NEAMS TH CRAB

September 2021

Guillaume Giudicelli[1], Alexander Lindsay[1], Ramiro Freile[2], and Jieun Lee[2]

[1]*Computational Frameworks, Idaho National Laboratory, Idaho Falls, Idaho 83415*
[2]*Texas A&M University, 423 Spence St, College Station, TX 77843*

Idaho National Laboratory

NEAMS TH CRAB

Guillaume Giudicelli[1], Alexander Lindsay[1], Ramiro Freile[2], and Jieun Lee[2]

[1]Computational Frameworks, Idaho National Laboratory, Idaho Falls, Idaho 83415
[2]Texas A&M University, 423 Spence St, College Station, TX 77843

September 2021

Idaho National Laboratory
Computational Frameworks
Idaho Falls, Idaho 83415

http://www.inl.gov

Page intentionally left blank

ACRONYMS

BEA Battelle Energy Alliance

DOE U.S. Department of Energy

FHR fluoride salt-cooled high temperature reactor

FLiBe Beryllium Lithium Fluoride salt

INS incompressible Navier-Stokes

INSFV incompressible Navier-Stokes finite volume

MMS Method of Manufactured Solutions

MOOSE Multiphysics Object Oriented Simulation Environment

MSFR Molten Salt Fast Reactor

NEAMS Nuclear Engineering Advanced Modeling and Simulation

PINSFV porous incompressible Navier-Stokes finite volume

RC Rhie Chow

TRISO Tri-structural Isotropic particle fuel

WCNSFV weakly compressible Navier-Stokes finite volume

ABSTRACT

The MOOSE framework is a library designed to make it straightforward for physicists and engineers to model partial differential equations using finite element and finite volume methods. Finite elements have been a part of the framework library since MOOSE's inception over a decade ago. Initial finite volume capability, however, was added only as recently as May of 2020. Since that time, significant work has been conducted to develop fluid modeling capability based on the finite volume method. Much of this work has taken place in MOOSE's `navier_stokes` module and in the NEAMS program's coarse mesh CFD code Pronghorn. We report here on development in MOOSE and Pronghorn of incompressible, porous incompressible, and weakly compressible finite volume simulation capabilities and their application to the modeling of MSR and FHR advanced reactor concepts.

ACKNOWLEDGEMENTS

Page intentionally left blank

CONTENTS

FIGURES

TABLES

Page intentionally left blank

1. Introduction

In this report we outline the recent developments in the Multiphysics Object Oriented Simulation Environment (MOOSE) and Pronghorn of incompressible and weakly compressible simulation capability and its application to modeling of advanced reactor concepts. Sections 2 to 4 outline incompressible Navier-Stokes finite volume (INSFV), PINSFV, and weakly compressible Navier-Stokes finite volume (WCNSFV) implementations respectively. Section 5 describes application of PINSFV to a fluoride salt-cooled high temperature reactor (FHR) concept. Finally, Section 6 describes application of both INSFV and WCNSFV implementations to modeling of a Molten Salt Fast Reactor (MSFR).

2. Finite Volume formulation of the incompressible Navier Stokes equations

2.1 Equations

The finite volume free-flow incompressible model (INSFV) implements conservation of mass and momentum and optionally energy and passive scalars. The conservation of mass or fluid continuity equation is given by

$$\nabla \cdot (\rho_f \mathbf{v}) = 0, \tag{1}$$

where ρ_f is the fluid density and $\vec{v}$ is the physical velocity, henceforward referred to in this section simply as the velocity. For strictly incompressible flow, ρ_f can be dropped from the equation. However, because the same mass advection kernel is reused for both incompressible and weakly compressible flows, we keep ρ_f in our implementation. The conservation of momentum equation for the fluid is

$$\rho_f \frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\rho_f \mathbf{v} \otimes \mathbf{v}) = \nabla \cdot (\mu \nabla \mathbf{v}) - \nabla p + \mathbf{F}_g, \tag{2}$$

where t is the time, μ is the fluid dynamic viscosity, p is the pressure, and $\mathbf{F}$ represents body forces such as gravitational force or buoyancy from the Boussinesq approximation. The fluid energy conservation equation is

$$\rho_f c_{pf} \frac{\partial (\epsilon T_f)}{\partial t} + \nabla \cdot (\rho_f c_{pf} \mathbf{v} T_f) = \nabla \cdot (k_f \nabla T_f) + \dot{q}_f, \tag{3}$$

where c_{pf}, T_f, and k_f represent the fluid specific heat, fluid temperature, and thermal conductivity respectively. $\dot{q}_f$ is the fluid extrinsic heat source. Passive scalars can be modeled using equations like

$$\rho_f \frac{\partial c}{\partial t} + \nabla \cdot (\rho_f \mathbf{v} c) = \nabla \cdot (D \nabla c) + \dot{S} \tag{4}$$

where c represents the scalar concentration, D represents the scalar diffusivity, and $\dot{S}$ represents the volumetric rate of production of c.

2.2 Implementation

2.2.1 Kernels (FVKernels)

The following kernels are implemented in INSFV. The kernels are developed with the MOOSE model of moving all terms to the left hand side. The divergence theorem is applied for advection and diffusion kernels to reduce the derivative order and intrinsically conserve quantities by balance of face fluxes. For the mass equation:

- `INSFVMassAdvection`: implements $\nabla \cdot (\rho_f \vec{v})$

For the momentum equation:

- `INSFVMomentumTimeDerivative`: implements $\rho_f \dfrac{\partial \mathbf{v}}{\partial t}$

- `INSFVMomentumAdvection`: implements $\nabla \cdot (\rho_f \mathbf{v} \otimes \mathbf{v})$

- `INSFVMomentumDiffusion`: implements $-\nabla \cdot (\mu \nabla \mathbf{v})$

- `INSFVMomentumPressure`: implements ∇p as a volumetric term

- `INSFVMomentumGravity`: implements a body force of the form $-\rho_f \mathbf{g}$ where $\mathbf{g}$ is the gravity vector

- `INSFVBodyForce`: implements a generic body force equal to $-\mathbf{f}$ where f is a user-provided function

- `INSFVMixingLengthReynoldsStress`: this kernel uses a mixing-length model to compute the Reynolds stress, $-\rho_f \overline{\vec{u}'\vec{u}'}$, which appears in Reynolds-averaged momentum equations. The velocity scaling is computed using Smagorinsky's formulation. This is further detailed in subsection 6.2.1.

- `INSFVMomentumBoussinesq`: this object adds a $\rho_f \alpha \mathbf{g}(T - T_{ref})$ term where α is the thermal expansion coefficient, T is the temperature, and T_{ref} is a reference temperature. The term above introduces the Boussinesq approximation into the incompressible Navier-Stokes (INS) equations, which allows for modeling natural convection.

For the energy equation:

- `INSFVEnergyTimeDerivative`: implements $\rho_f c_{pf} \dfrac{\partial(\epsilon T_f)}{\partial t}$

- `INSFVEnergyAdvection`: implements $\nabla \cdot (\rho_f c_{pf} \mathbf{v} T_f)$

- `FVDiffusion`: this is a generic kernel that be used to implement any conduction, diffusion, or even viscous operator although the latter is implemented with the `INSFVMomentumDiffusion` object introduced above for RC purposes. `FVDiffusion` implements a generic $-\nabla \cdot (D \nabla u)$ term where D is the diffusivity or thermal conductivity and u is the variable being diffused or conducted `FVBodyForce`: a generic kernel of form $-f$ where f is a user-provided function. This object can be used to model the $\dot{S}$ term in the energy equation

2.2.2 Boundary conditions (FVBoundaryConditions)

The following boundary conditions are available for the pressure or mass equation:

- `INSFVOutletPressure`: this object simply wraps `FVFunctionDirichletBC`, so a required parameter is `function` describing the pressure along an outlet boundary. The `variable` parameter should correspond to the pressure variable. `INSFVOutletPressureBC` also inherits from `INSFVFullyDevelopedBC` which allows for proper modification to the coefficients used to compute the RC interpolation in `INSFVMomentumAdvection`. When applying a `INSFVOutletPressureBC` for the pressure, no `FVBoundaryConditions` should be given for the velocity on the same `boundary`. In this way a zero viscous flux is implicitly applied for the velocity on the `boundary`.

