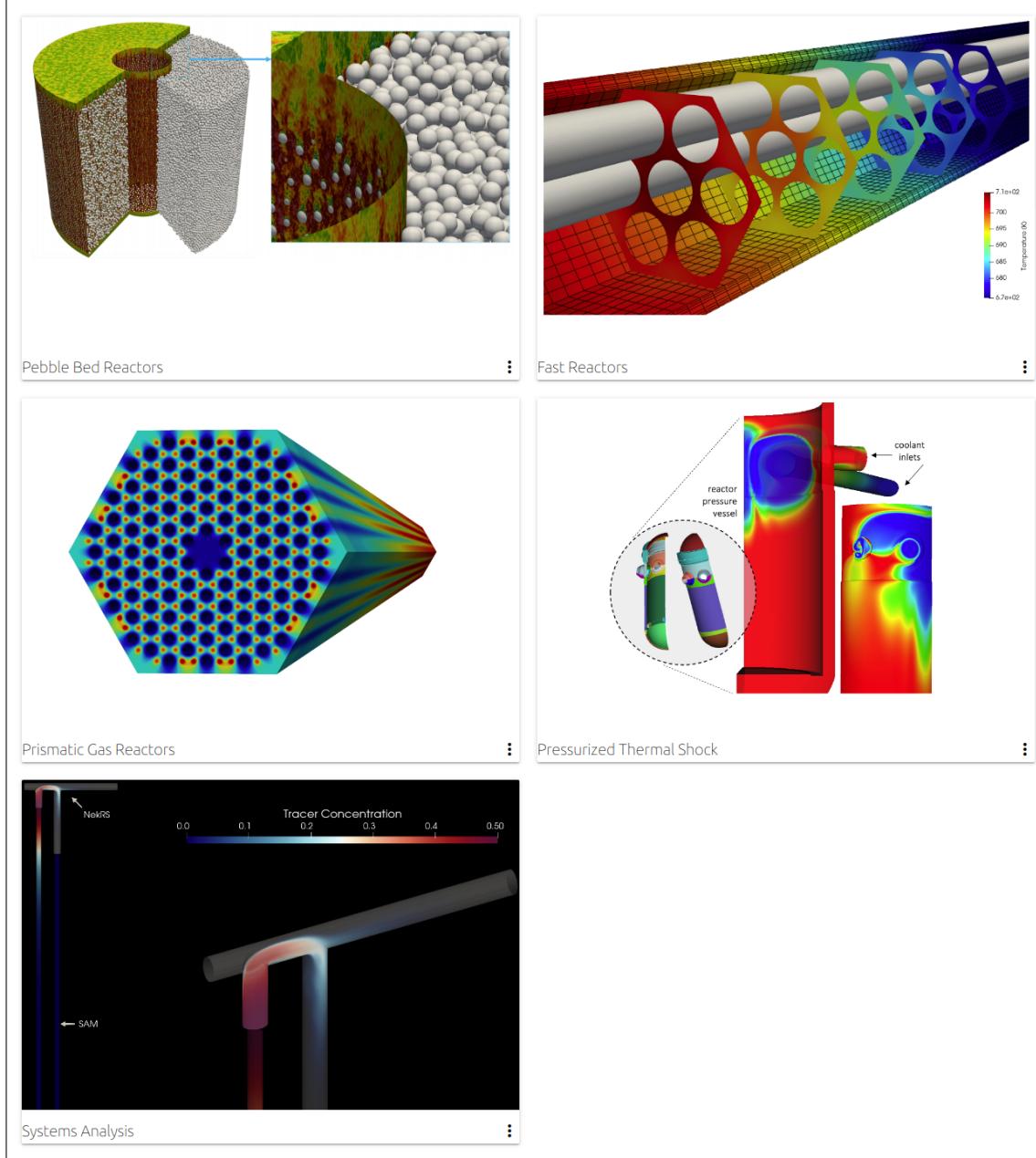


Figure 34: Snapshot of Cardinal's front page, highlighting the SAM-NekRS coupling of the double T-junction.

6.1 Cardinal's Documentation and Test Suite

The documentation for Cardinal's `NekRSSeparateDomainProblem` class is located at <https://cardinal.cels.anl.gov/source/problems/NekRSSeparateDomainProblem.html>. A snapshot of the documentation is shown in Fig. 35, and the full documentation is attached in Appendix A. This documentation provides users with information on how to correctly utilize the problem class for coupling. Namely, there's information on how Cardinal sends coupled values to NekRS's `usrwork` array and how users can correctly access specific variables when setting NekRS boundary conditions. To extend the coupling's versatility, the documentation is written in an agnostic manner to show that the class allows NekRS to be coupled with any MOOSE-based STH code.

NekRSSeparateDomainProblem

This class allows for coupling of NekRS to a 1D T/H code such as SAM or THM. This coupling is performed using the "separate-domain" coupling strategy, where the coupling between codes is performed purely by updating boundary conditions between the domains. For most applications where there's a large flow loop, the 1D T/H code should model most of the loop, and NekRS should model only a portion of the loop that is of interest for CFD simulation.

This class must be used in conjunction with two other classes in Cardinal:

1. [NekRSMesh](#), which builds a mirror of the NekRS mesh in a MOOSE format so that all the usual [Transfers](#) understand how to send data into/out of NekRS. The settings on [NekRSMesh](#) also determine which coupling type (listed above) is used.
2. [NekTimeStepper](#), which allows NekRS to control its own time stepping.

