Verification of moltres for multiphysics simulations of fast-spectrum molten salt reactors

Sun Myung Park [a,*], Madicken Munk [a]

[a] *University of Illinois at Urbana-Champaign, Department of Nuclear, Plasma and Radiological Engineering, Urbana, IL 61801, United States*

ARTICLE INFO	ABSTRACT
	Modeling strongly coupled neutronics and thermal–hydraulics in liquid-fueled MSRs requires robust and flexible multiphysics software for accurate simulations at reasonable computational costs. In this paper, we present Moltres and its neutronics and thermal–hydraulics modeling capabilities relevant to multiphysics reactor analysis. As a MOOSE-based application, Moltres provides various multiphysics coupling schemes and time-stepping methods, including fully coupled solves with implicit time-stepping. We verified Moltres' MSR modeling capabilities against a multiphysics numerical benchmark developed for software dedicated to modeling fast-spectrum MSRs. The results show that Moltres performed comparably to participating software packages in the benchmark; the majority of the relevant quantities fell within one standard deviation of the benchmark average. Among the participating multiphysics tools in the benchmark, Moltres agrees closest to the multiphysics tool from the Delft University of Technology due to similarities in the numerical solution techniques and meshing schemes. © 2022 Elsevier Ltd. All rights reserved.

1. Introduction

Molten Salt Reactors (MSRs) have attracted increasing interest over the past two decades with various ongoing research programs funding MSR development in the EU (CORDIS, 2021), China (Dai and Dolan, 2017), and the US (DOE, 2021). These programs include efforts to develop reactor analysis software specifically tuned for MSRs. Reactor analysis software inform design choices that maximize safety, reduce proliferation risks, and improve fuel efficiency in the next generation of advanced reactors.

Liquid-fueled MSRs present new challenges in computational reactor safety analysis arising from fuel movement. MSRs feature strong negative temperature reactivity feedback in the primary fuel salt, leading to strong and near-instantaneous interactions between reactor power and thermal–hydraulics. Thus, unexpected changes in coolant flow and temperature significantly affect reactor power and vice versa, requiring tighter coupling between these physics in the software. Additionally, MSR simulation software must include capabilities to model the movement of delayed neutron precursors and heat generation in the fuel salt—physical phenomena absent in solid-fuel reactors. The movement of delayed neutron precursors impacts the effective delayed neutron fraction in the core and consequently also impacts the transient behavior of

the reactor. At the same time, the fuel salt serves a dual role of generating fission heat and cooling the reactor core, leading to stronger coupling between heat generation and salt flow dynamics than traditional Light Water Reactors (LWRs).

MSR research in the early 2000s led to the development of several novel MSR simulation tools such as Cinsf1D (Lecarpentier and Carpentier, Jan. 2003), SIMMER-III (Rineiski et al., 2005), DYN1D-MSR (Krepel et al., 2005), and DYN3D-MSR (Krepel et al., 2007). These works influenced many subsequent developments contributing to the more advanced MSR simulation tools available today.

Some of these simulation tools employ *tight coupling* to couple separate single-physics neutronics and thermal–hydraulics software. For instance, researchers at the Delft University of Technology (TU Delft) coupled the 3D neutron diffusion software DALTON (Boer et al., 2010) and the Computational Fluid Dynamics (CFD) software HEAT (de Zwaan et al., 2007) to perform a safety analysis of the Molten Salt Fast Reactor (MSFR) (Fiorina et al., 2014). In a later effort from the same institute, Tiberga et al. (2019) coupled PHANTOM-S_N and DGFlows in their participation in the CNRS benchmark study (Tiberga et al., 2020). Developed at the Centre National de la Recherche Scientifique (CNRS), the CNRS benchmark facilitates code-to-code verification of MSR multiphysics software (Aufiero et al., 2018). Another multiphysics package was developed at the Paul Scherrer Institute (PSI) coupling the thermal–hydraulics system software TRAC/RELAP Advanced

* Corresponding author.

E-mail address: smpark3@illinois.edu (S.M. Park).

Computational Engine (TRACE) (NRC, 2007) with the nodal neutron diffusion software Purdue Advanced Reactor Core Simulator (PARCS) (Downar et al., 2010) for the safety analysis of the MSFR (Pettersen and Mikityuk, 2016). Coupling single-physics software to form an integrated multiphysics solver allows researchers to leverage older, well-validated, single-physics software. These single-physics software are also highly optimized for solving specific types of partial differential equations (PDEs) relevant to the investigated system.

With modern advancements in computing hardware and growing access to high-performance computing systems, others have developed multiphysics solvers by coupling the CFD software OpenFOAM (Ltd, 2021) with the Monte Carlo particle transport software Serpent (Leppanen et al., Aug. 2014), thus achieving high-fidelity neutronics calculations in transient reactor analyses. To that end, Serpent has a multiphysics interface supporting coupling with other physics software (Leppanen, 2013). Laureau et al. (2017) separately developed another technique called the Transient Fission Matrix (TFM) method through the introduction of additional time-dependence operators to conventional fission matrices typically used to accelerate source convergence in Monte Carlo neutronics calculations. The TFM method pre-calculates three TFMs of the reactor system in Serpent and interpolates the matrix values during the actual transient calculations to incorporate the effects of temperature-induced density change and Doppler effect on the neutron cross sections and ultimately the neutron flux. Blanco (2020) took a more integrated approach by compiling Serpent as an internal `C`-based function within Open-FOAM's `C++`-based framework. This approach reduced the required data transfers between Serpent and OpenFOAM as both software have access to shared memory during runtime. Their integrated solver employed the Quasi-Static method for transient neutronics calculations and ran Serpent Monte Carlo calculations several times per timestep until the solver reached convergence.

Another MSR simulation approach involves developing "all-in-one" multiphysics software which handle all multiphysics calculations and data transfer internally. Among earlier efforts, Nicolino et al. (2008) and Zhang et al. (2009) recognized the need for more robust multiphysics coupling techniques and higher-fidelity thermal–hydraulics solutions to accurately capture complex flow profiles in pool-type MSRs. They each independently developed unnamed multiphysics solvers and demonstrated them with non-moderated MSR designs. Later, Li et al. (2015) demonstrated the steady-state and transient analysis capabilities of COUPLE, a neutronics and thermal–hydraulics software developed at the Karlsruhe Institute of Technology. Others adopted extensible software frameworks for developing numerical solvers to develop multiphysics reactor analysis software. Examples of these software frameworks include the commercial COMSOL Multiphysics® software (COMSOL, 2021), the aforementioned open-source CFD toolbox OpenFOAM, and the open-source finite-element framework Multiphysics Object-Oriented Simulation Environment (MOOSE) (Gaston et al., 2015). Researchers at Politecnico di Milano (PoliMi) developed a MSR simulation tool in COMSOL and modeled the Molten Salt Breeder Reactor (MSBR) as a single axisymmetric fuel channel with a uniform flow profile (Cammi et al., 2011), followed by the Molten Salt Reactor Experiment (MSRE) core also as a single axisymmetric fuel channel with parabola-shaped laminar flow (Cammi et al., 2012). They later expanded on their approach by modeling the MSRE upper plenum, downcomer, lower plenum, primary heat exchanger, and secondary heat exchanger as 0D systems (lumped-parameter models), and substituting the 2D fuel channel with a 3D fuel channel which more closely resembled the actual fuel channels in the MSRE (Zanetti et al., 2015). Beyond graphite-moderated MSRs, they also modeled the MSFR in the same publica-

tion that featured TU Delft's DALTON + HEAT coupled multiphysics solver.

Other institutes have dedicated significant development work towards OpenFOAM-based MSR simulation tools. Aufiero et al. (2014) first introduced an OpenFOAM model developed at PoliMi. Their model implemented a neutron diffusion model and a Reynolds-averaged Navier–Stokes (RANS)-based turbulence model with incompressible flow to demonstrate 2D and 3D transient analyses of the MSFR. Later advancements in the PoliMi solver include a fuel compressibility model with helium bubble tracking to study fuel compressibility effects (Cervi et al., 2019) and a SP_3 neutron transport model for improved neutronics calculations (Cervi et al., 2019) in the MSFR. GeN-Foam is another OpenFOAM-based tool developed by Fiorina et al. (2015) as a general reactor multiphysics solver applicable to MSRs and other reactor types. GeN-Foam features neutron diffusion, SP_3, and S_N neutronics models (Fiorina et al., 2016; Fiorina et al., 2015; Fiorina et al., 2019), and thermo-mechanical modeling for reactor expansion effects. Using GeN-Foam, (Altahhan et al., 2020) developed and optimized a liquid-fuel MSR design while Shi and Fratoni (2021) benchmarked precursor drift effects in an MSRE model.

Finally, within MOOSE, simulation tools capable of modeling MSRs include: Griffin (Wang et al., 2021); and Moltres (Lindsay, 2017)—the subject of this work. Griffin primarily tackles radiation transport problems, but the MOOSE framework facilitates multiphysics coupling with MOOSE-based applications for other physics such that all applications share the same data structure. This feature eliminates computational costs from external data transfers and optionally allows for *fully coupled* solves in which the application solves all physics simultaneously. Similarly, Moltres benefits from the highly-integrated cross-compatibility within the ecosystem of MOOSE-based applications. Abou-Jaoude et al. (2020) coupled Griffin with Pronghorn, another MOOSE-based application for advanced reactor thermal–hydraulics modeling, to demonstrate several steady-state MSR simulation capabilities defined in the CNRS benchmark. Lindsay et al. (2018) first demonstrated Moltres' MSR modeling capabilities on 2D axisymmetric and 3D Cartesian models of the MSRE with fixed velocity flow on a fully coupled neutronics and thermal–hydraulics solve. We later demonstrated some of Moltres' more recent developments through modeling a 2D axisymmetric model of the MSFR for steady-state operation and transient accident analysis (Park, 2020). The latter study introduced delayed neutron precursor (DNP) flow looping, DNP drift and temperature advection–diffusion coupling to incompressible flow, and decay heat modeling capabilities.

The mutual compatibility among different physics applications within the MOOSE framework simplifies the work required to strongly couple different physics together to solve novel multiphysics problems. For MSR simulations in Moltres such as those in this study, we coupled Moltres' MSR modeling capabilities with MOOSE's `Navier-Stokes` and `Heat Conduction` physics modules (Peterson et al., May 2018) for general thermal–hydraulics modeling.

This paper presents results of the CNRS benchmark from Moltres to verify its capabilities for modeling fast-spectrum MSRs. Moltres is an open-source multiphysics simulation software for advanced reactors. Making Moltres open-source promotes quality and participation through transparency and ease of peer review. The source code (Lindsay, 2017) is available on GitHub (GitHub, 2017). Moltres leverages `git` for version control, and integrated testing to protect existing capabilities while concurrently supporting continued code development. Moltres depends on the MOOSE finite element framework for its meshing and parallel, nonlinear Newton–Krylov solver capabilities. Therefore Moltres has access

to *fully coupled* methods with implicit time-stepping. *Full coupling* in the context of numerical methods refers to solving multiple equations simultaneously in a larger system of equations. In this study, we applied full coupling for the transient cases in Phase 2 of the CNRS benchmark.

Users also have the flexibility of separating the equations through *tight coupling* in which coupling is achieved through fixed-point iterations. For instance, users can employ full coupling to solve the multigroup neutron diffusion equations and the Navier–Stokes equations as two large, separate systems of equations to handle the strong coupling within those systems, thereby benefitting from solver stability and convergence over fewer iterations. These two systems can then be tightly coupled via outer fixed-point iterations to handle the relatively weak (but still objectively strong) coupling between the neutronics and thermal–hydraulics. Other than potentially lower compute times, this segregated approach allows users to adopt different mesh resolutions for the neutronics and thermal–hydraulics solvers. Alternatively, users can subcycle multiple timesteps of one system during one timestep of the other system. This flexibility helps users handle different spatial and temporal scales of various time-dependent phenomena in reactors. In this study, we applied tight coupling to run criticality calculations coupled with pseudo-transient calculations of the buoyancy-driven salt flow for the steady-state cases in Phase 1 of the CNRS benchmark.

For the reader's convenience, we wish to note that Moltres, GeN-Foam (Fiorina et al., 2015), and the PoliMi OpenFOAM solver (Aufiero et al., 2014) share several common characteristics as multiphysics solvers for reactor analysis. In terms of numerical methods, all three solvers implement general PDE discretization methods for solving deterministic models on unstructured meshes and highly scalable solution algorithms, though Moltres uses finite element method (FEM) and OpenFOAM uses finite volume method (FVM). They also adopt open-source philosophies and inherit `C++` object-oriented programming structures. Lastly, all three solvers benefit immensely from being a part of broader ecosystems of compatible tools, eliminating external data transfers and facilitating solver coupling.

2. CNRS Benchmark

The CNRS Benchmark (Tiberga et al., 2020) is a numerical benchmark for multiphysics software dedicated to modeling MSRs. It consists of three phases and eight steps in total. Each step is a well-defined subproblem for systematically assessing the capabilities of MSR software and pinpointing sources of discrepancies between software. Phase 0 consists of three single-physics problems in fluid dynamics, neutronics, and temperature, respectively. Phase 1 consists of four coupled steady-state problems. Lastly, Phase 2 consists of one coupled, time-dependent problem.