- `INSFVSymmetryPressureBC`: though applied to the pressure, this object ensures that the velocity perpendicular to a symmetry boundary is zero by setting the mass flow rate across the symmetry boundary to zero. In addition to the `INSFVSymmetryPressureBC`, a `INSFVSymmetryVelocityBC` (described below) should be applied for every velocity component on a symmetry boundary. `INSFVMassAdvectionOutflowBC`: this object computes the residual and Jacobian contribution of the incompressible version of the mass continuity equation, e.g. $\nabla \cdot \mathbf{v} = 0$ along the domain boundary. This boundary condition should be used when it is desired that the pressure value be extrapolated at the boundary face. If the user wishes to have a certain value for pressure at the outflow boundary, then they should use `INSFVOutletPressureBC`.

The following boundary conditions are available for the momentum equations:

- `INSFVInletVelocityBC`: this object also simply wraps `FVFunctionDirichletBC`, so a required parameter is `function` describing the velocity along an inlet boundary. The `variable` parameter should correspond to a velocity component variables. If applying `INSFVInletVelocity` for any velocity component on a given `boundary`, then an `INSFVInletVelocity` should be specified for every velocity component on that `boundary`. A `FVBoundaryCondition` for pressure should not be applied on the same `boundary`. Instead a value for the pressure at the inlet will be extrapolated from the interior.

- `INSFVMomentumAdvectionOutflowBC`: this object implements the $\nabla \cdot \left(\rho_f \mathbf{v} \otimes \mathbf{v} \right)$ component term along a domain boundary. It imposes the condition that the normal gradient of each velocity component at the boundary is zero.

- `INSFVNaturalFreeSlipBC`: this object implements a free slip boundary condition. It should be applied to each velocity component. This BC operates very simply by setting the total momentum boundary flux, e.g. the sum of advective and viscous fluxes, to zero.

- `INSFVNoSlipWallBC`: as for other objects above, this object simply wraps `FVFunctionDirichletBC`. So the required parameter is `function` describing the boundary wall velocity for the velocity component specified with `variable`. If applying `INSFVNoSlipWallBC` for any velocity component on a given `boundary`, then an `INSFVNoSlipWallBC` should be specified for every velocity component on that `boundary`.

- `INSFVSymmetryVelocityBC`: this object implements a symmetry boundary condition for the velocity. It applies boundary forces such that the gradient of the velocity parallel to the boundary is zero in the boundary normal direction. A `INSFVSymmetryVelocityBC` should be applied for every velocity component on a symmetry boundary. Similarly an `INSFVSymmetryPressureBC` should be applied for the pressure on the symmetry boundary.

- `INSFVWallFunctionBC`: this object imposes a wall shear stress to the momentum equation according to the standard wall functions. Besides imposing the corresponding shear stress, it ensures zero advective flux at the wall (non-penetrability).
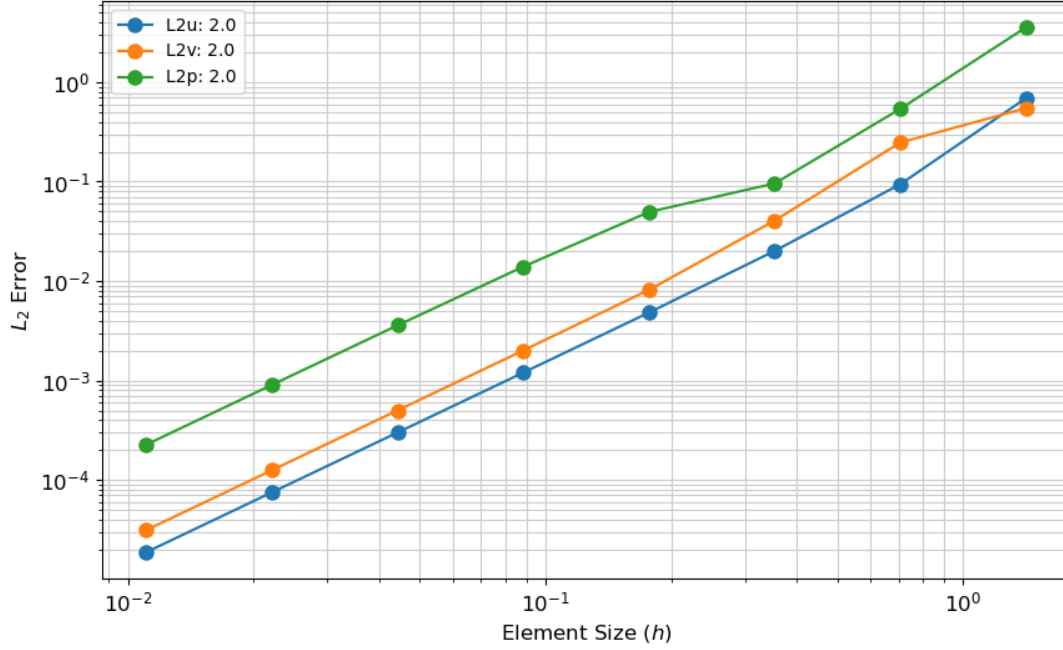
Figure 1: MMS plot for no-slip Cartesian channel flow configuration using RC interpolation for the velocity and an average interpolation for the advected quantities

2.3   Verification and Validation

2.3.1   Method of manufactured solution

The MMS is used to test the order of convergence of the numerical schemes implemented. We verify that a second order convergence rate is achieved for all nonlinear variables (velocity and pressure) using both average and RC interpolations for velocity and average interpolation for the advected quantity for a variety of problem configurations, including channel flow and cavity flow and in both Cartesian and axisymmetric coordinate systems. An example MMS verification plot is shown in Figure 1.

3. Finite volume implementation of the porous media incompressible Navier Stokes equations

The macroscale modeling approach is established to predict the mass, momentum, and heat transfer phenomena in porous media. The continuum concept is applied by taking an average of properties within the representative elementary volume [7, 8]. However, the model still requires knowledge of the closures for estimating the effective properties based on the pore-scale characteristics knowledge. Therefore, the Darcy-Forchheimer model is used with the Navier-Stokes equations at the continuum scale [9].

Progress on PINSFV was previously reported in [1], with a different format. More emphasis was placed on the format of the input file and the parameters to each kernel.