Therefore, we recommend first reading the documentation for the above classes before proceeding here.

The coupling's setup is controlled using the `coupling_type`, which provides information for how the NekRS is coupled to the 1D T/H code, via NekRS's `inlet`, `outlet`, or '`inlet outlet`'.

⚠️ WARNING

This class currently only supports dimensional solutions coming from NekRS. Nondimensional support is in progress.

Velocity, temperature, and scalar coupling

[Figure 1](#) shows the coupling of velocity (V), temperature (T), and scalar01 (S01) for `coupling_type = 'inlet outlet'`, where the 1D T/H code is coupled to NekRS's inlet and outlet boundaries.

Scalar coupling is allowed for up to 3 scalars (scalar01, scalar02, and scalar03) using the optional `coupled_scalars` parameter. Coupling scalars is useful when running a 1D T/H coupled to a NekRS RANS k-tau simulation with coupled passive scalar transport. For such a case, NekRS may use k and tau as scalar01 and scalar02, respectively, and the coupled passive scalar as scalar03. For coupling just scalar03, one would set `coupled_scalars = scalar03`.

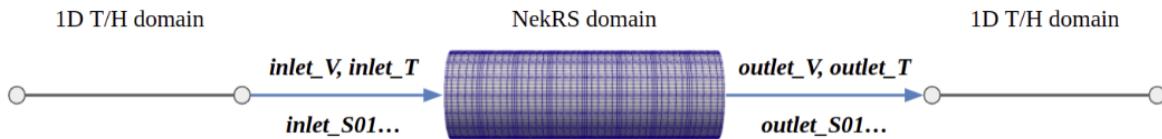


Figure 1: Example boundary transfers for 'inlet outlet' coupling.

All `inlet_XX` are single values passed from the 1D T/H domain to the NekRS inlet boundary provided by `inlet_boundary`. For information on implementing these transferred values, refer to [this section](#) of the documentation.

All `outlet_XX` values are automated [NekSideAverage](#) postProcessors that are created on the NekRS boundary ID given by `outlet_boundary`. These postProcessors can then be transferred to the 1D T/H code using the [MultiAppPostprocessorTransfer](#) system.

Figure 35: Snapshot of Cardinal's `NekRSSeparateDomainProblem` coupling documentation.

A full test suite was added to Cardinal to ensure the `NekRSSeparateDomainProblem` class works correctly as Cardinal is continually updated. The tests are based on MOOSE's Test System and contains several regression tests using `CSVDiff` MOOSE-type tests. The test suite also includes several error checks using `RunException` MOOSE-type tests to ensure Cardinal throws the correct error to users when they provide incorrect inputs to the problem class. One such error check makes sure users provide coupled boundary IDs that do not exist within the NekRS mesh.

6.2 SAM's Code Modifications, Documentation and Testing

Additions were made to SAM's official repository under merge request 668 such that SAM can utilize a new domain-overlapping coupled friction factor. Code modifications were made to SAM's `PBOneDFluidComponent` such that a user can choose where and when to utilize a domain-overlapping friction factor. This is done by controlling new parameters of a boolean `overlap_coupled` and a Postprocessor name `overlap_pp`. Explicitly, `overlap_coupled` turns on the domain-overlapping coupling for a specific `PBOneDFluidComponent` component, and `overlap_pp` provides a Postprocessor value for the bracketed term of Eq. 4. This methodology allows for flexibility when coupling SAM to any MOOSE MultiApp system, whether it be NekRS, another CFD code or any other higher-order methodology.

Additional documentation has been added to SAM's user guide for `PBOneDFluidComponent` regarding the added user parameters. Also, additional information was added under the wall friction factor correlation

documentation specifying how `overlap_coupled` and `overlap_pp` should be used.

A test has been added to SAM utilizing a SAM-SAMmultiD coupling, where the SAMmultiD model acts as a surrogate for a CFD model. A schematic of the test's setup is shown in Fig. 36, where the SAMmultiD model includes flow through a 3D porous medium. Due to the porous medium, the SAMmultiD model predicts a higher pressure drop than the nominal SAM 1D model.

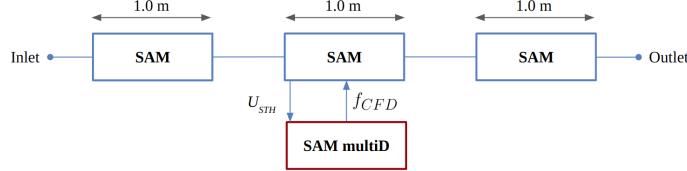


Figure 36: Schematic of the SAM-SAMmultiD coupling test.

The test follows a transient varying the inlet Reynolds number over time following Eq. 9. To limit time consumption, the test included in the official SAM repository only contains a fraction of the entire transient. For completeness here, the entire transient is carried out and simulated.

$$Re_{inlet}(t) = \begin{cases} 100, & t \leq 0.2 \text{ s}, \\ 100 + 100(t - 0.2), & 0.2 \text{ s} < t < 1.2 \text{ s}, \\ 200, & t \geq 1.2 \text{ s}, \end{cases} \quad (9)$$

The transient pressure drop is compared between SAM standalone, SAMmultiD standalone, and coupled SAM-SAMmultiD simulations using domain decomposition and overlapping coupling methods. Results of the test are shown in Fig. 37 and match the same trends shown in previous SAM-NekRS coupled simulations that used a similar setup. This test therefore ensures SAM's official repository can utilize the domain-overlapping coupling for coupling SAM to higher-order models.