As shown in Fig. 1, the domain geometry is a 2 m×2 m square cavity filled with LiF–BeF$_2$-UF$_4$ molten salt at an initial temperature of 900 K (Tiberga et al., 2020). Standard vacuum boundary conditions apply for neutron flux along all boundaries whereby outgoing neutrons are considered lost, while homogeneous boundary conditions apply for delayed neutron precursors. No-slip boundary conditions apply for velocity variables in the cavity, except along the top boundary for Steps 0.1, 0.3, 1.1, 1.2, and 1.4, which impose forced flow in the form of lid-driven cavity flow. For the temperature variable, all boundaries are insulated, and we simulate salt cooling with the following volumetric heat sink equation:

$$q'''(\vec{r}) = \gamma(900 - T(\vec{r})) \tag{1}$$
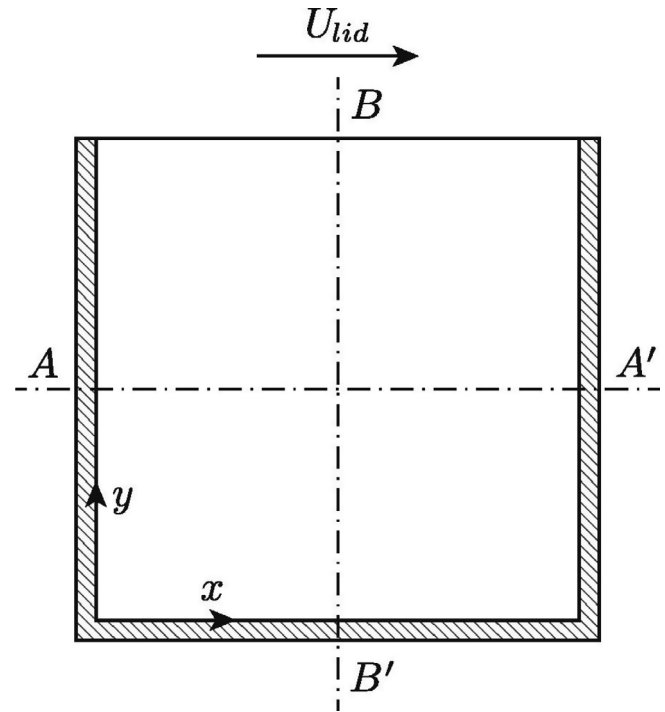
where

Fig. 1. 2 m×2 m 2D domain of the CNRS Benchmark. U_{lid} represents the velocity along the top boundary. Various quantities are measured along the centerlines AA' and BB' for comparison. From Tiberga et al. (2020).

$q''' =$ volumetric heat sink[W · m^{-3}],

$\gamma =$ heat transfer coefficient [W · m^{-3} · K^{-1}],

$T(\vec{r}) =$ temperature at point $\vec{r}$[K].

Tiberga et al. (2020) used Serpent 2 (Leppanen et al., Aug. 2014) with the JEFF-3.1 library (Koning et al., 2006) to generate multigroup neutronics data for the LiF–BeF$_2$-UF$_4$ salt in the domain at 900 K, which they condensed into six energy groups and eight precursor groups. We direct readers to their paper for the group constant data (Tiberga et al., 2020). In addition, the benchmark prescribes the following equations to govern the temperature dependence in the cross sections and the neutron diffusion coefficients:

$$\Sigma_i(T) = \Sigma_i(T_{ref}) \frac{\rho_{fuel}(T)}{\rho_{fuel}(T_{ref})} \tag{2}$$

and

$$D(T) = D(T_{ref}) \frac{\rho_{fuel}(T_{ref})}{\rho_{fuel}(T)} \tag{3}$$

where

$\Sigma_i =$ relevant macroscopic cross section[cm^{-1}],

$D =$ neutron diffusion coefficient [cm^2 · s^{-1}],

$\rho_{fuel} =$ density of the fuel salt[kg · m^{-3}],

$T_{ref} =$ reference temperature $= 900$K.

The benchmark also prescribes incompressible Navier–Stokes flow with the Boussinesq approximation for evaluating the salt flow in the domain but does not restrict the type of neutronics model. The following subsections briefly detail each benchmark step, with Table 1 listing the relevant input parameters and observables.

Table 1
Input parameters and observables of each benchmark step.

Step	Input parameters	Observables
0.1	• $U_{lid} = 0.5$ m·s^{-1}	• Velocity components (u_x, u_y) along AA' and BB'
0.2	• $U_{lid} = 0$ m·s^{-1} • $T = 900$ K • $P = 1$ GW	• Fission rate density $\sum_g^6 \Sigma_{f,g} \phi_g(\vec{r})$ along AA' • Reactivity ρ
0.3	• Fixed flow field from Step 0.1 for $U_{lid} = 0.5$ m·s^{-1} • Fixed heat source distribution $\sum_g^6 \epsilon_g \Sigma_{f,g} \phi_g(\vec{r})$ from Step 0.2 • $\gamma = 10^6$ W·m^{-3}·K^{-1}	• Temperature T along AA' and BB'
1.1	• Fixed flow field from Step 0.1 for $U_{lid} = 0.5$ m·s^{-1} • $T = 900$ K • $P = 1$ GW	• Delayed neutron source $\sum_i^8 \lambda_i C_i$ along AA' and BB' • Reactivity change between Step 1.1 and Step 0.2, $\Delta\rho = \rho - \rho_{s_{0.2}}$
1.2	• Fixed flow field from Step 0.1 for $U_{lid} = 0.5$ m·s^{-1} • $P = 1$ GW • $\gamma = 10^6$ W·m^{-3}·K^{-1}	• Temperature T along AA' and BB' • Reactivity change between Step 1.2 and Step 1.1, $\Delta\rho = \rho - \rho_{s_{1.1}}$ • Change in fission rate density $\sum_g^6 \Sigma_{f,g} \phi_g(\vec{r}) - \left[\sum_g^6 \Sigma_{f,g} \phi_g(\vec{r})\right]_{s_{0.2}}$
1.3	• $P = 1$ GW • $U_{lid} = 0$ m·s^{-1} • $\gamma = 10^6$ W·m^{-3}·K^{-1}	• Velocity components (u_x, u_y) along AA' and BB' • Temperature T along AA' and BB' • Delayed neutron source $\sum_i^8 \lambda_i C_i$ along AA' and BB' • Reactivity change from Step 0.2 $\Delta\rho = \rho - \rho_{s_{0.2}}$
1.4	• $\gamma = 10^6$ W·m^{-3}·K^{-1} • P variable in the range $[0,1]$ GW with a step of 0.2 GW • U_{lid} variable in the range $[0, 0.5]$ m·s^{-1} with a step of 0.1 m·s^{-1}	• Reactivity change between Step 1.4 and Step 0.2, $\Delta\rho = \rho - \rho_{s_{0.2}}$, for all permutations of P and U_{lid} values
2.1	• $\gamma = 10^6$ W·m^{-3}·K^{-1} • Steady-state solution from Step 1.4 for $U_{lid} = 0.5$ m·s^{-1} and $P = 1.0$ GW	• Power gain and shift as a function of the perturbation frequency

2.1. Phase 0: Single physics

In this preliminary phase, the steady-state solutions of individual physics are studied without any multiphysics coupling.

2.1.1. Step 0.1: Velocity field
This step investigates the steady-state incompressible flow distribution in the domain from a lid-driven cavity flow by imposing a non-zero horizontal velocity along the top boundary. In addition, this step provides a fixed velocity field for Steps 0.3, 1.1, and 1.2.

2.1.2. Step 0.2: Neutronics
This step tests neutronics capabilities through a criticality eigenvalue problem in a static, isothermal fuel configuration by solving for the fission rate density and effective multiplication factor k_{eff}. This step also aims to identify deviations in results attributable to differences in neutronics models and approximations. The total power P is fixed to normalize the neutron fluxes. For neutronics models conforming to the six neutron energy group structure provided by Tiberga et al. (2020), fission rate density is calculated as:

$$\text{Fission rate density} = \sum_g^6 \Sigma_{f,g} \phi_g(\vec{r}) \tag{4}$$

where

$\Sigma_{f,g}$ = macroscopic fission cross section for neutron in group g,

ϕ_g = neutron flux in group g.

2.1.3. Step 0.3: Temperature
This step assesses passive scalar transport capability for determining the temperature distribution independently from the fluid flow and neutronics problems by imposing fixed velocity and fission heat source distributions from Steps 0.1 and 0.2, respectively. Similar to the fission rate density, the heat source distribution in six-group neutronics models is calculated as:

$$\text{Heat source distribution} = \sum_g^6 \epsilon_g \Sigma_{f,g} \phi_g(\vec{r}) \tag{5}$$

where

ϵ_g = average fission energy released by neutrons in group g.

2.2. Phase 1: Steady-state coupling

Phase 1 builds towards simulating a fully-coupled multiphysics steady-state system by gradually introducing coupling between various physics present in a fast-spectrum molten salt system. All simulations are solved as steady-state criticality eigenvalue problems.

2.2.1. Step 1.1: Circulating fuel
This step investigates the effects of fuel salt flow on the neutronics, namely the reactivity loss from the movement of precursors. The delayed neutron precursors are allowed to drift under the fixed velocity field from Step 0.1 while keeping the temperature T fixed at 900 K. With eight precursor groups, the delayed neutron source is calculated as:

$$\text{Delayed neutron source} = \sum_i^8 \lambda_i C_i \tag{6}$$

where

λ_i = average decay constant of delayed neutron precursors in precursor group i,
C_i = concentration of delayed neutron precursors in precursor group i.

2.2.2. Step 1.2: Power coupling
This step assesses the capability to accurately reproduce the change in neutron flux distribution due to the fuel density reactivity feedback between the neutron fluxes and the temperature distribution. We solve for the steady-state neutron flux and temperature distributions under the fixed velocity field from Step 0.1 and a volumetric heat sink described by Eq. 1.

2.2.3. Step 1.3: Buoyancy
Building on the previous step, we replace the fixed velocity field with buoyancy-driven flow arising from the temperature gradients for a fully-coupled multiphysics problem without forced flow. Barring any significant discrepancies in the previous steps, this step assesses the capability to reproduce the correct buoyancy-driven flow profile and the subsequent effects on the neutronics and temperature distribution due to precursor drift and fuel density reactivity feedback.

2.2.4. Step 1.4: Full coupling
This step introduces forced flow to the fully-coupled problem through the non-zero U_{lid} boundary condition. Thus, this problem

most closely represents a molten salt system with 1) flow driven by an external force, 2) buoyancy flow effects, 3) DNP drift, and 4) thermal feedback effects on the neutronics. We solve for the k_{eff} under a range of U_{lid} and P values given in Table 1.

2.3. Phase 2: Time dependent coupling

In this phase, the transient response of the fully coupled nonlinear system is studied.

2.3.1. Step 2.1: Forced convection transient

Linear perturbation analyses are performed by introducing periodic perturbations to the heat transfer coefficient γ and studying the gain and phase shift of the response in the total power P. For the initial conditions, the steady-state solution from Step 1.4 with $U_{lid} = 0.5 \text{ m·s}^{-1}$ and $P = 1$ GW is used. This initial configuration is made exactly critical by scaling the neutron source terms, from fission and DNP decay, by the inverse of the criticality eigenvalue solution from Step 1.4.

γ is uniformly perturbed according to small-amplitude sine waves given as:

$$\gamma = \gamma_0[1 + 0.1 \sin(2\pi f)] \tag{7}$$

where

$\gamma_0 = 10^6 \text{W} \cdot \text{m}^{-3} \cdot \text{K}^{-1}$,
$f \in \{0.0125, 0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}\text{Hz}$.

The benchmark defines power gain as:

$$\text{Power gain} = \frac{(P_{max} - P_{avg})/P_{avg}}{(\gamma_{max} - \gamma_{avg})/\gamma_{avg}} \tag{8}$$

The subscripts denote the maximum and time-averaged values of P and γ.

3. Moltres

In this section, we describe Moltres (Lindsay et al., 2018), the multiphysics reactor simulation tool, and the specific modeling approach for simulating the CNRS Benchmark cases in Moltres. Much of Moltres' development focuses on meeting the needs of MSR multiphysics.

3.1. Description of Moltres

Moltres models coupled neutronics and thermal–hydraulics in reactors. While generally applicable to most reactor concepts, much of Moltres' development focuses on meeting the needs of MSR multiphysics. Together with MOOSE's (Permann et al., 2020) `Heat Conduction` and `Navier-Stokes` (Peterson et al., May 2018) modules, Moltres solves the multigroup neutron diffusion equations, for an arbitrary number of energy and precursor groups, and thermal–hydraulics equations simultaneously on the same mesh (or separately through fixed-point iterations if desired). Previously, Lindsay et al. (2018) demonstrated Moltres' MSR neutronics modeling capabilities with 1D salt flow in 2D-axisymmetric and 3D models of the MSRE. The neutron flux and temperature distributions agreed qualitatively with legacy MSRE data, albeit with some minor quantitative discrepancies due to simplifications and assumptions in the reactor geometry. Moltres has since undergone further development to support 1) the looping of DNP drift back into the reactor core, 2) coupling the aforementioned DNP drift to incompressible Navier–Stokes velocity flows within the reactor core, and 3) a decay heat model to simulate decay heat from fission products.