3.1 Equations

The finite volume implementation of PINSFV utilizes the equations accounting for the conservation of fluid mass, momentum, energy, and solid energy. First, the fluid continuity equation is given by

$$\nabla \cdot \vec{v}_D = 0, \tag{5}$$

where $\vec{v}_D$ is the superficial velocity or Darcy velocity. The superficial velocity is calculated by

$$\vec{v}_D = \epsilon \vec{V}, \tag{6}$$

where $\vec{V}$ is the interstitial/physical velocity, and ϵ is the porosity, which is the fluid volume divided by the total volume. The conservation of momentum equation for the fluid is

$$\rho_f \frac{\partial \mathbf{v}_D}{\partial t} + \rho_f \nabla \cdot \left(\frac{\mathbf{v}_D \otimes \mathbf{v}_D}{\epsilon} \right) = \nabla \cdot \left(\mu \nabla \frac{\mathbf{v}_D}{\epsilon} \right) - \epsilon \nabla p + \epsilon (\mathbf{F}_g + \mathbf{F}_f), \tag{7}$$

where ρ_f is the fluid density, t is the time, μ is the fluid dynamic viscosity, p is the pressure, $\mathbf{F}_g$ is the gravitational force, and $\mathbf{F}_f$ is the frictional force. The Boussinesq buoyancy is also added in the gravity term. The fluid energy conservation equation is

7

$$\rho_f c_{pf} \frac{\partial (\epsilon T_f)}{\partial t} + \rho_f c_{pf} \nabla \cdot (\mathbf{v}_D T_f) = \nabla \cdot (\kappa_f \nabla T_f) + \alpha (T_s - T_f) + \dot{q}_f, \tag{8}$$

where c_{pf}, T_f, T_s, κ_f, and α represent the fluid specific heat, fluid temperature, solid temperature, effective fluid thermal conductivity, and average convective heat transfer coefficient, respectively. $\dot{q}_f$ is the fluid extrinsic heat source. At last, the solid energy conservation equation is formulated as

$$\rho_s c_{ps} \frac{\partial (1 - \epsilon) T_s}{\partial t} = \nabla \cdot (\kappa_s \nabla T_s) + \alpha (T_f - T_s) + \dot{q}_s \tag{9}$$

where ρ_s is the solid density, c_{ps} is the solid specific heat, and $\dot{q}_s$ is the solid extrinsic heat source.

3.2 Implementation

PINSFV leverages the INSFV implementation as much as possible to maximize code reuse and reduce the maintenance burden. Numerous kernels only require small modifications from their non-porous flow equivalent, so class inheritance is leveraged as much as possible for that purpose.

3.2.1 Kernels (FVKernels)

The following kernels are implemented in PINSFV. For the mass equation:

- `PINSFVMassAdvection` which implements the mass equation $\nabla \cdot \mathbf{v}_D = 0$

For the momentum equation:

- `PINSFVMomentumTimeDerivative` for the time derivative in the momentum equation $\rho_f \frac{\partial \mathbf{v}_D}{\partial t}$. This kernel is essentially `INSFVMomentumTimeDerivative`, but accepts a different variable type, a `PINSFVSuperficialVelocityVariable`, which is only meant to be used for porous flow

- `PINSFVMomentumAdvection` for part of the advection term in the momentum equation $\rho_f (\frac{\nabla \mathbf{v}_D \cdot \mathbf{v}_D}{\epsilon})$

- `PINSFVMomentumAdvectionPorosityGradient` for the second part of the advection term in the momentum equation $-\rho_f (\frac{\nabla \epsilon \cdot \mathbf{v}_D \mathbf{v}_D}{\epsilon^2})$

- `PINSFVMomentumDiffusion` for the diffusion term in the momentum equation $\nabla \cdot (\mu \nabla \frac{\mathbf{v}_D}{\epsilon})$

- `PINSFVMomentumPressure` for the pressure term in the momentum equation $\epsilon \nabla p$

- `PINSFVMomentumPressureFlux` for a different formulation of the pressure term, using the divergence theorem (integration by parts) to make it a surface term rather than a volumetric term, equal to the sum over all element faces of $\epsilon_{face} p_{face}$

- `PNSFVMomentumPressureRZ` for an additional volumetric term in RZ geometry when integrating the pressure term by parts $-\dfrac{\epsilon p}{r}$ with r the radius

- `PINSFVMomentumPressurePorosityGradient` for an additional volumetric term $-p\nabla\epsilon$ when integrating the pressure term by parts

- `PINSFVMomentumBoussinesq` for the Boussinesq buoyancy term $\rho\alpha(T - T_0)$ with α an expansion coefficient and T_0 a reference temperature

- `PNSFVMomentumGravity` for the gravity term $\rho\vec{g}$

For the fluid phase energy equation:

- `PNSFVEnergyTimeDerivative` for the time derivative of the fluid energy equation (selected using a boolean) $\rho_f c_{pf} \dfrac{\partial(\epsilon T_f)}{\partial t}$

- `PINSFVEnergyAdvection` for the advection term in the fluid energy equation $\rho_f c_{pf} \nabla \cdot (\mathbf{v}_D T_f)$. The derivative of the the porosity term may be enabled using a boolean

- `PINSFVEnergyDiffusion` for the diffusion term in the fluid energy equation (selected using a boolean) $\nabla \cdot (\epsilon k_f \nabla T_f)$ using the fluid diffusivity for formulating this term. This is one potential formulation of this diffusion term

- `PINSFVEnergyEffectiveDiffusion` for the diffusion term in the fluid energy equation (selected using a boolean) $\nabla \cdot (\kappa_f \nabla T_f)$ using the fluid effective diffusivity for formulating this term.

- `PINSFVEnergyAmbientConvection` for the heat transfer between the fluid and solid phases $\alpha(T_s - T_f)$

For the solid phase energy equation:

- `PNSFVEnergyTimeDerivative` for the time derivative of the solid phase energy equation (selected using a boolean) $\rho_s c_{ps} \dfrac{\partial (1-\epsilon) T_s}{\partial t}$

- `PINSFVEnergyDiffusion` for the diffusion term in the solid phase energy equation (selected using a boolean) $\nabla \cdot (\kappa_s \nabla T_s)$

- `PINSFVEnergyAmbientConvection` for the heat transfer between the solid and fluid phases $\alpha (T_f - T_s)$

3.2.2 Boundary conditions (FVBoundaryConditions)

The following boundary conditions were implemented in PINSFV. The default boundary condition in finite volume is the zero flux boundary condition. This is equivalent to no penetration / no flow.

- `PINSFVMomentumAdvectionOutflowBC` for outflow boundary conditions where the pressure is specified. This boundary condition only computes an advection residual, and could be replaced by executing the advection kernel on the boundary.

- `PINSFVSymmetryVelocityBC` for symmetry boundary conditions (such as R=0 in RZ or symmetry planes in Cartesian coordinates). These simply divide the residual from the INSFV `SymmetryVelocityBC` by ϵ to adjust for the change in variables

The following boundary conditions from INSFV may be used in PINSFV with no particular change:

- `INSFVOutletPressureBC` for specifying a known pressure at the outlet of a system

- `INSFVSymmetryPressureBC` for symmetry boundary conditions in the mass equation

by substituting the velocity with the superficial velocity:

- `INSFVInletVelocityBC` for specifying an inlet velocity. This boundary condition should be used for each velocity component separately

- `INSFVNaturalFreeSlipBC`, for free-slip walls, which only indicates how to compute the Rhie Chow interpolation but does not contribute to the residual

- `INSFVNoSlipWallBC`, for no-slip walls, which only indicates how to compute the Rhie Chow interpolation but does not contribute to the residual

Other boundary conditions, such as specifying the velocity magnitude and direction over the outlet boundary, are currently not implemented.

3.2.3 Actions

These kernels and boundary conditions will be wrapped into actions to simplify and shorten the input file syntax in future work. Actions further remove implementation details from the eyes of users, which no longer have the detail of the implemented equations. However, they require extra attention as missing parameters to the action could remove entire terms from the equation.

3.3 Testing

The PINSFV module is extensively tested using the MOOSE continuous integration tool CIVET [10].