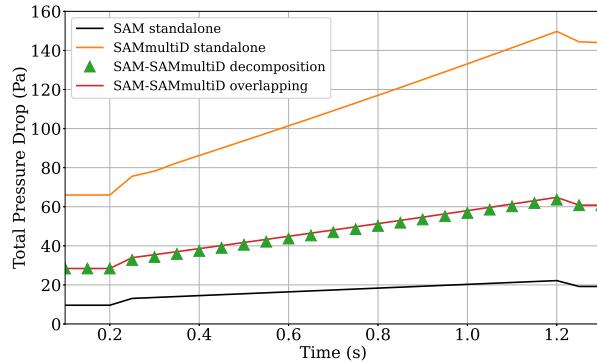


Figure 37: Transient results for the SAM-SAMmultiD coupling test.

6.3 Continued Cardinal and SAM Work

A full Cardinal tutorial is in progress utilizing the Double T-junction validation case from Sec. 5.1 and will be added in the near future. The domain overlapping coupling is also being added to Cardinal and will likely be held within its own problem class separate from the domain decomposition's class. In addition, SAM is receiving another test considering additional energy coupling.

7 CONCLUSIONS

The coupling between the STH code SAM and the CFD code NekRS has been presented using domain decomposition and hybrid domain overlapping coupling methods with an explicit coupling scheme. The coupling was verified against several canonical open and closed-loop configurations. Furthermore, the numerical stability of each coupling method was compared using a pump-driven closed loop and natural circulation loop, both of which contain tightly-coupled loop configurations. In both cases, domain overlapping was shown to be more stable, especially in the natural circulation loop case where the domain decomposition method's simulation diverged.

The hybrid domain overlapping coupling method was validated against a double T-junction experiment from open literature. The present SAM-NekRS coupled simulation was able capture the complexity of 3D mixing in the double T-junction. Furthermore, the SAM-NekRS coupling provided superior results matching experiment data better than previous TRACE-CFX coupling of the same system. This is partly due to SAM utilizing higher-order numerical methods than TRACE, resulting in low amount of numerical diffusion that would greatly affect passive scalar transport. Also, the NekRS model utilizes a different RANS model than the CFX model, so the NekRS RANS $k-\tau$ model likely predicts a flow field more similar to the experiment as compared to the CFX RANS SST model.

The SAM-NekRS coupling has been implemented into Cardinal as an open-source resource for the nuclear energy research community. The coupling is held within its own problem class and contains thorough documentation and a full test suite to ensure ease-of-use for users of Cardinal. The coupling is continually updated and will contain a full user tutorial as well as the domain overlapping coupling implementation in the near future. Also, the domain-overlapping coupled friction factor has been implemented into the official SAM repository with additional documentation and testing. Further additions will be made as more robust testing is included.

For further validation, the TALL-3D facility's open STH/CFD coupling benchmark was used with its TG03.S301.04 transient from forced to natural circulation. In this validation case, STAR-CCM+ was used as the CFD code instead of NekRS. The SAM-STAR-CCM+ coupled model's results were able to meet all three criteria of successful modeling of the transient as outlined by the TALL-3D facility's creators. The SAM standalone model was only able to met two of the three criteria. The TALL-3D facility's semi-blind TG03.S301.03 benchmark transient has also been ran for additional validation of the coupling and is currently being finalized.

For application of the coupling method presented, SAM-STAR-CCM+ coupling is being applied to the Advanced Burner Test Reactor's Protected Loss of Forced Cooling accident. A SAM model for this transient already existed within the SAM repository, and the SAM team at ANL kindly provided their STAR-CCM+ model for use in the coupling application. In particular, special thanks goes to the original creators of the model: Rui Hu, J.W. Thomas, Enerel Munkhzul and T.H. Fanning. Only minor changes were needed to update the STAR-CCM+ model to then run the SAM-STAR-CCM+ coupled simulation. This coupling application is already being ran with preliminary results already obtained. The coupling application is just being finalized now.