To perform neutronics calculations, Moltres requires homogenized group constant data from dedicated high-fidelity neutronics software such as the NEWT module in SCALE (DeHart et al., 2011), Serpent 2 (Leppanen et al., Aug. 2014), or OpenMC (Romano et al., 2015). Users can run a Python script in Moltres' Github repository, which automatically reads user-provided SCALE/Serpent 2/OpenMC output data files and creates Moltres-compatible JSON or text files containing all required group constant data.

We present the governing equations for the various physics models implemented in Moltres. Moltres solves for the neutron fluxes governed by the multigroup neutron diffusion equations given by:

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} = \nabla \cdot D_g \nabla \phi_g - \Sigma_g^r \phi_g + \sum_{g' \neq g}^{G} \Sigma_{g' \to g}^s \phi_{g'} + \chi_g^p \sum_{g'=1}^{G}(1-\beta)v\Sigma_{g'}^f \phi_{g'} + \chi_g^d \sum_i^I \lambda_i C_i \tag{9}$$

where

v_g = average speed of neutrons in group g,
ϕ_g = neutron flux in group g,
t = time,
D_g = diffusion coefficient of neutrons in group g,
Σ_g^r = macroscopic cross section for removal of neutrons from group g,
$\Sigma_{g' \to g}^s$ = macroscopic cross section of scattering from groups g' to g,
χ_g^p = prompt fission spectrum for neutrons in group g,
G = total number of discrete neutron groups,
v = average number of neutrons produced per fission,
Σ_g^f = macroscopic fission cross section for neutron in group g,
χ_g^d = delayed fission spectrum for neutrons in group g,
I = total number of delayed neutron precursor groups,
β = total delayed neutron fraction.

The delayed neutron precursor concentrations are governed in Moltres by the following equation:

$$\frac{\partial C_i}{\partial t} = \beta_i \sum_{g'=1}^{G} v\Sigma_{g'}^f \phi_{g'} - \lambda_i C_i - \vec{u} \cdot \nabla C_i + \nabla \cdot D_P \nabla C_i \tag{10}$$

where

β_i = delayed neutron fraction of precursor group i,
λ_i = average decay constant of delayed neutron precursors in precursor group i,
C_i = concentration of delayed neutron precursors in precursor group i,
$\vec{u}$ = molten salt flow velocity vector,
D_P = diffusion coefficient of the delayed neutron precursors.

The last two terms in Eq. 10 represent the advection and diffusion terms, respectively, to model the movement of DNP in liquid-fuel MSRs.

The governing equation for temperature in Moltres is an advection–diffusion equation with a fission heat source term Q_f given by:

$$\rho c_p \frac{\partial T}{\partial t} = -\rho c_p \vec{u} \cdot \nabla T + \nabla \cdot (k\nabla T) + Q_f \tag{11}$$

and

$$Q_f = \sum_{g=1}^{G} \epsilon_g \Sigma_g^f \phi_g \tag{12}$$

where

ρ = density of the molten salt,
c_p = specific heat capacity of molten salt,
T = temperature of molten salt,
k = effective thermal conductivity of molten salt,
Q_f = fission heat source,
ϵ_g = average fission energy released by neutrons in group g.

Lastly, the governing equations for the incompressible Navier–Stokes flow in Moltres are given by:

$$\rho \frac{\partial \vec{u}}{\partial t} = -\rho (\vec{u} \cdot \nabla)\vec{u} - \nabla p + \mu \nabla^2 \vec{u} + \rho \alpha \vec{g}(T - T_{\text{ref}}) \tag{13}$$

and

$$\nabla \cdot \vec{u} = 0 \tag{14}$$

where

p = pressure,
μ = dynamic viscosity,
α = coefficient of thermal expansion,
$\vec{g}$ = gravitational force vector,
T_{ref} = reference temperature at which the nominal density is provided.

The velocity, temperature, and delayed neutron precursor variables are all susceptible to numerical node-to-node oscillations near discontinuous boundary conditions commonly observed when resolving advection-dominated transport using continuous FEM (Kuhlmann et al., 2018). MOOSE's `Navier-Stokes` module provides the Streamline-Upwind Petrov–Galerkin (SUPG) stabilization scheme (Brooks et al., 1982) for the velocity and temperature variables to combat these oscillations. We refer readers to Peterson et al. (May 2018) for details on the implementation of these methods in the `Navier-Stokes` module. For the delayed neutron precursor variables, we discretized them using discontinuous shape functions supported by MOOSE's discontinuous Galerkin finite element method (DGFEM) solver. The cell-centered DGFEM scheme does not produce the aforementioned numerical oscillations observed with node-centered continuous FEM.

3.2. Modeling approach

[1] For this work, we ran the benchmark cases on a uniformly-spaced mesh consisting of 200×200 elements (0.01 m×0.01 m each). This mesh resolution was sufficient for mesh convergence as further refinement to 0.005 m×0.005 m produced a small 0.1 pcm increase in the reactivity for Step 0.2. We adopted the group constant data provided by Tiberga et al. (2020). Next, we discretized most of the relevant variables, i.e., neutron fluxes, velocity components, pressure, and temperature, using continuous, first-order Lagrange shape functions. The only exceptions are the precursor concentration variables, which we discretized using piecewise constant shape functions for the DGFEM solver mentioned in Section 3.1. In terms of solver accuracy, DGFEM with piecewise constant discretization is similar to first-order FVM because they share the same number of degrees of freedom. We interpolated the resulting discontinuous, cell-centered precursor values to obtain the nodal values for results analysis. Given the high Schmidt number (2×10^8) (Tiberga et al., 2020) of the salt, we neglected the precursor diffusion term in Eq. 10 as it has no observable effect on the distribution.

As mentioned in Section 3.1, the `Navier-Stokes` and `Heat Conduction` modules from MOOSE provide some of the capabilities for modeling incompressible flow and heat transfer. In particular, we stabilized the incompressible flow and temperature governing equations using the SUPG stabilization method implemented in MOOSE (Peterson et al., May 2018). Without SUPG stabilization, we observed spurious numerical oscillations in the velocity and temperature near the top boundary due to the singularity on the top left corner where different velocity boundary conditions meet. We also applied the Pressure-Stabilizing

Petrov–Galerkin (PSPG) stabilization scheme (Hughes et al., 1986) from the Navier–Stokes module (Peterson et al., May 2018) which enables equal-order discretizations in the velocity and pressure variables. Equal-order discretizations with PSPG are computationally cheaper and more convenient to work with than implementing higher-order velocity discretizations for stability without PSPG (Chapelle et al., 1993).

We performed all neutronics criticality calculations in Steps 0.2, 1.1, 1.2, 1.3, and 1.4 using the inverse power method solver in MOOSE and other calculations in Steps 0.1, 0.3, and 2.1 using the Preconditioned Newton–Krylov solver (Gaston et al., 2015). The coupled steady-state problems in Steps 1.2, 1.3, and 1.4 required segregated solvers due to the unique problem setups involving time-independent neutronics criticality calculations and pseudo-transient thermal–hydraulics calculations.

For the time-dependent cases in Step 2.1, we employed fully coupled solves with a second-order implicit Backward Differential Formula (BDF2) time-stepping scheme. We set the timestep sizes to 1/200th of the perturbation period for each driving frequency in the heat transfer coefficient. Table 2 shows the timestep sizes. We assumed the systems reached asymptotic behavior when the magnitudes of neighboring power peaks differed by less than 0.001% for at least ten wavelengths. Under this assumption, the phase shift measurements between adjacent waves always converged before the magnitude measurements of the power peaks.

Table 3 compares the numerical methods, meshing schemes, and neutronics models of Moltres and the four participating software packages in the CNRS benchmark paper (Tiberga et al., 2020). The SP_N and S_N neutronics models refer to the simplified P_N spherical harmonics and S_N discrete ordinates neutron transport models, respectively. Based on the solvers and methods of solution, Moltres is most similar to the PHANTOM-S_N + DGFlows (Tiberga et al., 2019) multiphysics package from TU Delft with the S_2 neutron transport model. Participants from CNRS and Paul Scherrer Institute (PSI) employed non-uniform meshes with smaller mesh elements near the boundaries, while we and the participants from PoliMi and TU Delft employed uniform meshes.

4. Results and Discussion

In this section, we compare the results from Moltres for each CNRS Benchmark step to the results in the benchmark paper (Tiberga et al., 2020). The software packages from CNRS and TU Delft each report two sets of results arising from different angular discretizations in their neutronics models for Steps 0.2, 1.1, 1.2, 1.3, 1.4, and 2.1. We labeled these sets of results as CNRS-SP_1 and CNRS-SP_3; and TUD-S_2 and TUD-S_6, respectively. The authors performed code-to-code verification by sampling observable values at 201 equidistant points along the centerlines AA' and BB' and reporting the discrepancy ϵ_c of each observable from each software (indexed by c) for each measured observable Q_c (not to be confused with fission heat source Q_f), relative to the average of that same observable Q_{avg} from all participating software. Variables ϵ_c and Q_{avg} are calculated as:

$$\epsilon_c = \sqrt{\frac{\sum_{i=1}^{N_p} \left[Q_c(\vec{r}_i) - Q_{avg}(\vec{r}_i)\right]^2}{\sum_{i=1}^{N_p} Q_{avg}^2(\vec{r}_i)}} \tag{15}$$

and

$$Q_{avg}(\vec{r}_i) = \frac{1}{N_c} \sum_{c=1}^{N_c} Q_c(\vec{r}_i) \tag{16}$$

---

[1] The input files for all benchmark cases are available on the Moltres GitHub repository at https://github.com/arfc/moltres/tree/devel/problems/2021-cnrs-benchmark.

Table 2
Timestep sizes used for the time-dependent cases in Step 2.1, corresponding to 1/200th of the perturbation period.

Frequency [Hz]	0.0125	0.025	0.05	0.1	0.2	0.4	0.8
Timestep size [s]	0.2	0.2	0.1	0.05	0.025	0.0125	0.00625

Table 3
List of software packages and their corresponding model specifications for the CNRS Benchmark simulations (Tiberga et al., 2020).

Software	Institute	Numerical method	Mesh	Neutronics model
OpenFOAM	Centre national de la recherche scientifique (CNRS)	Finite volume	200×200; Non-uniform	SP_1 & SP_3
OpenFOAM	Politecnico di Milano (PoliMi)	Finite volume	400×400; Uniform	Neutron diffusion
GeN-Foam	Paul Scherrer Institute (PSI)	Finite volume	200×200; Non-uniform	Neutron diffusion
PHANTOM-S_N+DGFlows	Delft University of Technology (TUD)	Discontinuous finite element	50×50; Uniform	S_2 & S_6
Moltres (This work)	University of Illinois at Urbana-Champaign (UIUC)	Continuous & discontinuous finite element	200×200; Uniform	Neutron diffusion

where

$Q_c(\vec{r_i})$ = value of observable Q at location r_i from software c,
N_p = number of sampling points of quantity $Q = 201$,
N_c = number of participating software packages.

The average discrepancy ϵ over all simulation tools is calculated as:

$$\epsilon = \frac{1}{N_c} \sum_{c=1}^{N_c} \epsilon_c \tag{17}$$

We adopted the averaged values ϵ and Q_{avg} directly from the reference work (Tiberga et al., 2020) without including our results in the calculations. We note that the benchmark does not provide a reference solution, and a significantly erroneous value from one of the software packages could heavily skew the discrepancy values. Nevertheless, the benchmark paper reports good agreement among their software packages.

For observables measured along the centerlines AA' and/or BB', Tables 4 and 6 report the discrepancy ϵ_c of each observable from Moltres relative to the average of the benchmark participants Q_{avg} alongside the average discrepancy ϵ of the benchmark participants. We also reproduce corresponding plots in the benchmark paper for every observable along AA' or BB' in Figs. 2, 3, 4, 6, 7, 8, and 10 for a qualitative comparison of the results from Moltres and the benchmark participants. Given the significant overlap in the plot curves, these figures omit results from CNRS-SP_1 and TUD-S_2 to reduce cluttering. Readers may also refer to A for tables of observable values at nine equidistant points along AA' and BB' from Moltres and the benchmark participants. We provide these tables for ease of review and a direct comparison to corresponding data tables from (Tiberga et al., 2020). The full dataset of all observable results used in this results analysis is available at (Park et al., 2021). Lastly, Table 5 reports all reactivity and change in reactivity results from Steps 0.2, 1.1, 1.2, and 1.3.

4.1. Phase 0 results & discussion

Figs. 2–4 show that Moltres accurately reproduced all three sets of results in Phase 0 for the velocity field, fission rate density, and temperature. Table 4 reports the discrepancy values from Moltres for Phase 0 and the corresponding average and standard deviation (SD) of the discrepancy values from the benchmark participants (Tiberga et al., 2020). Moltres performs very well as most discrep-ancy values are either lower than or fall within one SD of the benchmark average discrepancies. The discrepancy value for T along centerline BB' in Step 0.3 is the only exception, with its value of 0.164% being larger than the benchmark average by three SD.