3.3.1 Regression testing

Each new contribution is automatically checked and regressions are prevented using regression testing. The following configurations are tested:

- 2D square channel with free-slip boundary conditions

- 2D RZ cylindrical channel, free-slip

- 2D channel with no slip boundary conditions

- 2D channel, no slip, with pressure driven flow

- 2D channel, no slip, with an average pressure imposed rather than pressure outlets

- half of a square 2D channel, using symmetry boundary conditions

- half of a cylindrical channel, using symmetry boundary conditions in RZ geometry

11

- 2D channel with uniform friction

For a heated channel, which tests the implementation of the energy equation:

- a heated 2D channel, modeling only the fluid phase

- a heated 2D RZ cylindrical channel

- a heated 2D channel with the effective diffusivity formulation

- a 2D heated channel, modeling the heating of both the fluid and solid phases

- a 2D heated channel, with both phases, modeling buoyancy using the Boussinesq approximation

The porosity jump is a major problem of interest and is subject to dedicated testing with a 1D and 2D channel test. The testing currently highlights the oscillations. Both a smooth and discontinuous porosity gradient are considered for the test. Removing these oscillations will be a subject of future work.

Finally, conservation tests were created for a straight 2D channel and an expanding 2D channel. These tests showed that PINSFV conserves mass, momentum and energy, though highlighted difficulties in post-processing mass flow rates on faces. The Rhie Chow interpolation is currently performed by the advection kernels to obtain the face velocity. The postprocessors cannot reproduce this calculation, and have to resort to average interpolation. This limits their usefulness in checking for the conservation of energy and momentum.

3.3.2 Method of manufactured solution

In addition to regression testing, MMS is used to test the order of convergence of the numerical schemes implemented. As for Section 2.3.1 we verify that a second order convergence rate is achieved for all variables using Rhie Chow interpolation for velocity and average interpolation for the advected quantity. This is checked for both a 1D and a 2D channel.

This test is also run with a continuously varying porosity profile, for both a 1D and 2D channel, and second order convergence is also achieved for those cases.

(a) average velocity interpolation   (b) RC velocity interpolation
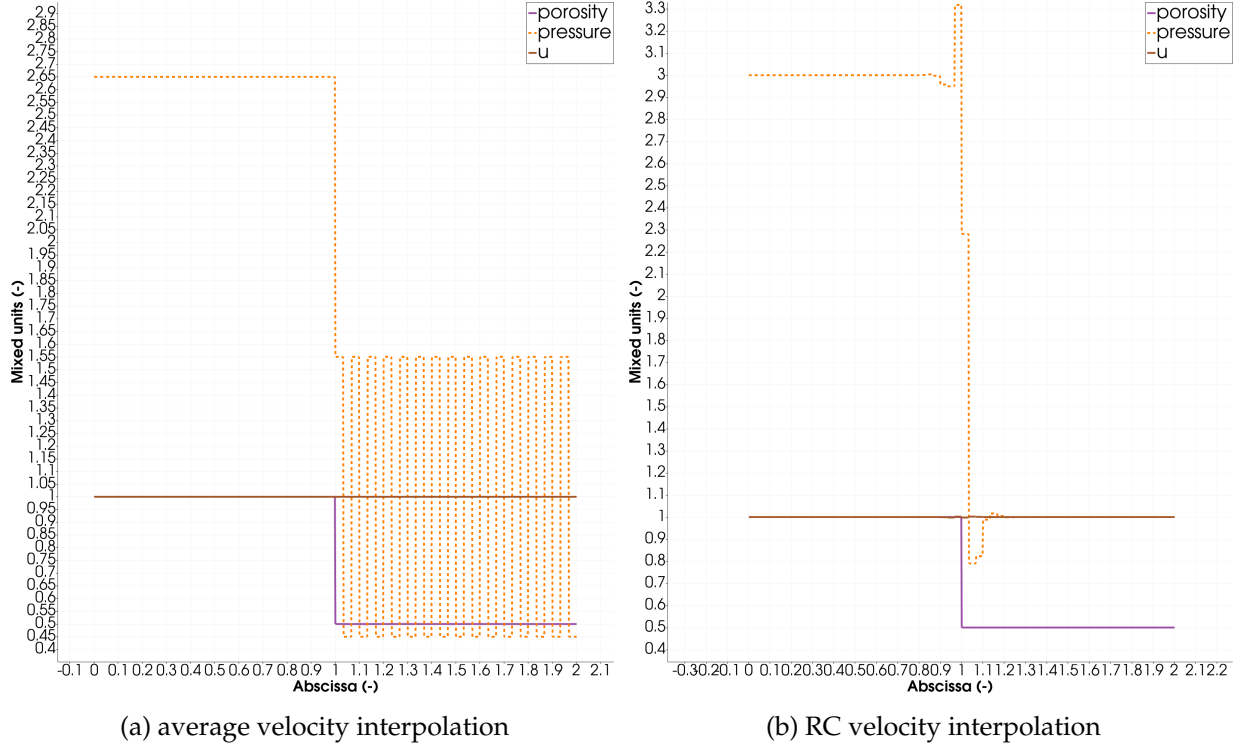
Figure 2: Pressure, superficial velocity, and porosity profiles for simple 1D channel simulations with a porosity jump.

3.3.3 Coverage testing

Code coverage testing enables developers to make sure that all the produced code is covered by unit and regression testing. Any modification to the code that happens in an uncovered region is guaranteed not to break any test, even if incorrect. So the smaller the uncovered region, the more likely an erroneous modification of the code will be caught by the test suite. The current coverage of PINSFV is above 80%, in line with the coverage in MOOSE and in the Navier Stokes module. The kernels that are currently not covered will be covered in a corrective action following this report. A screenshot of the coverage report is shown in Figure 5

3.4 Verification & Validation

The verification and validation of Pronghorn is imperative to assert its accuracy when predicting the fluid velocity, temperature, and pressure in high-temperature gas-cooled reactors. Jieun Lee et al. validated the finite volume porous media model implementation in Pronghorn with pressure drop measurements performed at Texas A&M University [3], and it was also verified by

13

Figure 3: Continuously varying porosity profile, forcing term and solver output for a fine mesh case used for a MMS study, extracted from [1]