References

- Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil M. Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Václav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. PETSc Web page. <https://petsc.org/>, 2022. URL <https://petsc.org/>.
- J. Benton, G. Kalitzin, and A. Gould. Application of two-equation turbulence models in aircraft design. AIAA 34th Aerospace Sciences Meeting and Exhibit, 01 1996. doi: 10.2514/6.1996-327. URL <https://arc.aiaa.org/doi/abs/10.2514/6.1996-327>.
- Davide Bertolotto. *Coupling a System Code with Computational Fluid Dynamics for the Simulation of Complex Coolant Reactivity Effects*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2011.
- Davide Bertolotto, Annalisa Manera, Simon Frey, Horst-Michael Prasser, and Rakesh Chawla. Single-phase mixing studies by means of a directly coupled cfd/system-code tool. *Annals of Nuclear Energy*, 36(3):310–316, 2009. ISSN 0306-4549. doi: <https://doi.org/10.1016/j.anucene.2008.11.027>. URL <https://www.sciencedirect.com/science/article/pii/S0306454908003095>. PHYSOR 2008.
- N. Chalmers, A. Karakus, A. P. Austin, K. Swirydowicz, and T. Warburton. libParanumal: a performance portable high-order finite element library, 2022. URL <https://github.com/paranumal/libparanumal>. Release 0.5.0.
- I.J.V. De Kock, M.M. Stempniewicz, F. Roelofs, G. Hu, and R. Hu. Comparison of the approaches followed towards modeling of the tall-3d facility using the systems thermal hydraulics codes sam and spectra. In *Proceedings NURETH*, 2022.
- P. Fischer, S. Kerkemeier, M. Min, Y. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, and T. Warburton. NekRS, a GPU-Accelerated Spectral Element Navier-Stokes Solver, April 2021a. URL <https://arxiv.org/abs/2104.05829>. arXiv:2104.05829.
- Paul Fischer, Elia Merzari, Misun Min, Stefan Kerkemeier, Yu-Hsiang Lan, Malachi Phillips, Thilina Rathnayake, April Novak, Derek Gaston, Noel Chalmers, and Tim Warburton. Highly optimized full-core reactor simulations on summit, 2021b. URL <https://arxiv.org/abs/2110.01716>.
- Paul Fischer, Elia Merzari, Misun Min, Stefan Kerkemeier, Yu-Hsiang Lan, Malachi Phillips, Thilina Rathnayake, April Novak, Derek Gaston, Noel Chalmers, and Tim Warburton. Highly optimized full-core reactor simulations on summit, 2021c. URL <https://arxiv.org/abs/2110.01716>.
- Derek R. Gaston, Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andriš, Yaqi Wang, Michael P. Short, Danielle M. Perez, Michael R. Tonks, Javier Ortensi, Ling Zou, and Richard C. Martineau. Physics-based multiscale coupling for full core nuclear reactor simulation. *Annals of Nuclear Energy*, 84:45–54, 2015.
- Christophe Geuzaine and Jean-François Remacle. 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, September 2009. ISSN 0029-5981. doi: 10.1002/nme.2579. URL <http://dx.doi.org/10.1002/nme.2579>.
- Dmitry Grishchenko, Marti Jelsov, Kaspar Kööp, Aram Karbojian, Walter Villanueva, and Pavel Kudinov. The tall-3d facility design and commissioning tests for validation of coupled sth and cfd codes. *Nuclear Engineering and Design*, 290:144–153, 2015. ISSN 0029-5493. doi: <https://doi.org/10.1016/j.nucengdes.2014.11.045>. URL <https://www.sciencedirect.com/science/article/pii/S0029549314006803>. Thermal-Hydraulics of Innovative Nuclear Systems.

Timothy Grunloh. *A Novel Multi-Scale Domain Overlapping CFD/STH Coupling Methodology for Multi-Dimensional Flows Relevant to Nuclear Applications*. PhD thesis, University of Michigan, 2016.

T.P. Grunloh and A. Manera. A novel domain overlapping strategy for the multiscale coupling of cfd with 1d system codes with applications to transient flows. *Annals of Nuclear Energy*, 90:422–432, 2016. ISSN 0306-4549. doi: <https://doi.org/10.1016/j.anucene.2015.12.027>. URL <https://www.sciencedirect.com/science/article/pii/S0306454915300372>.

Rui Hu et al. Sam theory manual. Technical Report ANL/NSE-17/4 Rev. 1, Argonne National Laboratory, 2021.

Alexander J. Huning, Sriram Chandrasekaran, and Srinivas Garimella. A review of recent advances in htgr cfd and thermal fluid analysis. *Nuclear Engineering and Design*, 373:111013, 2021. ISSN 0029-5493. doi: <https://doi.org/10.1016/j.nucengdes.2020.111013>. URL <https://www.sciencedirect.com/science/article/pii/S0029549320305070>.

Aaron Huxford, Victor Coppo Leite, Elia Merzari, Ling Zou, Victor Petrov, and Annalisa Manera. Development of innovative overlapping-domain coupling between sam and nekrs. In *Proceedings NURETH*, 2022.

I. E. Idelchik. *Handbook of hydraulic resistance (2nd revised and enlarged edition)*. 1986.

Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. **libMesh**: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4): 237–254, 2006. <https://doi.org/10.1007/s00366-006-0049-3>.

S.P. Kok, J.C.; Spekreijse. Efficient and accurate implementation of the k-omega turbulence model in the NLR multi-block Navier-Stokes system. Technical Report NLR NLR-TP-2000-144, National Aerospace Laboratory, 2000.

Pavel Kudinov and Dmitry Grishchenko. Tall-3d sth/cfd simulations. Technical Report SESAME – 654935 – D5.13, KTH Royal Institute of Technology, 2019a.

Pavel Kudinov and Dmitry Grishchenko. Tall-3d simulation validation sumary report. Technical Report SESAME – 654935 – D5.17, KTH Royal Institute of Technology, 2019b.

Pavel Kudinov and Dmitry Grishchenko. Tall-3d first series simulations summary report. Technical Report SESAME – 654935 – D5.12, KTH Royal Institute of Technology, 2019c.

P.J. Linstrom and Eds. W.G. Mallard. Nist chemistry webbook. Technical Report NIST Standard Reference Database Number 69, National Institute of Standards and Technology, Gaithersburg, MD, 2022.