We note in Fig. 4 that the T distribution from Moltres is almost identical to the corresponding distributions from CNRS-SP_3 and TUD-S_6 along most of centerline BB'. However, Fig. 5 shows a significant spread in the T distributions along BB' from all software packages near the top boundary. At $y = 2.0$ m, Moltres underpredicts the temperature at 912.3 K compared to the benchmark participants' values which range between 930.3 K and 948.1 K (Refer to Table 12 for the numerical values). This point on the top boundary lies directly downstream of the velocity boundary condition discontinuity at the top-left corner. Corner singularities are generally tricky to approximate with continuous Galerkin methods (Kuhlmann et al., 2018). The SUPG stabilization scheme dampens numerical oscillations by introducing pointwise artificial thermal diffusivity, which depends strongly on the inverse of local velocity magnitude (Peterson et al., May 2018). Therefore, while the SUPG scheme was very effective in eliminating spurious numerical oscillations everywhere else, it provides little damping along the top boundary due to the relatively large non-zero velocity boundary condition. On the other hand, the temperature values in the rest of the domain and the average discrepancies of the other variables show that Moltres can still accurately reproduce the expected results, and the temperature deviations along the top boundary do not impact the overall integrity of our results.

Lastly, we observe in Table 5 that the reactivity ρ value of 465.6 pcm from Moltres falls well within the range of ρ values from the benchmark which ranges from 353.7 pcm up to 578.1 pcm. Given that Moltres adopts the neutron diffusion model, our ρ value agrees closest to the results from the software packages, which also adopt the neutron diffusion model or theoretically-equivalent models such as the SP_1 and S_2 neutron transport models, namely CNRS-SP_1, PoliMi, PSI, and TUD-S_2.
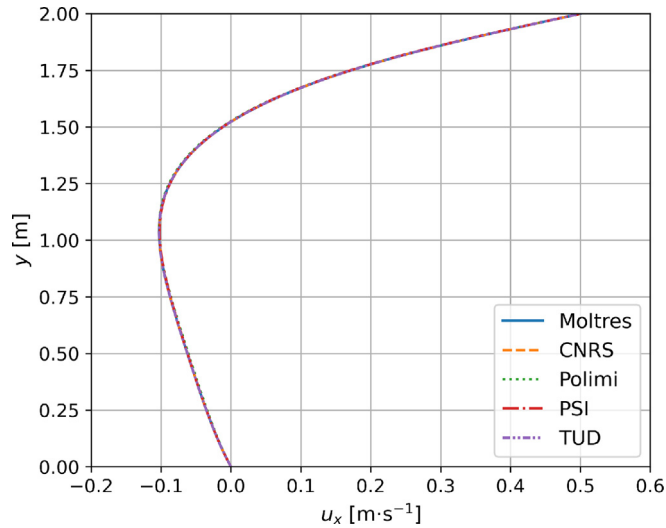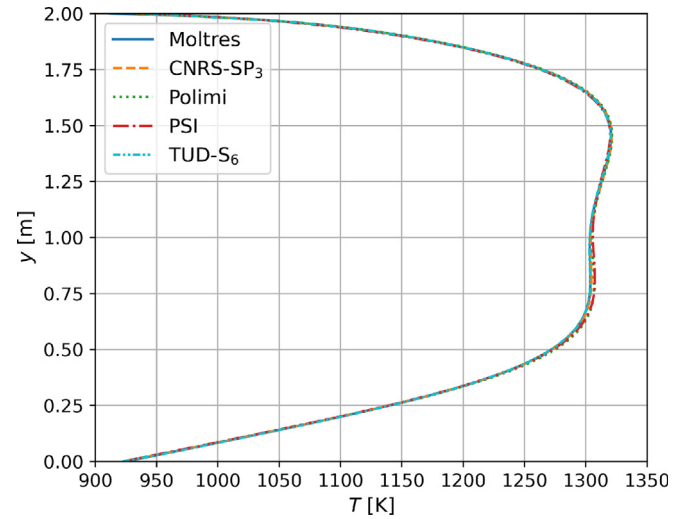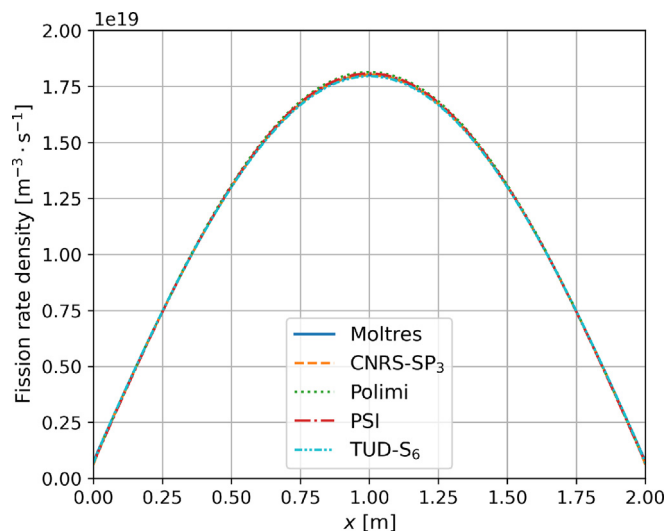
4.2. Phase 1 results & discussion

Table 6 shows the discrepancy values from Moltres relative to the average and SD of the benchmark participants for Steps 1.1, 1.2, and 1.3, and the corresponding average discrepancy values from the benchmark (Tiberga et al., 2020). The subsequent subsections discuss the results for each benchmark step in Phase 1.

2

Fig. 6. Step 1.1 – Delayed neutron source along AA' (top) and BB' (bottom).

Fig. 7. Step 1.2 – Temperature distribution and change in fission rate density along AA'.

mark value of 0.080% compared to the corresponding temperature discrepancy values for Step 0.3 and 1.2.

In Table 5, we observe that the change in ρ from Moltres (-1207.7 pcm) falls within the range of reported benchmark values and matches the software from TUD-S_2 (-1208.5 pcm) the closest. This agreement is likely due to the similar solvers (diffusion and S_2 neutronics models) and methods of solution (finite element method on uniform meshes) employed by our respective software.

4.2.4. Step 1.4: Full coupling

Fig. 9 shows 2D temperature distribution and velocity streamlines from Moltres for Step 1.4 with $U_{lid} = 0.5$ m·s^{-1} and $P = 1$ GW. Table 7 shows the change in ρ under the various U_{lid} and P values. We refer readers to Tiberga et al.'s paper (Tiberga et al., 2020) for the benchmark participants' corresponding values. The change in ρ values from Moltres all fall within the range of benchmark values for all cases. Furthermore, the $\Delta\rho$ values are all within 1.1 pcm of the corresponding values from the TUD-S_2 model in the benchmark paper. Given that the S_2 discrete ordinates method is theoretically equivalent to the multigroup neutron diffusion method, this indicates that Moltres is largely consistent with the benchmark participants outside of differences from the neutronics models.
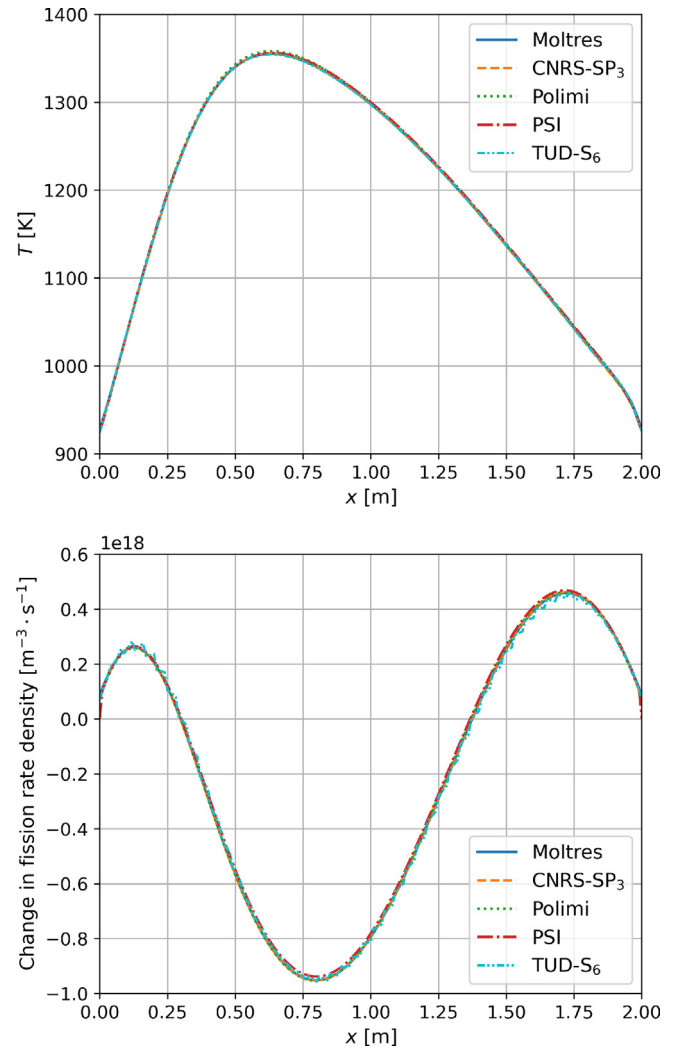
4.3. Phase 2 results & discussion

Lastly, the following subsection discusses the results for the transient cases in Step 2.1, which involve measuring the response in power output to periodic perturbations in the heat transfer coefficient.

4.3.1. Step 2.1: Forced convection transient

Fig. 10 shows the Bode gain and phase shift plots of the response in power output in the fully coupled system. Along with the average discrepancy values from Table 8, the results show that Moltres is consistent with the benchmark. The gain data points from all MSR software agree closely with one another. Moltres reports an average discrepancy value of 0.496%, slightly lower than the benchmark average of 0.587%. On the other hand, the phase shift data points show a greater spread over the various driving frequencies. We note the different time-stepping schemes and time-step sizes among the different software packages, which are likely responsible for the variations in the phase shift. Even with a precision of $\pm 0.9°$ for each phase shift value, Moltres accurately reproduces the correct trend with a lower average discrepancy (1.741%) than the benchmark participants' average (2.176%).
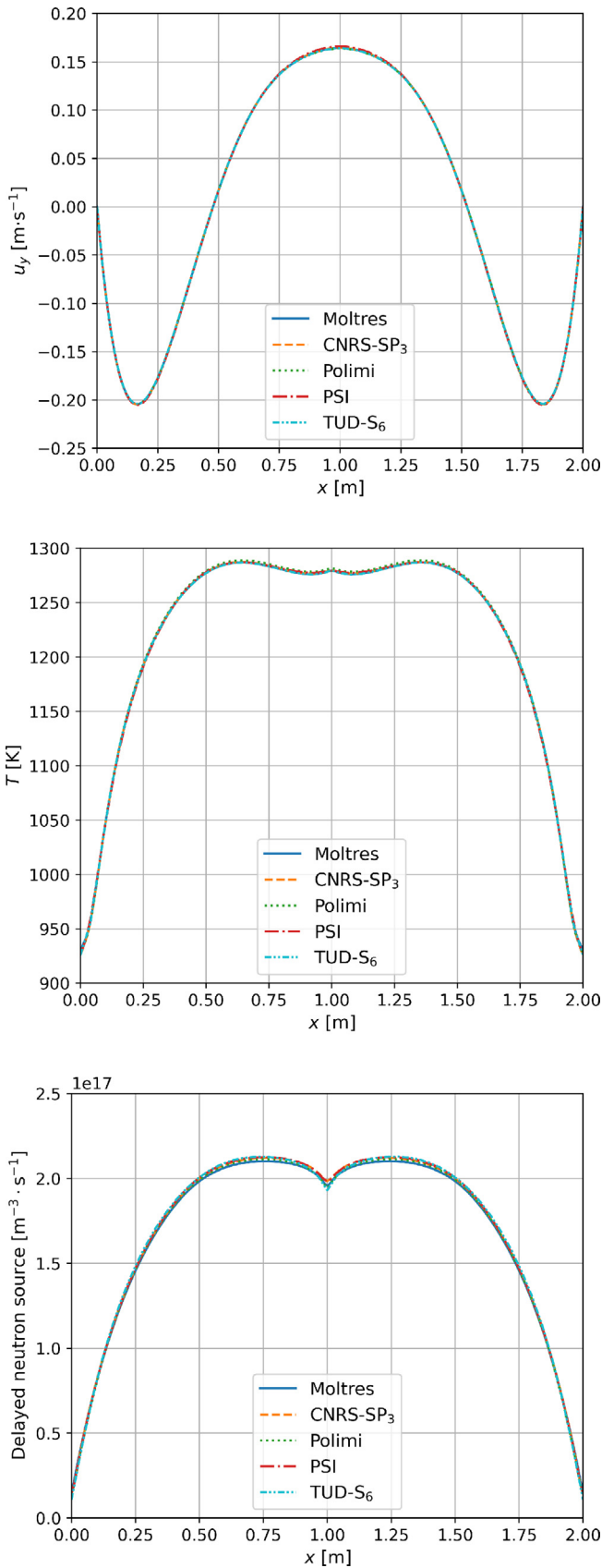
Fig. 10. Step 2.1 – Bode gain and phase plots of the frequency response of the fully coupled system.