(a) average velocity interpolation

(b) RC velocity interpolation

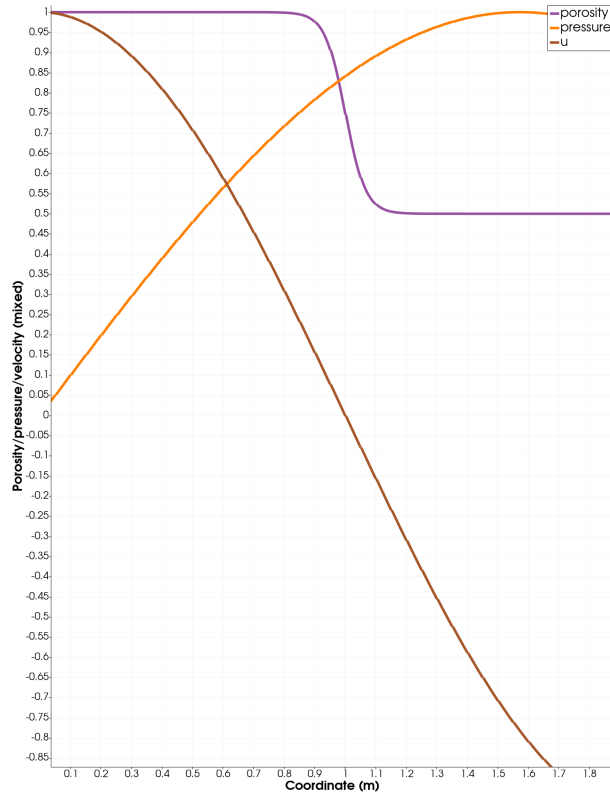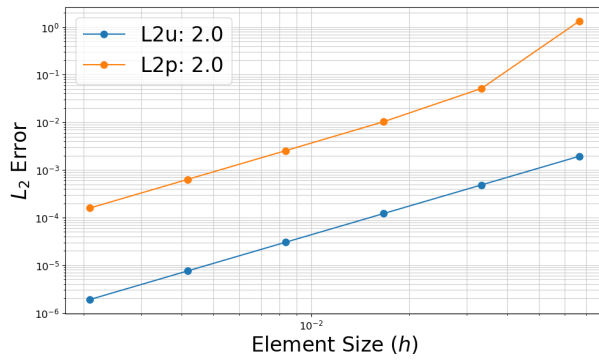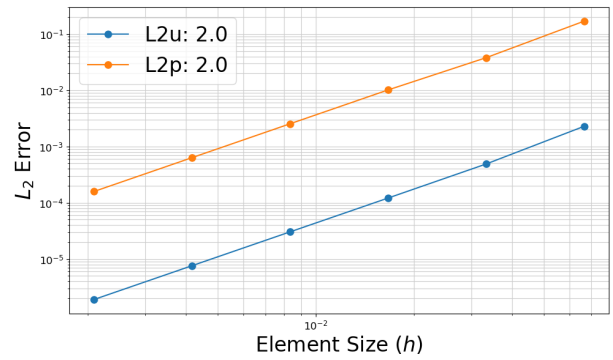Figure 4: Error convergence plot verifying expected second order convergence rate with average advected quantity interpolation in PINSFV, extracted from [1].
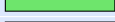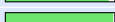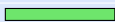
14
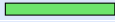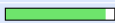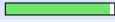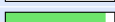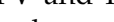
PCNSFVDensityTimeDerivative.C		100.0 %	13 / 13	100.0 %	3 / 3
PCNSFVFluidEnergyHLLC.C		66.7 %	16 / 24	62.5 %	5 / 8
PCNSFVHLLC.C		94.7 %	72 / 76	100.0 %	4 / 4
PCNSFVKT.C		98.7 %	151 / 153	100.0 %	5 / 5
PCNSFVKTDC.C		100.0 %	39 / 39	100.0 %	7 / 7
PCNSFVMassHLLC.C		64.7 %	11 / 17	62.5 %	5 / 8
PCNSFVMomentumHLLC.C		65.4 %	17 / 26	62.5 %	5 / 8
PINSFVEnergyAdvection.C		89.5 %	17 / 19	100.0 %	3 / 3
PINSFVEnergyAmbientConvection.C		100.0 %	18 / 18	100.0 %	3 / 3
PINSFVEnergyDiffusion.C		95.5 %	21 / 22	100.0 %	3 / 3
PINSFVEnergyEffectiveDiffusion.C		95.2 %	20 / 21	100.0 %	3 / 3
PINSFVMassAdvection.C		100.0 %	10 / 10	100.0 %	3 / 3
PINSFVMomentumAdvection.C		93.4 %	113 / 121	100.0 %	6 / 6
PINSFVMomentumAdvectionPorosityGradient.C		93.9 %	31 / 33	100.0 %	3 / 3
PINSFVMomentumBoussinesq.C		91.7 %	11 / 12	100.0 %	3 / 3
PINSFVMomentumDiffusion.C		95.7 %	44 / 46	100.0 %	3 / 3
PINSFVMomentumPressure.C		91.7 %	11 / 12	100.0 %	3 / 3
PINSFVMomentumPressureFlux.C		96.0 %	24 / 25	100.0 %	3 / 3
PINSFVMomentumPressurePorosityGradient.C		0.0 %	0 / 21	0.0 %	0 / 3
PINSFVMomentumTimeDerivative.C		0.0 %	0 / 9	0.0 %	0 / 2
PNSFVEnergyTimeDerivative.C		0.0 %	0 / 23	0.0 %	0 / 3
PNSFVMomentumFriction.C		92.3 %	36 / 39	100.0 %	3 / 3
PNSFVMomentumGravity.C		100.0 %	11 / 11	100.0 %	3 / 3
PNSFVMomentumPressureRZ.C		0.0 %	0 / 11	0.0 %	0 / 3
PNSFVPGradEpsilon.C		100.0 %	14 / 14	100.0 %	3 / 3

Figure 5: Coverage report for kernels in PINSFV and PCNSFV. The two columns next to the progress bar are for line coverage, while the two columns to the right are for function coverage

STAR-CCM+ simulation results.

The TAMU pebble bed test facility design is shown in Figure 6. The behavior of the gas flow in the pebble bed is estimated by experiments assuming isothermal conditions. The experiments cover the flow range from laminar to turbulent regimes for validating correlations. The aspect ratios are relatively small to observe the near-wall effects of pebble bed reactors.

In Figure 7, for the axisymmetric model, the uniform mesh with 1,024 cells is used for the free flow region where only the pure fluid is provided, and 14,080 elements are used in the porous region. For both the Pronghorn and STAR-CCM+ finite volume cases, the boundary conditions are the inlet velocity, outlet pressure, and free-slip conditions on the walls. The initial conditions are the ambient temperature and pressure. STAR-CCM+ v16.02.009 generates the reference solutions for quantities, such as the pressure drop and velocity profile, and the steady state solutions under laminar/turbulent flow conditions are obtained with the Darcy-Forchheimer model. The second-order upwind scheme is used for convection, and the segregated SIMPLE solver is used.

The average porosity of a randomly packed bed was estimated by correlations. We investigated twelve different average porosity correlations from [11]. Consequently, Mueller, Sato, and Zou and Yu correlations were selected by satisfying their validity conditions and considering the experimental setups comparable to the TAMU isothermal experiments. These correlations were
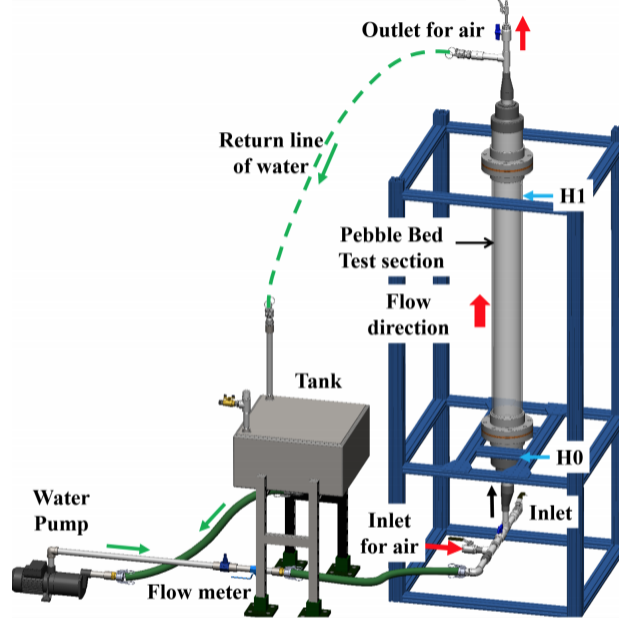
Figure 6: Schematic of the pebble bed test facility from [2].
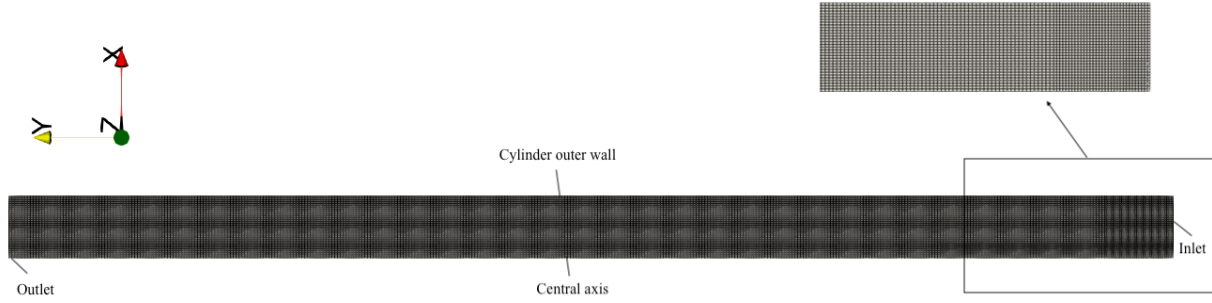
Figure 7: Geometry and fine mesh of the Pronghorn and STAR-CCM+ models (axisymmetric modeling) [3].