E. Merzari, H. Yuan, M. Min, D. Shaver, R. Rahaman, P. Shriwise, P. Romano, A. Talamo, Y. Lan, D. Gaston, R. Martineau, P. Fischer, and Y. Hassan. Cardinal: A lower length-scale multiphysics simulator for pebble-bed reactors. *Nuclear Technology*, 2021. doi: <https://doi.org/10.1080/00295450.2020.1824471>. URL <https://www.tandfonline.com/doi/full/10.1080/00295450.2020.1824471>.

Elia Merzari, Jun Fang, Dillon R. Shaver, Yu-Hsiang Lan, Misun Min, Paul F. Fischer, Ronald Omaha Rahaman, and Paul K. Romano. Initial full core smr simulations with nekrs. Technical report, Argonne National Laboratory, 10 2020. URL <https://www.osti.gov/biblio/1762135>.

April Novak, Dillon Shaver, and Bo Feng. Conjugate heat transfer coupling of nekrs and moose for bypass flow modeling. In *Proceedings ANS*, 2021.

April Novak, David Andrs, Patrick Shriwise, Dillon Shaver, Paul Romano, E Merzari, and P Keutelian. Coupled monte carlo and thermal-hydraulics modeling of a prismatic gas reactor fuel assembly using cardinal. In *Proceedings PHYSOR*, 2022a.

April Novak, Patrick Shriwise, Ronald Rahaman, Paul Romano, E Merzari, and D Gaston. Coupled monte carlo transport and conjugate heat transfer for wire-wrapped bundles within the moose framework. In *Proceedings NURETH*, 2022b.

OECD. Handbook on lead-bismuth eutectic alloy and lead properties, materials compatibility, thermal-hydraulics and technologies. Technical Report 978-92-64-99002-9, OECD/Nuclear Energy Agency Nuclear Science Committee, 2007.

Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, Jason M. Miller, John W. Peterson, Andrew E. Slaughter, Roy H. Stogner, and Richard C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 2020. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2020.100430>. URL <http://www.sciencedirect.com/science/article/pii/S2352711019302973>.

Dillon Shaver, Aleks Obabko, Ananias Tomboulides, Victor C. Leite, Yu-Hsiang Lan, MiSun Min, Paul F. Fischer, and Christopher Boyd. Nek5000 developments in support of industry and the NRC. Technical Report ANL/NSE-20/48, Argonne National Laboratory, 2020.

B.L. Smith, U. Bieder, E. Graffard, M. Heitsch, M. Henriksson, T. Höhne, E. Komen, J. Mahaffy, F. Moretti, T. Morii, P. Mühlbauer, U. Rohde, M. Scheuerer, C.-H. Song, and G. Zigh. Assessment of computational fluid dynamics (cfd) for nuclear reactor safety problems. Technical Report NEA/CSNI/R(2007)13, OECD Nuclear Energy Agency, 2008.

Zeyun Wu, Cihang Lu, Sarah Morgan, Sama Bilbao y Leon, and Matthew Bucknor. A status review on the thermal stratification modeling methods for sodium-cooled fast reactors. *Progress in Nuclear Energy*, 125:103369, 2020. ISSN 0149-1970. doi: <https://doi.org/10.1016/j.pnucene.2020.103369>. URL <https://www.sciencedirect.com/science/article/pii/S0149197020301219>.

A SAM-NekRS Coupling Documentation in Cardinal

The following appendix contains the full documentation added to Cardinal for the `NekRSSeparateDomainProblem` class used for domain decomposition coupling.

NekRSSeparateDomainProblem

This class allows for coupling of NekRS to a 1D T/H code such as SAM or THM. This coupling is performed using the "separate-domain" coupling strategy, where the coupling between codes is performed purely by updating boundary conditions between the domains. For most applications where there's a large flow loop, the 1D T/H code should model most of the loop, and NekRS should model only a portion of the loop that is of interest for CFD simulation.

This class must be used in conjunction with two other classes in Cardinal:

1. [NekRSMesh](#), which builds a mirror of the NekRS mesh in a MOOSE format so that all the usual [Transfers](#) understand how to send data into/out of NekRS. The settings on [NekRSMesh](#) also determine which coupling type (listed above) is used.
2. [NekTimeStepper](#), which allows NekRS to control its own time stepping.

Therefore, we recommend first reading the documentation for the above classes before proceeding here.

The coupling's setup is controlled using the `coupling_type`, which provides information for how the NekRS is coupled to the 1D T/H code, via NekRS's `inlet`, `outlet`, or '`inlet outlet`'.

⚠ WARNING

This class currently only supports dimensional solutions coming from NekRS. Nondimensional support is in progress.

Velocity, temperature, and scalar coupling

[Figure 1](#) shows the coupling of velocity (V), temperature (T), and scalar01 (S01) for `coupling_type = 'inlet outlet'`, where the 1D T/H code is coupled to NekRS's inlet and outlet boundaries.

Scalar coupling is allowed for up to 3 scalars (scalar01, scalar02, and scalar03) using the optional `coupled_scalars` parameter. Coupling scalars is useful when running a 1D T/H coupled to a NekRS RANS k-tau simulation with coupled passive scalar transport. For such a case, NekRS may use k and tau as scalar01 and scalar02, respectively, and the coupled passive scalar as scalar03. For coupling just scalar03, one would set `coupled_scalars = scalar03`.