Fig. 8. Step 1.3 – Vertical velocity component, temperature distribution, and delayed neutron source along AA'.

Table 5
Reactivity ρ and change in reactivity $(\rho_a - \rho_b)$ values from Steps 0.2, 1.1, 1.2, and 1.3. All units are in pcm.

| Software	Step 0.2	Step 1.1	Step 1.2	Step 1.3
	$\rho_{s_{0.2}}$	$\rho_{s_{1.1}} - \rho_{s_{0.2}}$	$\rho_{s_{1.2}} - \rho_{s_{1.1}}$	$\rho_{s_{1.3}} - \rho_{s_{0.2}}$
Moltres	465.6	−62.7	−1142.2	−1207.7
CNRS-SP_1	411.3	−62.5	−1152.0	−1220.5
CNRS-SP_3	353.7	−62.6	−1152.7	−1220.7
PoliMi	421.2	−62.0	−1161.0	−1227.0
PSI	411.7	−63.0	−1154.8	−1219.6
TUD-S_2	482.6	−62.0	−1145.2	−1208.5
TUD-S_6	578.1	−60.7	−1122.0	−1184.4

### 4.4. Computational performance

We ran all of the simulations on Cray XE nodes on the Blue Waters supercomputer. Each XE node comprises two AMD Opteron^{TM} 6276 processors, for a total of 32 CPU cores per node, rated at a maximum clock speed of 3.2 GHz. Lindsay et al. (2018) previously reported good scaling performance of Moltres on the Blue Waters system.

Table 6
Discrepancy values from Moltres alongside the average and standard deviation of the discrepancy values of the benchmark participants for Phase 1.

Step	Observable	Centerline	Moltres [%]	Benchmark [%]	
				Average	SD
1.1	$\sum_i \lambda_i C_i$	AA'	0.603	0.346	0.166
		BB'	0.327	0.294	0.153
1.2	T	AA'	0.076	0.095	0.015
		BB'	0.179	0.089	0.012
	$\Delta\left[\sum_g^6 \Sigma_{f,g}\phi_g(\vec{r})\right]_{s_{1.2}-s_{0.2}}$	AA'	1.110	1.576	0.564
		BB'	1.089	1.133	0.392
1.3	u_x	AA'	0.123	0.691	0.566
	u_y	AA'	0.237	0.329	0.131
		BB'	0.238	0.356	0.217
	T	AA'	0.064	0.057	0.023
		BB'	0.070	0.080	0.024
	$\sum_i \lambda_i C_i$	AA'	1.043	0.460	0.190
		BB'	0.462	1.194	0.178
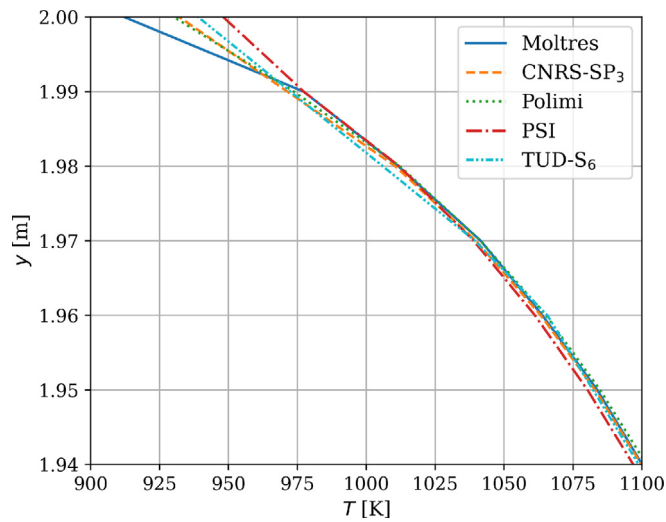
Fig. 5. Step 0.3 – Temperature distribution along BB' for y = 1.94 m to y = 2.00 m.
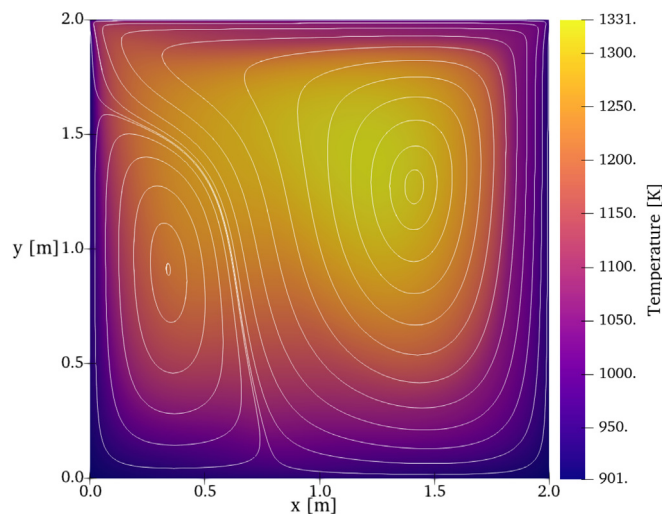
Fig. 9. Temperature distribution from Moltres for the fully coupled system (Step 1.4) with buoyancy effects, $P = 1$ GW, and $U_{lid} = 0.5$ m·s^{-1}. The lines correspond to the streamlines of the velocity field.

Table 7
Reactivity change in Step 1.4, relative to Step 0.2 under various U_{lid} and P values.

	$\rho_{s1.4} - \rho_{s0.2}$ [pcm]				
P [GW]					
U_{lid} [m·s^{-1}]	0.2	0.4	0.6	0.8	1.0
0.0	−263.7	−498.3	−730.9	−966.7	−1207.7
0.1	−265.9	−498.7	−730.6	−966.0	−1206.7
0.2	−268.1	−498.8	−729.4	−963.7	−1203.6
0.3	−269.9	−498.5	−727.8	−960.8	−1199.5
0.4	−271.9	−498.5	−726.5	−958.3	−1195.7
0.5	−274.2	−498.7	−725.6	−956.4	−1192.7

Table 8
Discrepancy values from Moltres alongside the average and standard deviation of the discrepancy values of the benchmark participants for Step 2.1.

Step	Observable	Moltres [%]	Benchmark [%]	
			Average	SD
2.1	Gain	0.493	0.587	0.244
	Phase shift	1.741	2.176	0.554

Table 9
Compute times required for one perturbation cycle in Step 2.1 for $f = 0.0125$ Hz and 0.8 Hz on 16 XE nodes (512 CPU cores).

f [Hz]	0.0125	0.8
Compute time [hours]	6.29	0.98

While the simulations for Phases 0 and 1 were not computationally intensive, some required more memory than the 8 GB to 16 GB typically available on most personal computers. Table 9 shows the compute times required per perturbation cycle of the highest and lowest perturbation frequencies for Step 2.1 on 16 XE nodes. The compute times of all other simulations in Step 2.1 fall between 0.98 and 6.29 h. The simulations with smaller perturbation frequencies required longer compute times because the larger timestep sizes led to greater changes in the variables per timestep. The compute times are comparable to the compute times for the CNRS-SP_1 and CNRS-SP_3 models on 20 Intel® Xeon® Gold 5118 processors reported in (Blanco et al., Jan. 2021; Blanco, 2020).

5. Conclusions

MSRs feature significant multiphysics interactions which present computational challenges for many existing multiphysics reactor analysis software. This paper presents code-to-code

verification of Moltres capabilities in modeling such multiphysics phenomena in fast-spectrum MSRs based on the CNRS benchmark (Tiberga et al., 2020). The CNRS benchmark assesses multiphysics MSR simulation software through several steps involving single-physics and coupled neutronics/thermal–hydraulics problems.

The results showed that Moltres is consistent with the participating software presented in the CNRS benchmark paper for the modeling of important phenomena in fast-spectrum MSRs. The percentage discrepancies in the various neutronics, velocity, and temperature quantities mostly fall below or within one standard deviation of the average of the benchmark participants. Minor deviations in the temperature in Steps 0.3 and 1.2 stem from the discontinuous velocity boundaries on the top corners in the lid-driven cavity flow. We showed that these deviations are limited to the top boundary of the domain and do not affect the rest of the physical parameters. The results from Moltres agree closest with the TUD-S_2 software package, which implements the S_2 discrete ordinates method for neutron transport on a uniform structured mesh with a DGFEM-based solver. These features make Moltres the most similar to the TUD-S_2 model compared to the other models that employ different neutron transport models, non-uniform meshes, and/or finite volume-based solvers.

This work verifies Moltres' capabilities for future work involving modeling and simulation of fast-spectrum MSRs. Fast-spectrum MSRs under consideration for modeling with Moltres include the European MSFR as a continuation of work done by Park (2020), and TerraPower's Molten Chloride Fast Reactor (MCFR) (TerraPower, 2021) from publicly available design specifications. Moltres can play an important role in supporting further MSR development through enabling transient accident safety analysis and design optimization studies on an open-source platform. An ongoing research project involves employing Moltres as a surrogate model for machine learning-based reactor design optimization. We note that Moltres also supports modeling solid-fueled reactors such as the High Temperature Gas-Cooled Reactor (HTGR) by disabling the precursor drift functionality as demonstrated by Fairhurst-Agosta (2020). Future work pertaining to further Moltres development include introducing an intermediate-fidelity turbulence model for highly turbulent flows in MSRs, improving neutronics accuracy in heterogeneous geometries, and enhancing the general computational performance of existing features.

CRediT authorship contribution statement

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Appendix A. Additional data tables

This Appendix presents observable values measured at nine equidistant points along the centerlines AA' and BB' from Moltres and the CNRS benchmark participants (Tiberga et al., 2020). We refer readers to Park et al., 2021 and Tiberga et al., 2019 for the full set of results from Moltres and the benchmark participants.

Table 10
Step 0.1 – Velocity components along centerlines AA' and BB'.

Observable	Code	Results along AA' (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
u_x (m s^{-1})	Moltres	0.000E+00	−1.923E-02	−5.372E-02	−8.371E-02	−1.025E-01	−1.043E-01	−7.975E-02	−3.080E-02	0.000E+00
	CNRS	0.000E+00	−1.924E-02	−5.372E-02	−8.369E-02	−1.025E-01	−1.043E-01	−7.972E-02	−3.080E-02	0.000E+00
	PoliMi	0.000E+00	−1.922E-02	−5.365E-02	−8.357E-02	−1.023E-01	−1.041E-01	−7.947E-02	−3.066E-02	0.000E+00
	PSI	0.000E+00	−1.929E-02	−5.366E-02	−8.332E-02	−1.018E-01	−1.034E-01	−7.912E-02	−3.072E-02	0.000E+00
	TUD	1.002E-06	−1.922E-02	−5.372E-02	−8.371E-02	−1.025E-01	−1.044E-01	−7.977E-02	−3.081E-02	4.198E-06
u_y (m s^{-1})	Moltres	0.000E+00	7.269E-02	8.579E-02	6.087E-02	1.250E-02	−4.794E-02	−9.612E-02	−8.722E-02	0.000E+00
	CNRS	0.000E+00	7.266E-02	8.575E-02	6.084E-02	1.251E-02	−4.789E-02	−9.606E-02	−8.722E-02	0.000E+00
	PoliMi	0.000E+00	7.139E-02	8.433E-02	6.007E-02	1.269E-02	−4.691E-02	−9.472E-02	−8.621E-02	0.000E+00
	PSI	0.000E+00	7.265E-02	8.534E-02	6.021E-02	1.230E-02	−4.734E-02	−9.536E-02	−8.720E-02	0.000E+00
	TUD	5.877E-06	7.269E-02	8.580E-02	6.089E-02	1.252E-02	−4.794E-02	−9.613E-02	−8.726E-02	−1.013E-05
Observable	Code	Results along BB' (point coordinates are expressed in m)								
		(1,0)	(1,0.25)	(1,0.5)	(1,0.75)	(1,1)	(1,1.25)	(1,1.5)	(1,1.75)	(1,2)
u_x (m s^{-1})	Moltres	0.000E+00	−3.518E-02	−6.243E-02	−8.723E-02	−1.025E-01	−8.770E-02	−1.146E-02	1.718E-01	5.000E-01
	CNRS	0.000E+00	−3.517E-02	−6.242E-02	−8.720E-02	−1.025E-01	−8.766E-02	−1.147E-02	1.717E-01	5.000E-01
	PoliMi	0.000E+00	−3.423E-02	−6.107E-02	−8.613E-02	−1.023E-01	−8.861E-02	−1.299E-02	1.706E-01	5.000E-01
	PSI	0.000E+00	−3.511E-02	−6.217E-02	−8.667E-02	−1.018E-01	−8.731E-02	−1.191E-02	1.705E-01	5.000E-01
	TUD	1.494E-06	−3.519E-02	−6.244E-02	−8.724E-02	−1.025E-01	−8.770E-02	−1.146E-02	1.718E-01	5.000E-01
u_y (m s^{-1})	Moltres	0.000E+00	5.161E-05	6.166E-04	3.842E-03	1.250E-02	2.525E-02	3.050E-02	1.501E-02	0.000E+00
	CNRS	0.000E+00	5.641E-05	6.309E-04	3.862E-03	1.251E-02	2.524E-02	3.048E-02	1.500E-02	0.000E+00
	PoliMi	0.000E+00	9.118E-05	7.484E-04	4.046E-03	1.269E-02	2.534E-02	3.050E-02	1.500E-02	0.000E+00
	PSI	0.000E+00	7.727E-05	6.822E-04	3.875E-03	1.230E-02	2.472E-02	2.994E-02	1.481E-02	0.000E+00
	TUD	1.501E-06	5.260E-05	6.209E-04	3.853E-03	1.252E-02	2.528E-02	3.053E-02	1.502E-02	7.987E-06

Table 11
Step 0.2 – Fission rate density along AA*ı*.