developed by the researchers with pebble and bed diameters similar to the current experiments, including the random packing method and the pebble material, such as acrylic glass. In Figure 8, the red line represents the aspect ratio of 7.33 used for the water cases. The average porosities calculated by the above correlations vary within the porosity range by $\sim$0.05, which is relatively large compared to the values of the averages. This would likely provide the differences in pressure drop when the empirical pressure drop correlations are applied in the Darcy-Forchheimer model (see Table 1).

The optimal porosity function was determined by examining the porosity functions in Table 1;

16
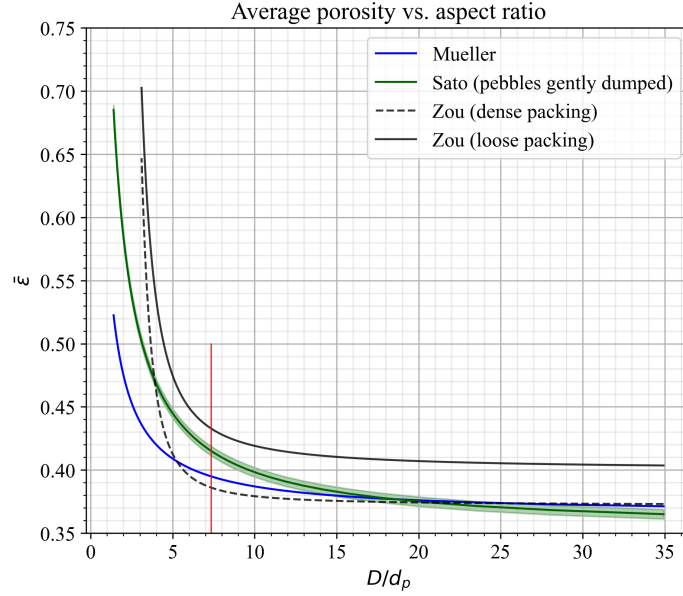
Figure 8: Comparison of average porosity correlations.

Table 1: Average porosities estimated by different correlations.

Model/ average porosity	$\bar{\varepsilon}$
Mueller	0.395
Sato (pebbles gently dumped)	0.4152
Zou (dense packing)	0.3861
Zou (loose packing)	**0.4329**

these functions were used with the Eisfeld and Schnitzlein and KTA pressure drop correlations. As a result, the Zou loose packing model approximates the pressure drop better than other models with regard to the validation metrics in Table 2. The equations for the validation metrics are described in detail in [12].

Table 2: Validation metrics for the Zou loose packing model [3].

Validation metrics/ model (1D calculation)	Eisfeld and Schnitzlein	KTA
Average relative difference (%)	37.61	39.82
RMSE (Pa)	52.04	51.17
RRMSE (%)	18.89	18.58
L^2 norm (Pa)	194.7	191.5
Average of the total normalized Euclidean distance	0.7963	0.7833

After determining the average porosity, numerous pressure drop correlations valid in the specific Reynolds number and aspect ratio ranges were implemented in both Pronghorn and STAR-CCM+ to estimate the pressure drop in the pebble bed. First, the GCIs for fine meshes are less than 1% with the lowest and highest particle Reynolds numbers; therefore, the uncertainty error bars, due to the numerical uncertainty, are estimated as a conservative error of 1% of the predicted pressure drop. This shows that the fine meshes, in general, provide numerical results close to the asymptotic solutions with the uncertainty errors less 1% and ensure a higher than 99% confidence level for the computed values.

For the experiments with the nearly incompressible fluid, water, the Montillet dense packing model results in the lowest RMSEs, RRMSEs, L^2 norms, and averages of the total noramlized Eulidean distance with Pronghorn and STAR-CCM+. The Montillet function was developed based on similar pebble and bed sizes, including the random packing of glass particles, and it considers the near-wall effect to improve the pressure drop prediction. Here, the aspect ratio of 7.33 is small such that the near-wall effect is probably influential. Although the average relative differences in Table 3 are beyond the maximum relative difference of 10.4% in [13], in Figure 9, overall, the overlap of the 95% confidence interval bars of the experiments and simulation results is more than the half of a single arm. Therefore, the two-tailed p-value is ≤ 0.05, and it provides strong evidence for the null hypothesis of the zero difference in means [14–16]. In other words, the statistical significant effect is most likely small. Figure 10 describes the gauge pressure and velocity magnitude profiles of the $Re_p = 676$ case given the Montillet dense packing function. Not only do the simulation results from Pronghorn and STARCCM+ agree well, but they comparably estimate the experimental pressure drop.

In conclusion, the finite volume Pronghorn porous media model predicts the pressure drop well relative to the STAR-CCM+ simulation results and 1D correlations, and it is also validated with the experimental measurements given different Reynolds number ranges and the specific aspect ratio. There exists strong evidence that there is no statistically significant difference in means, such as the pressure drop measurements, specific correlations, or simulations, provided that the overlap of their confidence intervals is more than the half of a single arm. Several validation metrics show similar levels of agreement to other studies. The precise average pebble bed poros-
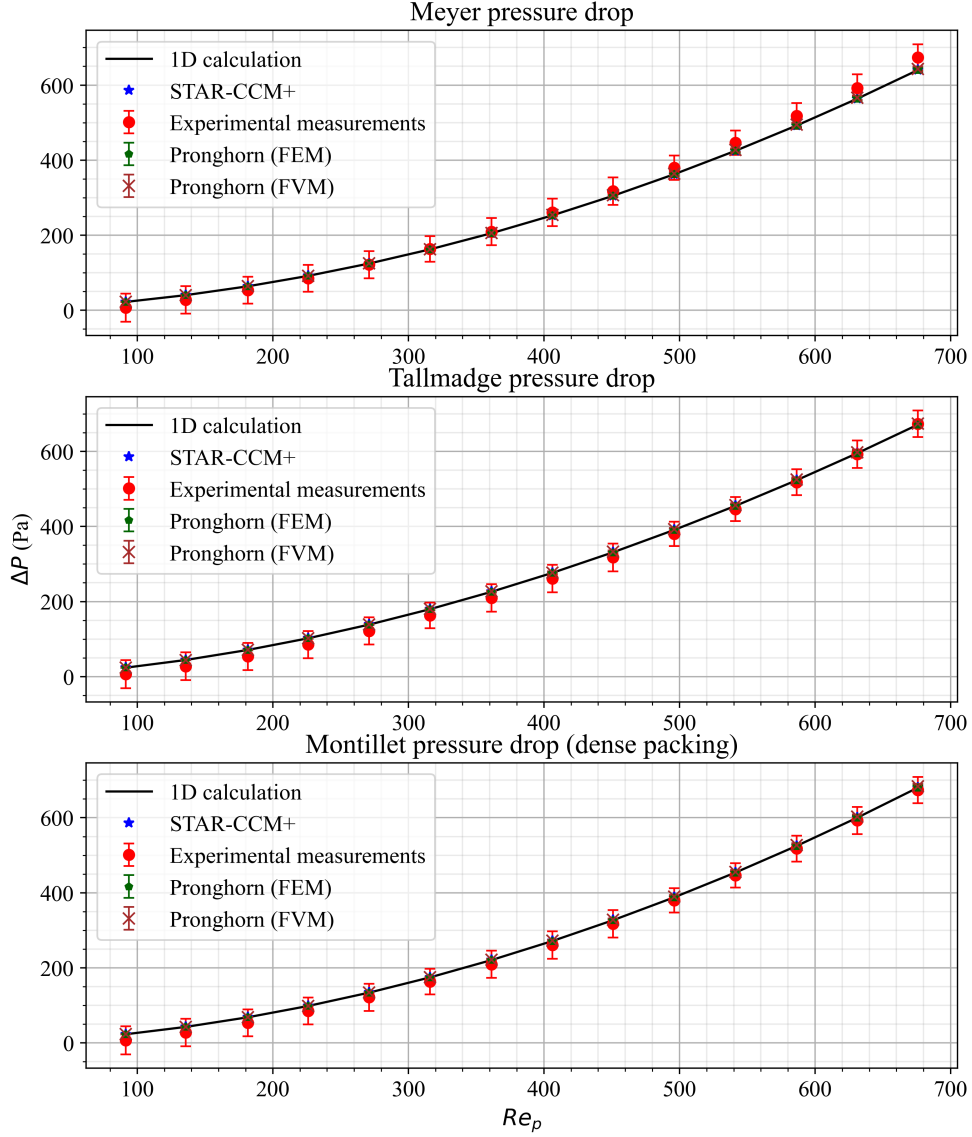
Figure 9: Pressure drop correlations for water [3].