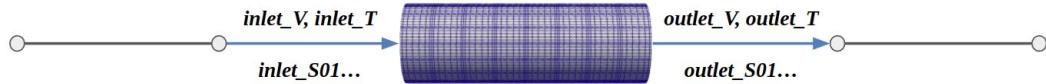


Figure 1: Example boundary transfers for 'inlet outlet' coupling.

All `inlet_XX` are single values passed from the 1D T/H domain to the NekRS inlet boundary provided by `inlet_boundary`. For information on implementing these transferred values, refer to [this section](#) of the documentation.

All `outlet_XX` values are automated `NekSideAverage` postProcessors that are created on the NekRS boundary ID given by `outlet_boundary`. These postProcessors can then be transferred to the 1D T/H code using the `MultiAppPostprocessorTransfer` system.

Pressure coupling

Pressure information is transferred between NekRS and the 1D T/H code using a global pressure drop because the 1D T/H code only needs the overall pressure drop over the NekRS domain. `inlet_P` and `outlet_P` `NekSideAverage` postProcessors are created along with the pressure drop `dP` postProcessor. **Figure 2** shows this process, which is performed no matter what `coupling_type` is given. `dP` can be transferred to the 1D T/H code using the `MultiAppPostprocessorTransfer` system.

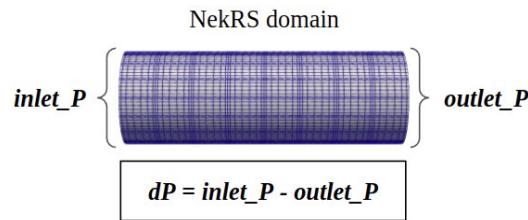


Figure 2: Example pressure drop calculation over the NekRS domain.

Transferring data between NekRS and the 1D T/H code

A summary of postProcessors generated is shown in the following table for velocity, temperature, and scalar01 coupling. PostProcessors for scalar02 and scalar03 coupling follow the same pattern as scalar01. The postProcessor created depends on (1) the `coupling_type` given, (2) whether or not NekRS is solving for temperature and (3) the optional `coupling_scalars` provided.

<code>postProcessor</code>	<code>coupling_type = created</code>	<code>coupling_type = inlet</code>	<code>coupling_type = outlet</code>	<code>coupling_type = 'inlet outlet'</code>
<code>dP</code>		✓	✓	✓
<code>inlet_V</code>		✓		✓
<code>outlet_V</code>			✓	✓

postProcessor	coupling_type = created	coupling_type = inlet	coupling_type = outlet	coupling_type = 'inlet outlet'
inlet_T		✓, if temperature solved		✓, if temperature solved
outlet_T			✓, if temperature solved	✓, if temperature solved
inlet_S01		✓, if scalar01 in coupled_scalars		✓, if scalar01 in coupled_scalars
outlet_S01			✓, if scalar01 in coupled_scalars	✓, if scalar01 in coupled_scalars

Example MultiApp transfers for velocity, temperature and scalar01 coupling are shown below for **coupling_type = 'inlet outlet'** and **coupled_scalars = 'scalar01'**. NekRS can run as either the MainApp or SubApp, but these example transfers assume NekRS is acting as the MainApp and the **1D_TH_code** as the SubApp.

```
[Transfers]
[dP_transfer]
  type = MultiAppPostprocessorTransfer
  to_multi_app = 1D_TH_code
  from_postprocessor = dP
  to_postprocessor = fromNekRS_pressureDrop
[]
[inlet_V_transfer]
  type = MultiAppPostprocessorTransfer
  from_multi_app = 1D_TH_code
  reduction_type = average
  from_postprocessor = toNekRS_velocity
  to_postprocessor = inlet_V
[]
[inlet_T_transfer]
  type = MultiAppPostprocessorTransfer
  from_multi_app = 1D_TH_code
  reduction_type = average
  from_postprocessor = toNekRS_temperature
  to_postprocessor = inlet_T
[]
[inlet_S01_transfer]
  type = MultiAppPostprocessorTransfer
  from_multi_app = 1D_TH_code
```

Using transferred data for NekRS inlet boundary conditions

Data is "sent" into NekRS by writing into the **nrs->usrwrk** scratch space array, which NekRS makes available within the boundary condition functions in the **.oudf** file (on device, this array is technically called the **nrs->o_usrwrk** array). **Table 1** shows the assignment of "slots" in the **nrs->usrwrk** scratch space array with quantities written by Cardinal. Because different quantities are written into Cardinal depending on the problem setup, if a particular slice is not needed for a case, it will just hold zero values. That is, the *order* of the various quantities is always the same in **nrs->usrwrk**.