Observable	Code	Results along AA*ı* (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
$\int_E \Sigma_f \Phi dE$ (m^{-3}s^{-1})	Moltres	7.701E+17	7.461E+18	1.303E+19	1.672E+19	1.801E+19	1.672E+19	1.303E+19	7.461E+18	7.701E+17
	CNRS-SP_1	6.896E+17	7.436E+18	1.305E+19	1.678E+19	1.809E+19	1.678E+19	1.305E+19	7.436E+18	6.896E+17
	CNRS-SP_3	6.206E+17	7.450E+18	1.303E+19	1.673E+19	1.802E+19	1.673E+19	1.303E+19	7.450E+18	6.206E+17
	PoliMi	7.780E+17	7.470E+18	1.310E+19	1.684E+19	1.815E+19	1.684E+19	1.310E+19	7.470E+18	7.780E+17
	PSI	8.622E+17	7.436E+18	1.305E+19	1.678E+19	1.809E+19	1.678E+19	1.305E+19	7.436E+18	8.622E+17
	TUD-S_2	6.626E+17	7.433E+18	1.307E+19	1.682E+19	1.814E+19	1.682E+19	1.307E+19	7.433E+18	6.626E+17
	TUD-S_6	6.833E+17	7.463E+18	1.300E+19	1.667E+19	1.796E+19	1.667E+19	1.300E+19	7.463E+18	6.833E+17

Table 12
Step 0.3 – Temperature distribution along centerlines AA*ı* and BB*ı*.

Observable	Code	Results along AA*ı* (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
T (K)	Moltres	9.251E+02	1.194E+03	1.357E+03	1.361E+03	1.303E+03	1.224E+03	1.131E+03	1.035E+03	9.251E+02
	CNRS-SP_1	9.253E+02	1.194E+03	1.358E+03	1.363E+03	1.305E+03	1.224E+03	1.131E+03	1.034E+03	9.251E+02
	CNRS-SP_3	9.236E+02	1.194E+03	1.357E+03	1.361E+03	1.304E+03	1.224E+03	1.131E+03	1.034E+03	9.235E+02
	PoliMi	9.253E+02	1.196E+03	1.361E+03	1.364E+03	1.305E+03	1.224E+03	1.132E+03	1.035E+03	9.252E+02
	PSI	9.253E+02	1.196E+03	1.356E+03	1.363E+03	1.306E+03	1.226E+03	1.133E+03	1.037E+03	9.252E+02
	TUD-S_2	9.212E+02	1.194E+03	1.359E+03	1.364E+03	1.305E+03	1.224E+03	1.131E+03	1.032E+03	9.225E+02
	TUD-S_6	9.219E+02	1.194E+03	1.356E+03	1.360E+03	1.303E+03	1.223E+03	1.131E+03	1.034E+03	9.233E+02
Observable	**Code**	**Results along BB*ı* (point coordinates are expressed in m)**								
		(1,0)	(1,0.25)	(1,0.5)	(1,0.75)	(1,1)	(1,1.25)	(1,1.5)	(1,1.75)	(1,2)
T (K)	Moltres	9.251E+02	1.140E+03	1.272E+03	1.303E+03	1.303E+03	1.313E+03	1.320E+03	1.264E+03	9.123E+02
	CNRS-SP_1	9.252E+02	1.139E+03	1.273E+03	1.305E+03	1.305E+03	1.314E+03	1.321E+03	1.265E+03	9.322E+02
	CNRS-SP_3	9.236E+02	1.140E+03	1.272E+03	1.304E+03	1.304E+03	1.313E+03	1.320E+03	1.265E+03	9.322E+02
	PoliMi	9.253E+02	1.140E+03	1.275E+03	1.307E+03	1.305E+03	1.313E+03	1.321E+03	1.265E+03	9.303E+02
	PSI	9.252E+02	1.139E+03	1.273E+03	1.307E+03	1.306E+03	1.312E+03	1.319E+03	1.263E+03	9.481E+02
	TUD-S_2	9.215E+02	1.139E+03	1.273E+03	1.305E+03	1.305E+03	1.315E+03	1.322E+03	1.265E+03	9.374E+02
	TUD-S_6	9.222E+02	1.140E+03	1.272E+03	1.303E+03	1.303E+03	1.312E+03	1.319E+03	1.264E+03	9.390E+02

Table 13
Step 1.1 – Delayed neutron source along centerlines AA*ı* and BB*ı*.

Observable	Code	Results along AA*ı* (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
$\sum_i \lambda_i C_i$ (m^{-3}s^{-1})	Moltres	1.338E+16	1.456E+17	2.213E+17	2.412E+17	2.268E+17	1.923E+17	1.463E+17	9.273E+16	1.196E+16
	CNRS-SP_1	1.335E+16	1.452E+17	2.212E+17	2.411E+17	2.268E+17	1.923E+17	1.461E+17	9.214E+16	1.316E+16
	CNRS-SP_3	1.251E+16	1.454E+17	2.209E+17	2.406E+17	2.264E+17	1.921E+17	1.462E+17	9.245E+16	1.233E+16
	PoliMi	1.321E+16	1.450E+17	2.219E+17	2.414E+17	2.266E+17	1.920E+17	1.459E+17	9.188E+16	1.292E+16
	PSI	1.325E+16	1.453E+17	2.214E+17	2.413E+17	2.270E+17	1.925E+17	1.463E+17	9.218E+16	1.314E+16
	TUD-S_2	1.093E+16	1.438E+17	2.228E+17	2.426E+17	2.278E+17	1.927E+17	1.464E+17	8.968E+16	1.184E+16
	TUD-S_6	1.132E+16	1.437E+17	2.212E+17	2.405E+17	2.261E+17	1.916E+17	1.461E+17	9.029E+16	1.224E+16
Observable	**Code**	**Results along BB*ı* (point coordinates are expressed in m)**								
		(1,0)	(1,0.25)	(1,0.5)	(1,0.75)	(1,1)	(1,1.25)	(1,1.5)	(1,1.75)	(1,2)
$\sum_i \lambda_i C_i$ (m^{-3}s^{-1})	Moltres	1.296E+16	1.199E+17	1.881E+17	2.191E+17	2.268E+17	2.264E+17	2.183E+17	1.760E+17	2.827E+16
	CNRS-SP_1	1.306E+16	1.190E+17	1.881E+17	2.193E+17	2.268E+17	2.261E+17	2.178E+17	1.754E+17	3.079E+16
	CNRS-SP_3	1.222E+16	1.193E+17	1.879E+17	2.189E+17	2.264E+17	2.257E+17	2.175E+17	1.753E+17	3.072E+16
	PoliMi	1.297E+16	1.186E+17	1.881E+17	2.194E+17	2.266E+17	2.260E+17	2.177E+17	1.756E+17	2.805E+16
	PSI	1.299E+16	1.189E+17	1.881E+17	2.195E+17	2.270E+17	2.261E+17	2.176E+17	1.752E+17	2.730E+16
	TUD-S_2	1.109E+16	1.174E+17	1.882E+17	2.203E+17	2.278E+17	2.281E+17	2.193E+17	1.768E+17	2.655E+16
	TUD-S_6	1.143E+16	1.178E+17	1.872E+17	2.186E+17	2.261E+17	2.264E+17	2.179E+17	1.761E+17	2.728E+16

Table 14
Step 1.2 – Temperature distribution and change in fission rate density relative to Step 0.2 along centerlines $AA\prime$ and $BB\prime$.

Observable	Code	Results along $AA\prime$ (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
T (K)	Moltres	9.279E+02	1.196E+03	1.341E+03	1.347E+03	1.298E+03	1.225E+03	1.137E+03	1.043E+03	9.279E+02
	CNRS-SP_1	9.280E+02	1.195E+03	1.341E+03	1.349E+03	1.298E+03	1.225E+03	1.136E+03	1.041E+03	9.278E+02
	CNRS-SP_3	9.262E+02	1.195E+03	1.341E+03	1.348E+03	1.298E+03	1.225E+03	1.137E+03	1.042E+03	9.260E+02
	PoliMi	9.281E+02	1.198E+03	1.343E+03	1.350E+03	1.300E+03	1.226E+03	1.138E+03	1.045E+03	9.280E+02
	PSI	9.282E+02	1.197E+03	1.340E+03	1.349E+03	1.300E+03	1.227E+03	1.139E+03	1.045E+03	9.280E+02
	TUD-S_2	9.235E+02	1.196E+03	1.343E+03	1.350E+03	1.300E+03	1.226E+03	1.137E+03	1.041E+03	9.250E+02
	TUD-S_6	9.243E+02	1.196E+03	1.340E+03	1.347E+03	1.298E+03	1.225E+03	1.137E+03	1.042E+03	9.258E+02
$\sum_g^6 \Sigma_{f,g}\phi_g(\vec{r}) - \left[\sum_g^6 \Sigma_{f,g}\phi_g(\vec{r})\right]_{S_{0.2}}$ (m s^{-1})	Moltres	8.640E+16	1.148E+17	−5.668E+17	−9.410E+17	−7.942E+17	−2.895E+17	2.540E+17	4.549E+17	8.604E+16
	CNRS-SP_1	7.800E+16	1.169E+17	−5.694E+17	−9.490E+17	−7.979E+17	−2.853E+17	2.626E+17	4.611E+17	7.776E+16
	CNRS-SP_3	7.016E+16	1.156E+17	−5.668E+17	−9.427E+17	−7.941E+17	−2.869E+17	2.575E+17	4.578E+17	6.993E+16
	PoliMi	7.556E+16	1.141E+17	−5.682E+17	−9.440E+17	−7.906E+17	−2.800E+17	2.628E+17	4.563E+17	7.522E+16
	PSI	2.188E+15	1.226E+17	−5.486E+17	−9.275E+17	−7.807E+17	−2.725E+17	2.702E+17	4.632E+17	1.835E+15
	TUD-S_2	7.203E+16	1.069E+17	−5.648E+17	−9.534E+17	−8.186E+17	−2.724E+17	2.628E+17	4.531E+17	7.194E+16
	TUD-S_6	7.304E+16	1.041E+17	−5.572E+17	−9.382E+17	−8.098E+17	−2.791E+17	2.466E+17	4.431E+17	7.274E+16

Observable	Code	Results along $BB\prime$ (point coordinates are expressed in m)								
		(1,0)	(1,0.25)	(1,0.5)	(1,0.75)	(1,1)	(1,1.25)	(1,1.5)	(1,1.75)	(1,2)
T (K)	Moltres	9.280E+02	1.149E+03	1.272E+03	1.300E+03	1.298E+03	1.304E+03	1.307E+03	1.253E+03	9.147E+02
	CNRS-SP_1	9.281E+02	1.148E+03	1.272E+03	1.301E+03	1.298E+03	1.304E+03	1.307E+03	1.253E+03	9.350E+02
	CNRS-SP_3	9.262E+02	1.149E+03	1.272E+03	1.300E+03	1.298E+03	1.303E+03	1.306E+03	1.253E+03	9.351E+02
	PoliMi	9.281E+02	1.150E+03	1.275E+03	1.304E+03	1.300E+03	1.304E+03	1.307E+03	1.253E+03	9.470E+02
	PSI	9.282E+02	1.148E+03	1.273E+03	1.303E+03	1.300E+03	1.303E+03	1.306E+03	1.252E+03	9.517E+02
	TUD-S_2	9.240E+02	1.148E+03	1.274E+03	1.302E+03	1.300E+03	1.306E+03	1.309E+03	1.254E+03	9.424E+02
	TUD-S_6	9.247E+02	1.149E+03	1.272E+03	1.300E+03	1.298E+03	1.303E+03	1.306E+03	1.253E+03	9.442E+02
$\sum_g^6 \Sigma_{f,g}\phi_g(\vec{r}) - \left[\sum_g^6 \Sigma_{f,g}\phi_g(\vec{r})\right]_{S_{0.2}}$ (m s^{-1})	Moltres	9.077E+16	2.601E+17	−2.295E+17	−6.406E+17	−7.942E+17	−7.522E+17	−4.901E+17	−3.940E+16	9.584E+16
	CNRS-SP_1	8.205E+16	2.640E+17	−2.264E+17	−6.431E+17	−7.979E+17	−7.557E+17	−4.923E+17	−3.953E+16	8.353E+16
	CNRS-SP_3	7.377E+16	2.615E+17	−2.273E+17	−6.407E+17	−7.941E+17	−7.525E+17	−4.902E+17	−3.919E+16	7.493E+16
	PoliMi	7.961E+16	2.639E+17	−2.256E+17	−6.389E+17	−7.942E+17	−7.536E+17	−4.922E+17	−4.164E+16	8.050E+16
	PSI	9.174E+16	2.639E+17	−2.287E+17	−6.506E+17	−8.020E+17	−7.479E+17	−4.856E+17	−3.479E+16	9.059E+16
	TUD-S_2	7.595E+16	2.513E+17	−2.336E+17	−6.589E+17	−8.186E+17	−7.703E+17	−4.964E+17	−4.656E+16	6.931E+16
	TUD-S_6	7.677E+16	2.436E+17	−2.373E+17	−6.540E+17	−8.098E+17	−7.612E+17	−4.906E+17	−4.534E+16	6.990E+16

Table 15

Step 1.3 – Velocity components, temperature distribution, and delayed neutron source along centerlines $AA\prime$ and $BB\prime$.