ity estimation has a large impact on the pressure drop, and the Montillet dense packing model provides an accurate pressure drop prediction while also considering the near-wall effect.

3.5 Ongoing and future efforts

The development of Pronghorn and its transition to a finite volume formulation continues in FY22. In this section we detail ongoing efforts which will be further reported next year.

Table 3: Validation metrics for the Montillet function.

Validation metrics/ model	Montillet (dense packing)	
	Pronghorn (FV)	STAR-CCM+
Average relative difference (%)	27.37[§]	26.92[§]
RMSE (Pa)	12.19	11.23
RMSRE	0.6742[*]	0.6715[*]
RRMSE (%)	4.424	4.075
L^2 norm (Pa)	45.59	42
Average of the total normalized Euclidean distance	0.1821	0.1671

[§]The Meyer function brings similar average relative differences of 24.16 and 24.26 for Pronghorn (FV) and STAR-CCM+, respectively.
[*]The Meyer function shows similar RMSREs of 0.6347 and 0.6324 for Pronghorn (FV) and STAR-CCM+, respectively.
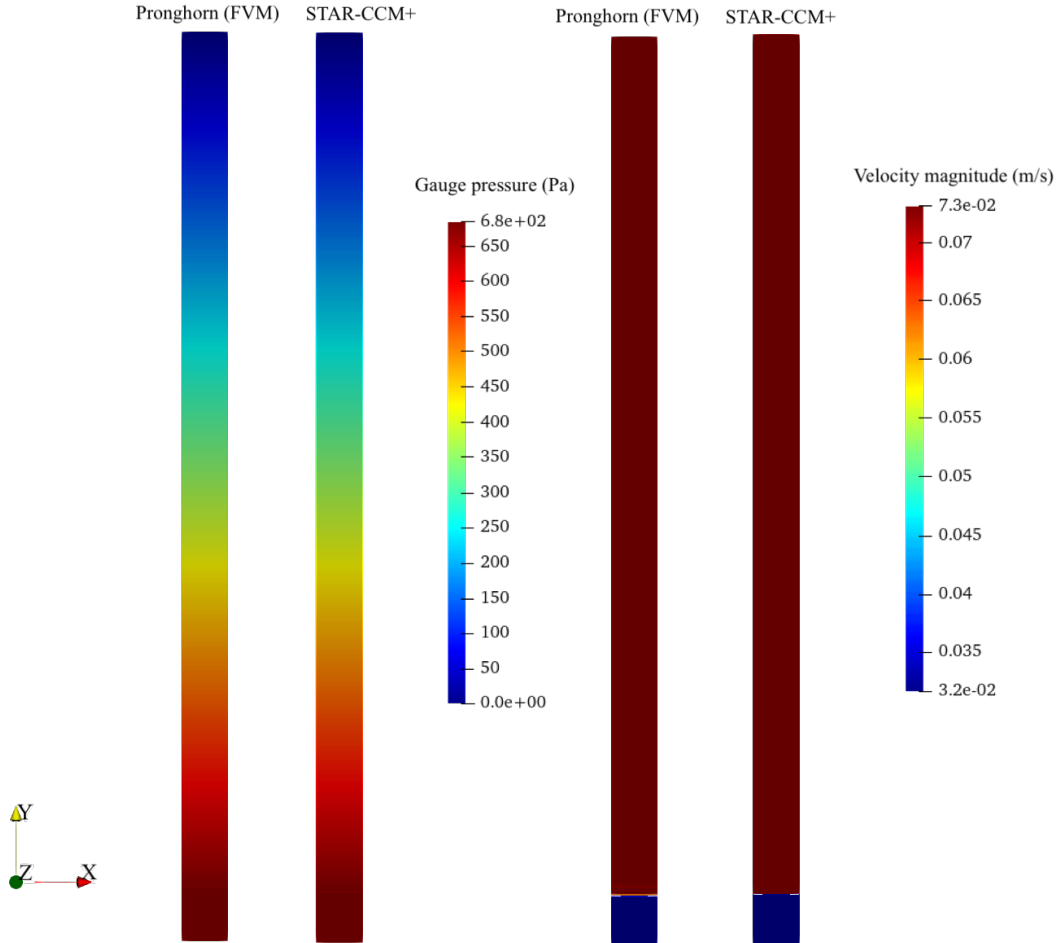
Figure 10: Gauge pressure and velocity magnitude profiles estimated by the Montillet dense packing function with Pronghorn and STAR-CCM+ for $Re_p = 676$.

3.5.1 Treatment of porosity jumps: face averaging of body forces in the Rhie Chow interpolation

Porosity jumps occur at interfaces between different homogenized regions, such as the core of a fluoride high temperature reactor and its reflector, to model bypass flow. This discontinuity in porosity is problematic for the solver as the gradient of the porosity is present several times in the equations and becomes ill-defined. This introduces oscillations.

Another challenge is that the friction volumetric terms are different on both sides of the interface. The Rhie Chow interpolation, which is in effect a momentum equation on the face, has to account for this. In its current form in PINSFV it does not. This leads to larger oscillations than the first issue. The necessary treatment of body forces in Rhie Chow interpolation is outlined in [17] and is reproduced below:

$$v_f = \bar{v}_f - \bar{D}^v_f(\nabla p_f - \bar{\nabla} p_f) + \bar{D}^v_f(B^v_f - \bar{\bar{B}}^v_f)$$ (10)

where $\bar{\bar{B}}^v_f)$ is the geometric surface average of the volumetric average of the surface average of the volumetric terms.

$$\bar{\bar{B}}^v_f) = g_C \bar{\bar{B}}^v_C + (1 - g_C)\bar{\bar{B}}^v_F$$ (11)

where F,C are neighbor and element centroids.