Table 1: Quantities written into the scratch space array by Cardinal

Slice	Quantity	When Will There be Non-Zero Values?	How to Access in the .oudf File
0	Inlet velocity	if <code>coupling_type</code> includes <code>inlet</code>	<code>bc->wrk[0 * bc->fieldOffset + bc->idM]</code>
1	Inlet temperature	if <code>coupling_type</code> includes <code>inlet</code> and NekRS's case files include a temperature solve	<code>bc->wrk[1 * bc->fieldOffset + bc->idM]</code>
2	Inlet scalar01	if <code>coupling_type</code> includes <code>inlet</code> and <code>coupled_scalars</code> includes <code>scalar01</code>	<code>bc->wrk[2 * bc->fieldOffset + bc->idM]</code>
3	Inlet scalar02	if <code>coupling_type</code> includes <code>inlet</code> and <code>coupled_scalars</code> includes <code>scalar02</code>	<code>bc->wrk[3 * bc->fieldOffset + bc->idM]</code>
4	Inlet scalar03	if <code>coupling_type</code> includes <code>inlet</code> and <code>coupled_scalars</code> includes <code>scalar03</code>	<code>bc->wrk[4 * bc->fieldOffset + bc->idM]</code>

The total number of slots in the scratch space that are allocated by Cardinal is controlled with the `n_usrwrk_slots` parameter. If you need to use extra slices in `nrs->usrwrk` for other custom user actions, simply set `n_usrwrk_slots` to be greater than the number of slots strictly needed for coupling. At the start of your Cardinal simulation, a table will be printed to the screen to explicitly tell you what each slice in the scratch space holds. Any extra slots are noted as `unused`, and are free for non-coupling use.

For example, if your case couples NekRS to MOOSE with `coupling_type = inlet` and `coupled_scalars = scalar02`, but has a NekRS case without a temperature solve, the slices normally dedicated to storing inlet temperature and scalar01 are still allocated (because we keep the order the same in `nrs->usrwrk`), but just won't hold any meaningful information. A table similar to the following would print out at the start of your simulation. You could use slices 4 onwards for custom purposes.

Listing 1: Table printed at start of Cardinal simulation that describes available scratch space for a case that couples NekRS to MOOSE via an inlet and with the second passive scalar, but without the first passive scalar or a temperature solve. A total of 7 slots are allocated by setting `n_usrwrk_slots` to 7

Slice	Quantity	How to Access in NekRS BCs
0	velocity	<code>bc->wrk[0 * bc->fieldOffset + bc->idM]</code>
1	temperature	<code>bc->wrk[1 * bc->fieldOffset + bc->idM]</code>
2	scalar01	<code>bc->wrk[2 * bc->fieldOffset + bc->idM]</code>
3	scalar02	<code>bc->wrk[3 * bc->fieldOffset + bc->idM]</code>
4	unused	<code>bc->wrk[4 * bc->fieldOffset + bc->idM]</code>
5	unused	<code>bc->wrk[5 * bc->fieldOffset + bc->idM]</code>
6	unused	<code>bc->wrk[6 * bc->fieldOffset + bc->idM]</code>

⚠ WARNING

Allocation of `nrs->usrwrk` and `nrs->o_usrwrk` is done automatically by `NekRSSeparateDomainProblem`. If you attempt to run a NekRS input file that accesses `bc->wrk` in the `.oudf` file *without* a Cardinal executable (i.e. using something like `nrsmpi case 4`), then that scratch space will have to be manually allocated in the `.udf` file, or else your input will seg fault. This use case will not be typically encountered by most users, but if you really do want to run the NekRS input files intended for a Cardinal case with the NekRS executable (perhaps for debugging), we recommend simply replacing `bc->wrk` by a dummy value, such as `bc->s = 0.0` for the inlet scalar use case. This just replaces a value that normally comes from MOOSE by a fixed value. All other aspects of the NekRS case files should not require modification.

In other words, the scratch space slots contain pointwise values for each (singly-valued) postprocessor value on all the Gauss-Lobatto-Legendre (GLL) points on the corresponding boundary. So, the values in each "slot" correspond to the following postprocessors:

- Slot 0: `inlet_V` postprocessor value
- Slot 1: `inlet_T` postprocessor value
- Slot 2: `inlet_S01` postprocessor value
- Slot 3: `inlet_S02` postprocessor value
- Slot 4: `inlet_S03` postprocessor value

This allows the user the freedom to choose what type of profile they want to implement at the inlet boundary, i.e. flat or fully-developed. Below are a few example implementations.

Example for a flat velocity(x) profile:

```
void velocityDirichletConditions(bcData *bc)
{
    bc->u = bc->wrk[bc->idM];
    bc->v = 0.0;
    bc->w = 0.0;
}
```

Example for a flat temperature and scalar profiles:

```
void scalarDirichletConditions(bcData *bc)
{
    if(bc->scalarId==0) bc->s = bc->wrk[bc->idM + bc->fieldOffset]; // temperature
    if(bc->scalarId==1) bc->s = bc->wrk[bc->idM + 2*bc->fieldOffset]; // scalar01
    if(bc->scalarId==2) bc->s = bc->wrk[bc->idM + 3*bc->fieldOffset]; // scalar02
    if(bc->scalarId==3) bc->s = bc->wrk[bc->idM + 4*bc->fieldOffset]; // scalar03
}
```

Example for fully-developed velocity(x) profile for laminar pipe flow:

```

void velocityDirichletConditions(bcData *bc)
{
    dfloat yi = bc->y;
    dfloat zi = bc->z;
    dfloat rsq = yi*yi + zi*zi;
    dfloat Rsq = 0.5*0.5; // Radius^2
    dfloat Uavg = bc->wrk[bc->idM];

    bc->u = 2*Uavg*(1-rsq/Rsq);
    bc->v = 0.0;
    bc->w = 0.0;
}

```

Other Features

This class mainly facilitates data transfers to and from NekRS. A number of other features are implemented in order to enable nondimensional solutions, improved communication, and convenient solution modifications. These are described in this section.