Observable	Code	Results along $AA\prime$ (point coordinates are expressed in m)								
		(0,1)	(0.25,1)	(0.5,1)	(0.75,1)	(1,1)	(1.25,1)	(1.5,1)	(1.75,1)	(2,1)
u_x (m s^{-1})	Moltres	0.000E+00	1.635E+00	2.300E+00	1.570E+00	−1.487E-08	−1.570E+00	−2.300E+00	−1.635E+00	0.000E+00
	CNRS-SP_1	0.000E+00	1.641E-02	2.310E-02	1.579E-02	1.250E-09	−1.579E-02	−2.310E-02	−1.641E-02	0.000E+00
	CNRS-SP_3	0.000E+00	1.636E-02	2.302E-02	1.572E-02	1.000E-09	−1.572E-02	−2.302E-02	−1.636E-02	0.000E+00
	PoliMi	0.000E+00	1.637E-02	2.312E-02	1.578E-02	−1.125E-10	−1.578E-02	−2.312E-02	−1.637E-02	0.000E+00
	PSI	0.000E+00	1.630E-02	2.263E-02	1.519E-02	−8.525E-09	−1.519E-02	−2.263E-02	−1.630E-02	0.000E+00
	TUD-S_2	6.054E-06	1.644E-02	2.316E-02	1.584E-02	−2.218E-06	−1.584E-02	−2.316E-02	−1.644E-02	−6.054E-06
	TUD-S_6	5.982E-06	1.631E-02	2.295E-02	1.566E-02	−2.196E-06	−1.566E-02	−2.295E-02	−1.631E-02	−5.982E-06
u_y (m s^{-1})	Moltres	0.000E+00	−1.769E+01	1.698E+00	1.371E+01	1.643E+01	1.371E+01	1.698E+00	−1.769E+01	0.000E+00
	CNRS-SP_1	0.000E+00	−1.777E-01	1.721E-02	1.376E-01	1.649E-01	1.376E-01	1.721E-02	−1.777E-01	0.000E+00
	CNRS-SP_3	0.000E+00	−1.771E-01	1.708E-02	1.372E-01	1.645E-01	1.372E-01	1.708E-02	−1.771E-01	0.000E+00
	PoliMi	0.000E+00	−1.767E-01	1.741E-02	1.368E-01	1.638E-01	1.368E-01	1.741E-02	−1.767E-01	0.000E+00
	PSI	0.000E+00	−1.779E-01	1.662E-02	1.376E-01	1.659E-01	1.376E-01	1.662E-02	−1.779E-01	0.000E+00
	TUD-S_2	−2.886E-05	−1.780E-01	1.735E-02	1.379E-01	1.650E-01	1.379E-01	1.735E-02	−1.780E-01	−2.886E-05
	TUD-S_6	−2.929E-05	−1.766E-01	1.694E-02	1.368E-01	1.639E-01	1.368E-01	1.694E-02	−1.766E-01	−2.929E-05
T (K)	Moltres	9.284E+02	1.192E+03	1.277E+03	1.284E+03	1.280E+03	1.284E+03	1.277E+03	1.192E+03	9.284E+02
	CNRS-SP_1	9.279E+02	1.193E+03	1.278E+03	1.284E+03	1.280E+03	1.284E+03	1.278E+03	1.193E+03	9.279E+02
	CNRS-SP_3	9.260E+02	1.193E+03	1.278E+03	1.284E+03	1.280E+03	1.284E+03	1.278E+03	1.193E+03	9.260E+02
	PoliMi	9.279E+02	1.193E+03	1.279E+03	1.286E+03	1.282E+03	1.286E+03	1.279E+03	1.193E+03	9.279E+02
	PSI	9.279E+02	1.191E+03	1.278E+03	1.284E+03	1.280E+03	1.284E+03	1.278E+03	1.191E+03	9.279E+02
	TUD-S_2	9.248E+02	1.193E+03	1.279E+03	1.285E+03	1.281E+03	1.285E+03	1.279E+03	1.193E+03	9.248E+02
	TUD-S_6	9.257E+02	1.193E+03	1.277E+03	1.283E+03	1.280E+03	1.283E+03	1.277E+03	1.192E+03	9.257E+02
$\sum_i \lambda_i C_i$ (m^{-3}s^{-1})	Moltres	1.443E+16	1.457E+17	1.984E+17	2.102E+17	1.959E+17	2.102E+17	1.984E+17	1.457E+17	1.443E+16
	CNRS-SP_1	1.499E+16	1.468E+17	2.001E+17	2.123E+17	1.988E+17	2.123E+17	2.001E+17	1.468E+17	1.499E+16
	CNRS-SP_3	1.409E+16	1.469E+17	2.000E+17	2.121E+17	1.986E+17	2.121E+17	2.000E+17	1.469E+17	1.409E+16
	PoliMi	1.436E+16	1.464E+17	1.992E+17	2.113E+17	1.944E+17	2.113E+17	1.992E+17	1.464E+17	1.436E+16
	PSI	1.477E+16	1.469E+17	2.005E+17	2.126E+17	1.983E+17	2.126E+17	2.005E+17	1.469E+17	1.477E+16
	TUD-S_2	1.041E+16	1.488E+17	2.017E+17	2.141E+17	1.942E+17	2.141E+17	2.017E+17	1.488E+17	1.041E+16
	TUD-S_6	1.086E+16	1.485E+17	2.006E+17	2.128E+17	1.931E+17	2.128E+17	2.006E+17	1.485E+17	1.086E+16

Observable	Code	Results along $BB\prime$ (point coordinates are expressed in m)								
		(1,0)	(1,0.25)	(1,0.5)	(1,0.75)	(1,1)	(1,1.25)	(1,1.5)	(1,1.75)	(1,2)
u_y (m s^{-1})	Moltres	0.000E+00	−1.511E-07	−1.568E-07	−1.023E-07	−1.487E-08	8.013E-08	1.580E-07	1.791E-07	0.000E+00
	CNRS-SP_1	0.000E+00	3.512E-02	8.947E-02	1.359E-01	1.649E-01	1.665E-01	1.307E-01	5.756E-02	0.000E+00
	CNRS-SP_3	0.000E+00	3.510E-02	8.933E-02	1.356E-01	1.645E-01	1.660E-01	1.303E-01	5.740E-02	0.000E+00
	PoliMi	0.000E+00	3.507E-02	8.909E-02	1.351E-01	1.638E-01	1.656E-01	1.302E-01	5.743E-02	0.000E+00
	PSI	0.000E+00	3.537E-02	9.055E-02	1.374E-01	1.659E-01	1.669E-01	1.309E-01	5.780E-02	0.000E+00
	TUD-S_2	−1.234E-05	3.510E-02	8.950E-02	1.360E-01	1.650E-01	1.667E-01	1.308E-01	5.763E-02	−2.808E-05
	TUD-S_6	−1.327E-05	3.506E-02	8.912E-02	1.352E-01	1.639E-01	1.655E-01	1.299E-01	5.719E-02	−2.872E-05
T (K)	Moltres	0.000E+00	3.512E+00	8.928E+00	1.355E+01	1.643E+01	1.658E+01	1.302E+01	5.734E+00	0.000E+00
	CNRS-SP_1	9.280E+02	1.067E+03	1.156E+03	1.226E+03	1.280E+03	1.315E+03	1.326E+03	1.283E+03	9.284E+02
	CNRS-SP_3	9.261E+02	1.067E+03	1.156E+03	1.226E+03	1.280E+03	1.315E+03	1.325E+03	1.282E+03	9.266E+02
	PoliMi	9.280E+02	1.067E+03	1.157E+03	1.228E+03	1.282E+03	1.317E+03	1.327E+03	1.284E+03	9.282E+02
	PSI	9.281E+02	1.068E+03	1.156E+03	1.226E+03	1.280E+03	1.314E+03	1.324E+03	1.281E+03	9.287E+02
	TUD-S_2	9.250E+02	1.066E+03	1.156E+03	1.227E+03	1.281E+03	1.316E+03	1.327E+03	1.283E+03	9.137E+02
	TUD-S_6	9.258E+02	1.069E+03	1.157E+03	1.226E+03	1.280E+03	1.314E+03	1.325E+03	1.282E+03	9.149E+02
$\sum_i \lambda_i C_i$ (m^{-3}s^{-1})	Moltres	9.279E+02	1.068E+03	1.156E+03	1.226E+03	1.280E+03	1.314E+03	1.325E+03	1.282E+03	9.283E+02
	CNRS-SP_1	1.479E+16	8.797E+16	1.377E+17	1.746E+17	1.988E+17	2.087E+17	2.017E+17	1.659E+17	1.721E+16
	CNRS-SP_3	1.383E+16	8.839E+16	1.378E+17	1.746E+17	1.986E+17	2.084E+17	2.015E+17	1.659E+17	1.631E+16
	PoliMi	1.451E+16	8.624E+16	1.344E+17	1.706E+17	1.944E+17	2.045E+17	1.977E+17	1.623E+17	1.583E+16
	PSI	1.477E+16	8.782E+16	1.373E+17	1.741E+17	1.983E+17	2.083E+17	2.014E+17	1.658E+17	1.689E+16
	TUD-S_2	1.337E+16	8.510E+16	1.335E+17	1.700E+17	1.942E+17	2.047E+17	1.980E+17	1.621E+17	1.243E+16
	TUD-S_6	1.378E+16	8.605E+16	1.335E+17	1.693E+17	1.931E+17	2.034E+17	1.969E+17	1.616E+17	1.300E+16

References

Abou-Jaoude, A., Hermosillo, A., Martin, N., Schunert, S., Wang, Y., Balestra, P., Jan. 2020. Coupled Multiphysics Simulation of Pool-Type Molten Salt Reactors Using Griffin/Pronghorn. pp. 1587–1590..

Altahhan, M.R., Bhaskar, S., Ziyad, D., Balestra, P., Fiorina, C., Hou, J., Smith, N., Avramova, M., 2020. Preliminary design and analysis of Liquid Fuel Molten Salt Reactor using multi-physics code GeN-Foam. Nucl. Eng. Des. 369,. URL:http://www.sciencedirect.com/science/article/pii/S0029549320303204 110826.

Aufiero, M., Rubiolo, P., 2018. Testing and Verification of Multiphysics Tools for Fast-Spectrum MSRs: The CNRS Benchmark. In: Transactions of the American Nuclear Society. Vol. 118 of Reactor Physics: General–I. American Nuclear Society, Philadelphia, Pennsylvania, pp. 837–840. URL:http://ansannual.org/wp-content/2018/Data/pdfs/382-25331.pdf.

Aufiero, M., Cammi, A., Geoffroy, O., Losa, M., Luzzi, L., Ricotti, M.E., Rouch, H., 2014. Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis. Chem. Eng. Sci. 111, 390–401. URL:http://www.sciencedirect.com/science/article/pii/S0009250914001146.

Blanco, J.A., Dec. 2020. Neutronic, thermohydraulic and thermomechanical coupling for the modeling of criticality accidents in nuclear systems. phdthesis, Université Grenoble Alpes [2020-....]. URL:https://tel.archives-ouvertes.fr/tel-03172978..

Blanco, J., Rubiolo, P., Dumonteil, E., Jan. 2021. Neutronic modeling strategies for a liquid fuel transient calculation. EPJ Web of Conferences 247, 06013.

Boer, B., Lathouwers, D., Kloosterman, J.L., Hagen, T.H.J.J.V.D., Strydom, G., May 2010. Validation of the DALTON-THERMIX Code System with Transient Analyses of the HTR-10 and Application to the PBMR. Nuclear Technology 170 (2), 306–321, publisher: Taylor & Francis _eprint: doi: 10.13182/NT10-A9485. URL:https://doi.org/10.13182/NT10-A9485..

Brooks, A.N., Hughes, T.J.R., 1982. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Eng. 32 (1), 199–259. URL:https://www.sciencedirect.com/science/article/pii/0045782582900718.

Cammi, A., Di Marcello, V., Luzzi, L., Memoli, V., Ricotti, M.E., 2011. A multi-physics modelling approach to the dynamics of Molten Salt Reactors. Ann. Nucl. Energy 38 (6), 1356–1372. URL:http://www.sciencedirect.com/science/article/pii/S0306454911000582.

Cammi, A., Fiorina, C., Guerrieri, C., Luzzi, L., 2012. Dimensional effects in the modelling of MSR dynamics: Moving on from simplified schemes of analysis to a multi-physics modelling approach. Nucl. Eng. Des. 246, 12–26. URL:http://www.sciencedirect.com/science/article/pii/S0029549311006273.

Cervi, E., Lorenzi, S., Cammi, A., Luzzi, L., 2019. Development of a multiphysics model for the study of fuel compressibility effects in the Molten Salt Fast Reactor. Chem. Eng. Sci. 193, 379–393. URL:http://www.sciencedirect.com/science/article/pii/S0009250918306742.

Cervi, E., Lorenzi, S., Cammi, A., Luzzi, L., 2019. Development of an SP3 neutron transport solver for the analysis of the Molten Salt Fast Reactor. Nucl. Eng. Des. 346, 209–219. URL:http://www.sciencedirect.com/science/article/pii/S0029549319300354.

Chapelle, D., Bathe, K.J., 1993. The inf-sup test. Computers Struct. 47 (4), 537–545. URL:https://www.sciencedirect.com/science/article/pii/004579499390340J.

COMSOL AB, 2021 COMSOL Multiphysics. URL:www.comsol.com..

CORDIS, 2021 Severe Accident Modeling and Safety Assessment for Fluid-fuel Energy Reactors — SAMOSAFER Project — H2020 — CORDIS — European Commission. URL:https://cordis.europa.eu/project/id/847527..

Dai, Z., 2017. Thorium molten salt reactor nuclear energy system (TMSR). In: Dolan, T.J. (Ed.), Molten Salt Reactors and Thorium Energy, 17. Woodhead Publishing, pp. 531–540. URL:https://www.sciencedirect.com/science/article/pii/B9780081011263000178.

DeHart, M.D., Bowman, S.M., 2011. Reactor Physics Methods and Analysis Capabilities in SCALE. Nucl. Technol. 174 (2), 196–213. URL:http://epubs.ans.org/?a=11720.

de Zwaan, S.J., Boer, B., Lathouwers, D., Kloosterman, J.L., 2007. Static design of a liquid-salt-cooled pebble bed reactor (LSPBR). Ann. Nucl. Energy 34 (1), 83–92. URL:https://www.sciencedirect.com/science/article/pii/S0306454906002295.

DOE, D. o. E., Jan. 2021. Office of Nuclear Energy Strategic Vision. Tech. rep., Office of Nuclear Energy. URL:https://www.energy.gov/sites/prod/files/2021/01/f82/DOE-NE Strategic Vision -Web - 01.08.2021.pdf..

Downar, T., Xu, Y., Seker, V., 2010. PARCS v3.0 U.S. NRC Core Neutronics Simulator. In: Theory Manual. University of Michigan, Ann Arbor, MI.

Fairhurst-Agosta, R., Dec. 2020. Multi-Physics and Technical Analysis of High-Temperature Gas-Cooled Reactors for Hydrogen Production. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL..

Fiorina, C., Lathouwers, D., Aufiero, M., Cammi, A., Guerrieri, C., Kloosterman, J.L., Luzzi, L., Ricotti, M.E., 2014. Modelling and analysis of the MSFR transient behaviour. Ann. Nucl. Energy 64 (Supplement C), 485–498. URL:http://www.sciencedirect.com/science/article/pii/S0306454913004118.

Fiorina, C., Clifford, I., Aufiero, M., Mikityuk, K., 2015. GeN-Foam: a novel OpenFOAM<error l="207" c="Undefined command "/> based multi-physics solver for 2D/3D transient analysis of nuclear reactors. Nucl. Eng. Des. 294, 24–37. URL:https://linkinghub.elsevier.com/retrieve/pii/S0029549315003829.

Fiorina, C., Kerkar, N., Mikityuk, K., Rubiolo, P., Pautz, A., 2016. Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform. Ann. Nucl. Energy 96, 212–222. URL:https://linkinghub.elsevier.com/retrieve/pii/S0306454916303309.

Fiorina, C., Radman, S., Koc, M.-Z., Pautz, A., Dec. 2019. DETAILED MODELLING OF THE EXPANSION REACTIVITY FEEDBACK IN FAST REACTORS USING OpenFOAM..

Gaston, D.R., Permann, C.J., Peterson, J.W., Slaughter, A.E., Andrš, D., Wang, Y., Short, M.P., Perez, D.M., Tonks, M.R., Ortensi, J., Zou, L., Martineau, R.C., 2015. Physics-based multiscale coupling for full core nuclear reactor simulation. Ann. Nucl. Energy 84, 45–54. URL:http://www.sciencedirect.com/science/article/pii/S030645491400543X.

GitHub, 2017. Build software better, together. URL:https://github.com..

Hughes, T.J.R., Franca, L.P., Balestra, M., 1986. A new finite element formulation for computational fluid dynamics: V. Circumventing the babuška-brezzi condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations. Comput. Methods Appl. Mech. Eng. 59 (1), 85–99. URL:https://www.sciencedirect.com/science/article/pii/0045782586900253.

Koning, A., Forrest, R., Kellett, M., Mills, R., Henriksson, H., Rugama, Y., 2006. The JEFF-3.1 nuclear data library: JEFF report 21. Tech. Rep. 21, OECD Nuclear Energy Agency, oCLC: 1039431841..

Krepel, J., Grundmann, U., Rohde, U., Weiss, F.-P., 2005. DYN1D-MSR dynamics code for molten salt reactors. Ann. Nucl. Energy 32 (17), 1799–1824. URL:http://www.sciencedirect.com/science/article/pii/S0306454905001702.

Krepel, J., Rohde, U., Grundmann, U., Weiss, F.-P., 2007. DYN3D-MSR spatial dynamics code for molten salt reactors. Ann. Nucl. Energy 34 (6), 449–462. URL:http://www.sciencedirect.com/science/article/pii/S0306454907000527.

Kuhlmann, H., Romanò, F., Apr. 2018. The Lid-Driven Cavity. In: Computational Methods in Applied Sciences. Journal Abbreviation: Computational Methods in Applied Sciences..

Laureau, A., Heuer, D., Merle-Lucotte, E., Rubiolo, P.R., Allibert, M., Aufiero, M., 2017. Transient coupled calculations of the Molten Salt Fast Reactor using the Transient Fission Matrix approach. Nucl. Eng. Des. 316 (Supplement C), 112–124, jC0004. URL:http://www.sciencedirect.com/science/article/pii/S002954931730081X.

Lecarpentier, D., Carpentier, V., Jan. 2003. A Neutronic Program for Critical and Nonequilibrium Study of Mobile Fuel Reactors: The Cinsf1D Code. Nucl. Sci. Eng. 143 (1), 33–46. https://doi.org/10.13182/NSE03-A2316.

Leppanen, J., 2013. Development of a dynamic simulation mode in serpent 2 Monte Carlo Code, 11..

Leppanen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., Aug. 2014. The Serpent Monte Carlo code: Status, development and applications in 2013. Ann. Nucl. Energy 82, 142–150.

Li, R., Wang, S., Rineiski, A., Zhang, D., Merle-Lucotte, E., 2015. Transient analyses for a molten salt fast reactor with optimized core geometry. Nucl. Eng. Des. 292, 164–176. URL:http://www.sciencedirect.com/science/article/pii/S0029549315002617.

Lindsay, A., 2017. Moltres, software for simulating Molten Salt Reactors. Https://github.com/arfc/moltres. URL:https://github.com/arfc/moltres.

Lindsay, A., Ridley, G., Rykhlevskii, A., Huff, K., 2018. Introduction to Moltres: An application for simulation of Molten Salt Reactors. Ann. Nucl. Energy 114, 530–540. URL:https://linkinghub.elsevier.com/retrieve/pii/S0306454917304760.

The OpenFOAM Foundation Ltd, 2021. OpenFOAM. URL:https://openfoam.org/.

Nicolino, C., Lapenta, G., Dulla, S., Ravetto, P., 2008. Coupled dynamics in the physics of molten salt reactors. Ann. Nucl. Energy 35 (2), 314–322. URL:http://www.sciencedirect.com/science/article/pii/S0306454907001454.

NRC, 2007. TRACE V5.0 User's Manual. Tech. rep. URL:https://www.nrc.gov/docs/ML1200/ML120060239.pdf.

Park, S.M., Aug. 2020. Advancement and Verification of Moltres for Molten Salt Reactor Safety Analysis. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL. URL:https://www.ideals.illinois.edu/handle/2142/108542.

Park, S.M., Munk, M., Huff, K.D., Sep. 2021. Results from Moltres for the CNRS Benchmark. Type: dataset. URL:https://zenodo.org/record/5534964.

Permann, C.J., Gaston, D.R., Andrš, D., Carlsen, R.W., Kong, F., Lindsay, A.D., Miller, J.M., Peterson, J.W., Slaughter, A.E., Stogner, R.H., Martineau, R.C., 2020. MOOSE: Enabling massively parallel multiphysics simulation. SoftwareX 11,. URL:https://www.sciencedirect.com/science/article/pii/S2352711019302973 100430.

Peterson, J., Lindsay, A., Kong, F., May 2018. Overview of the Incompressible Navier-Stokes simulation capabilities in the MOOSE Framework. Adv. Eng. Softw. 119, 68–92.

Pettersen, E.E., Mikityuk, K., 2016. Coupled multi-physics simulations of the Molten Salt Fast Reactor using coarse-mesh thermal-hydraulics and spatial neutronics. Master's thesis, MSc thesis, September 2016 (PDF). URL:http://samofar.eu/wp-content/uploads/2016/11/MScThesis-eirikEidePettersen.pdf..

Rineiski, A., Sinitsa, V., Maschek, W., Wang, S., 2005. Kinetics and cross-section developments for analyses of reactor transmutation concepts with SIMMER. In: Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications. American Nuclear Society, Palais des Papes, Avignon, France.

Romano, P.K., Horelik, N.E., Herman, B.R., Nelson, A.G., Forget, B., Smith, K., 2015. OpenMC: A state-of-the-art Monte Carlo code for research and development. Ann. Nucl. Energy 82 (Supplement C), 90–97. URL:http://www.sciencedirect.com/science/article/pii/S030645491400379X.

Shi, J., Fratoni, M., 2021. Gen-foam Model and Benchmark of Delayed Neutron Precursor Drift in the Molten Salt Reactor Experiment. EPJ Web of Conferences 247, 06040, jC0007. URL:https://www.epj-conferences.org/articles/epjconf/abs/2021/01/epjconf_physor2020_06040/epjconf_physor2020_06040.html.

TerraPower, Apr. 2021. TerraPower Molten Chloride Fast Reactor Technology Fact Sheet. URL:https://www.terrapower.com/wp-content/uploads/2021/04/TP_2021_MCFR_Technology.pdf..

Tiberga, M., De Oliveira, R.G.G., Cervi, E., Blanco, J.A., Lorenzi, S., Aufiero, M., Lathouwers, D., Rubiolo, P., Dec. 2019. Results of the CNRS benchmark. Type: dataset. URL:https://zenodo.org/record/3648639.

Tiberga, M., Lathouwers, D., Kloosterman, J., Aug. 2019. A DISCONTINUOUS GALERKIN FEM MULTI-PHYSICS SOLVER FOR THE MOLTEN SALT FAST REACTOR. In: Proceedings of the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2019). American Nuclear Society, Portland, OR..

Tiberga, M., de Oliveira, R.G.G., Cervi, E., Blanco, J.A., Lorenzi, S., Aufiero, M., Lathouwers, D., Rubiolo, P., 2020. Results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors. Ann. Nucl. Energy 142,. URL:http://www.sciencedirect.com/science/article/pii/S0306454920301262 107428.

Wang, Y., Schunert, S., Ortensi, J., Laboure, V., DeHart, M., Prince, Z., Kong, F., Harter, J., Balestra, P., Gleicher, F., Apr. 2021. Rattlesnake: A MOOSE-Based Multiphysics Multischeme Radiation Transport Application. Nuclear Technology 0 (0), 1–26, publisher: Taylor & Francis _eprint: doi: 10.1080/00295450.2020.1843348. doi: 10.1080/00295450.2020.1843348..

Zanetti, M., Cammi, A., Fiorina, C., Luzzi, L., 2015. A Geometric Multiscale modelling approach to the analysis of MSR plant dynamics. Prog. Nucl. Energy 83 (Supplement C), 82–98. URL:https://www.sciencedirect.com/science/article/pii/S0149197015000487.

Zhang, D.L., Qiu, S.Z., Su, G.H., Liu, C.L., 2009. Development of a steady state analysis code for a molten salt reactor. Ann. Nucl. Energy 36 (5), 590–603. URL:http://www.sciencedirect.com/science/article/pii/S030645490900019X.