Outputting the Scratch Array

This class (and all other NekRS wrappings in Cardinal) allows you to write slots in the **nrs->usrwrk** scratch space array to NekRS field files. This can be useful for viewing the data sent from MOOSE to NekRS (for problem classes that involve multiphysics), as well as to visualize custom user usage of **nrs->usrwrk**, such as for fetching a wall distance computation from the Nek5000 backend. To write the scratch space to a field file, set **usrwrk_output** to an array with each "slot" in the **nrs->usrwrk** array that you want to write. Then, specify a filename prefix to use to name each field file.

In the example below, the first two "slots" in the **nrs->usrwrk** array will be written to field files on the same interval that NekRS writes its usual field files. These files will be named **aaabrick0.f00001**, etc. and **cccbbrick0.f00001**, etc. Based on limitations in how NekRS writes its files, the fields written to these files will all be named **temperature** when visualized.

```

[Problem]
type = NekRSStandaloneProblem
casename = 'brick'
usrwrk_output = '0 1'
usrwrk_output_prefix = 'aaa ccc'
[]

```

([test/tests/nek_file_output/usrwrk/nek.i](#))

Input Parameters

▼ Required Parameters

coupling_type	NekRS boundary types to couple to a 1-D T/H code
----------------------	--

inlet_boundary	NekRS inlet boundary ID
-----------------------	-------------------------

outlet_boundary NekRS outlet boundary ID

▼ Optional Parameters

Cp_0 (1) Heat capacity parameter value for non-dimensional solution

L_ref (1) Reference length scale value for non-dimensional solution

T_ref (0) Reference temperature value for non-dimensional solution

U_ref (1) Reference velocity value for non-dimensional solution

boundary_restricted_elem_integrity_check (True) Set to false to disable checking o...

boundary_restricted_node_integrity_check (True) Set to false to disable checking o...

casename Case name for the NekRS input files; this is in .par, .udf, .oudf, and .re2. Can also...

constant_interval (1) Constant interval (in units of number of time steps) with which to s...

coupled_scalars NekRS scalars to couple to a 1-D T/H code

dT_ref (1) Reference temperature range value for non-dimensional solution

disable_fld_file_output (False) Whether to turn off all NekRS field file output writing (f...

error_on_jacobian_nonzero_reallocation (False) This causes PETSc to error if it had ...

extra_tag_matrices Extra matrices to add to the system that can be filled by objects whic...

extra_tag_solutions Extra solution vectors to add to the system that can be used by obje...

extra_tag_vectors Extra vectors to add to the system that can be filled by objects which c...

force_restart (False) EXPERIMENTAL: If true, a sub_app may use a restart file instead of u...

fv_bcs_integrity_check (True) Set to false to disable checking of overlapping Dirichlet a...

ignore_zeros_in_jacobian (False) Do not explicitly store zero values in the Jacobian ma...

kernel_coverage_check (True) Set to false to disable kernel->subdomain coverage che...

material_coverage_check (True) Set to false to disable material->subdomain coverage...

material_dependency_check (True) Set to false to disable material dependency check

minimize_transfers_in	(False)	Whether to only synchronize nekRS for the direction TO_...
minimize_transfers_out	(False)	Whether to only synchronize nekRS for the direction FR...
n_usrwrk_slots	(7)	Number of slots to allocate in nrs->usrwrk to hold fields either relate...
near_null_space_dimension	(0)	The dimension of the near nullspace
nek_mesh	(fluid)	NekRS mesh to act on
nl_sys_names	(nl0)	The nonlinear system names
nondimensional	(False)	Whether NekRS is solved in non-dimensional form
null_space_dimension	(0)	The dimension of the nullspace
output		Field(s) to output from NekRS onto the mesh mirror
parallel_barrier.messaging	(False)	Displays messaging from parallel barrier notificatio...
previous_nl_solution_required	(False)	True to indicate that this calculation requires a ...
restart_file_base		File base name used for restart (e.g. / or /LATEST to grab the latest file a...
rho_0	(1)	Density parameter value for non-dimensional solution
skip_additional_restart_data	(False)	True to skip additional data in equation system fo...
skip_nl_system_check	(True)	True to skip the NonlinearSystem check for work to do (e....
solve	(True)	Whether or not to actually solve the Nonlinear system. This is handy in the ca...
synchronization_interval	(constant)	When to synchronize the NekRS solution with the...
transpose_null_space_dimension	(0)	The dimension of the transpose nullspace
use_nonlinear	(True)	Determines whether to use a Nonlinear vs a Eigenvalue system (Au...
usrwrk_output		Usrwrk slot(s) to output to NekRS field files; this can be used for viewing t...
usrwrk_output_prefix		String prefix to use for naming the field file(s); only the first three ...
verbose_multiapps	(False)	Set to True to enable verbose screen printing related to Mult...
write_fld_files	(False)	Whether to write NekRS field file output from Cardinal. If true, this...

› Advanced Parameters

Input Files

- [\(test/tests/nek_errors/insufficient_scratch/nek_separate_domain.i\)](#)
- [\(test/tests/nek_separateddomain/invalid_params/nek.i\)](#)
- [\(test/tests/nek_separateddomain/transfers_velocity/nek.i\)](#)
- [\(test/tests/nek_separateddomain/transfers_temperature/nek.i\)](#)
- [\(test/tests/nek_errors/invalid_bid/nek_sep.i\)](#)