In a fully coupled solver, this interpolation is problematic, as we would rather not use previous estimate of the pressure and the velocity to compute the terms required in the interpolation. The kernel system does not offer sufficient flexibility for this purpose, and we will be implementing a `UserObject` that will store volumetric residual contributions for use within the same iteration for computing Rhie Chow velocities.

Other modifications to the Rhie Chow interpolation, proposed by Zhang [18] and Nordlund [19], may also be considered. One proposed modification is to use a baffle/interface approach to connect two subdomain solutions with a simple equation for a set pressure drop, as laid out by [20]. Finite volume interface kernels were developed to explore this second approach, but the baffle implementation has not been finalized or tested.

3.5.2 SANA benchmarks validation

The SANA experiments are porous media natural convection experiments and were used to validate the finite element version of Pronghorn [5, 21]. As part of this earlier validation effort, both a finite element solution and a reference solution using StarCCM+©were generated. These will be leveraged to support the verification of finite volume implementations in Pronghorn.

4. Finite Volume formulation of the Weakly Compressible Navier Stokes equations

4.1   Equations

The equations describing weakly compressible flow are very similar to those outlined in Section 2.1 but with the allowance for density variations in time and space. Most critically the density is brought into the time derivative term such that the mass, momentum, and energy time derivative terms become $\frac{\partial \rho_f}{\partial t}$, $\frac{\partial (\rho_f \mathbf{v})}{\partial t}$, and $\frac{\partial (\rho_f c_{pf} T_f)}{\partial t}$ respectively.

4.2   Implementation

The weakly compressible implementation of the Navier-Stokes equations reuses many of the objects from Section 2.2. What determines incompressible vs. weakly compressible simulation behavior is how the density is defined. We recently implemented a new capability in MOOSE for evaluation at arbitrary points in time and space of various MOOSE objects, including `FunctorMaterialProperties` `MooseVariables`, and `Functions`. Moreover, any of these functor objects, when retrieved through the `FunctorInterface`, can be interchanged, e.g. if a kernel has requested a functor of name `"foo"`, then it makes no difference to the kernel whether `"foo"` is actually a material property, variable, or function. The ability to arbitrarily evaluate at any point in time and space was critical to implementation of WCNSFV because we must be able to evaluate the density on arbitrary elements/cells when computing RC coefficients. This capability is also important when the viscosity is spatially or temporally varying.

Incompressible behavior is obtained when the density is defined as a constant functor. Weakly compressible behavior is obtained when the density is a functor dependent on temperature and pressure, e.g. when computed in the object `RhoFromPTFunctorMaterial`.

4.2.1   Kernels

WCNSFV leverages the same advection, diffusion, and body force kernels as INSFV (see Section 2.2.1 for those objects). The treatment of non-constant density in the transient terms is handled in a few new objects however:

23

- `WCNSFVMassTimeDerivative`: implements $\frac{\partial \rho_f}{\partial t}$

- `WCNSFVMomentumTimeDerivative`: implements $\frac{\partial (\rho_f \mathbf{v})}{\partial t}$

- `WCNSFVEnergyTimeDerivative`: implements $\frac{\partial (\rho_f c_{pf} T_f)}{\partial t}$

WCNSFV boundary conditions reuse the same objects outlined in Section 2.2.2.

4.3   Verification and Validation

4.3.1   Natural convection benchmarks

The WCNSFV implementation has been tested against some traditional square domain natural convection benchmarks. The results for Rayleigh numbers of 1,000 and 1,000,000 are shown in Figure 11 and Figure 12 respectively. A key difference between INSFV and WCNSFV modeling of natural convection is the the former requires the Boussinesq approximation kernel (`INSFVMomentumBoussinesq`) whereas the latter produces the correct natural convection physics without the additional kernel.
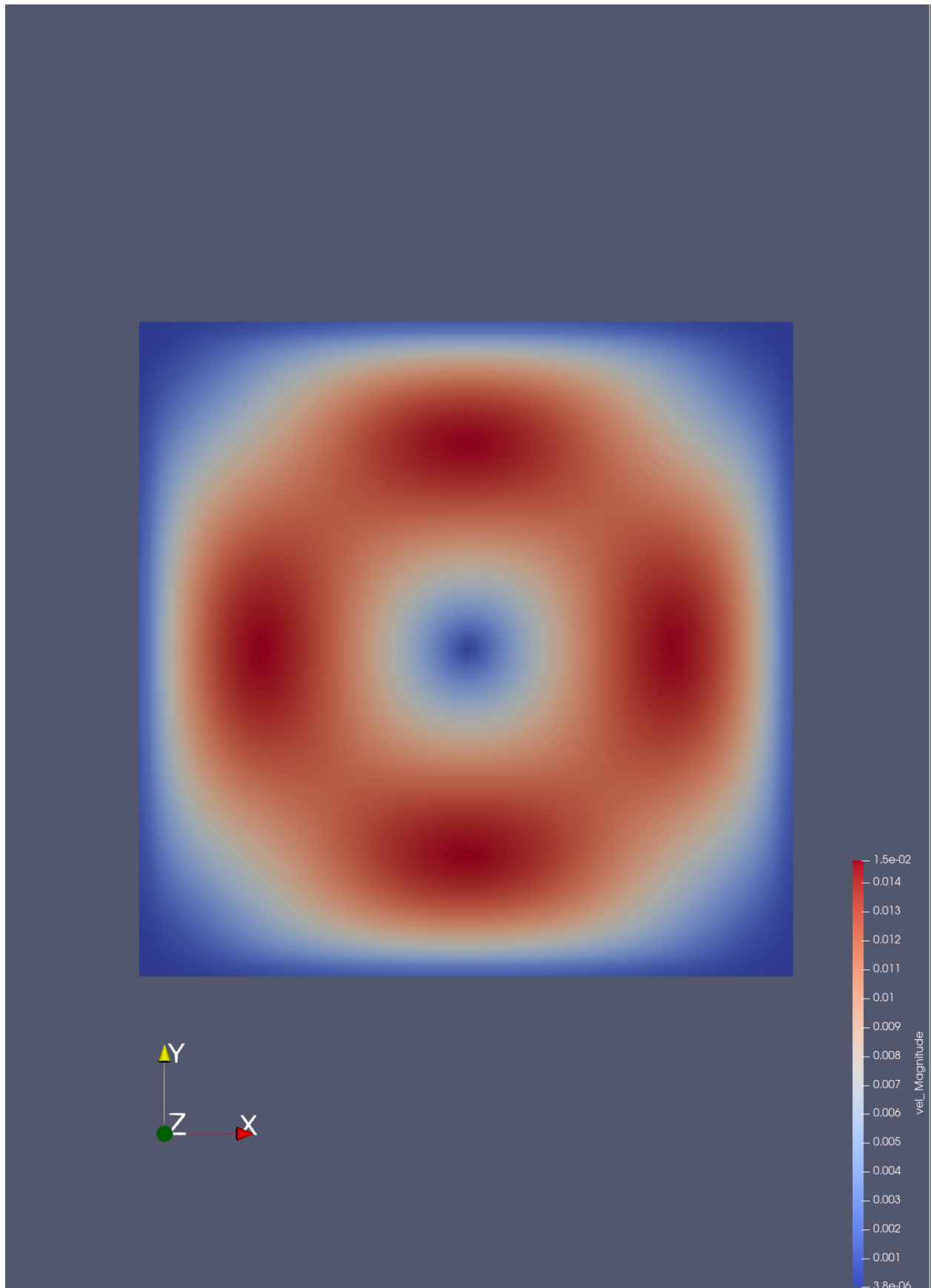
Figure 11: Velocity magnitude for a Raleigh number of 1,000
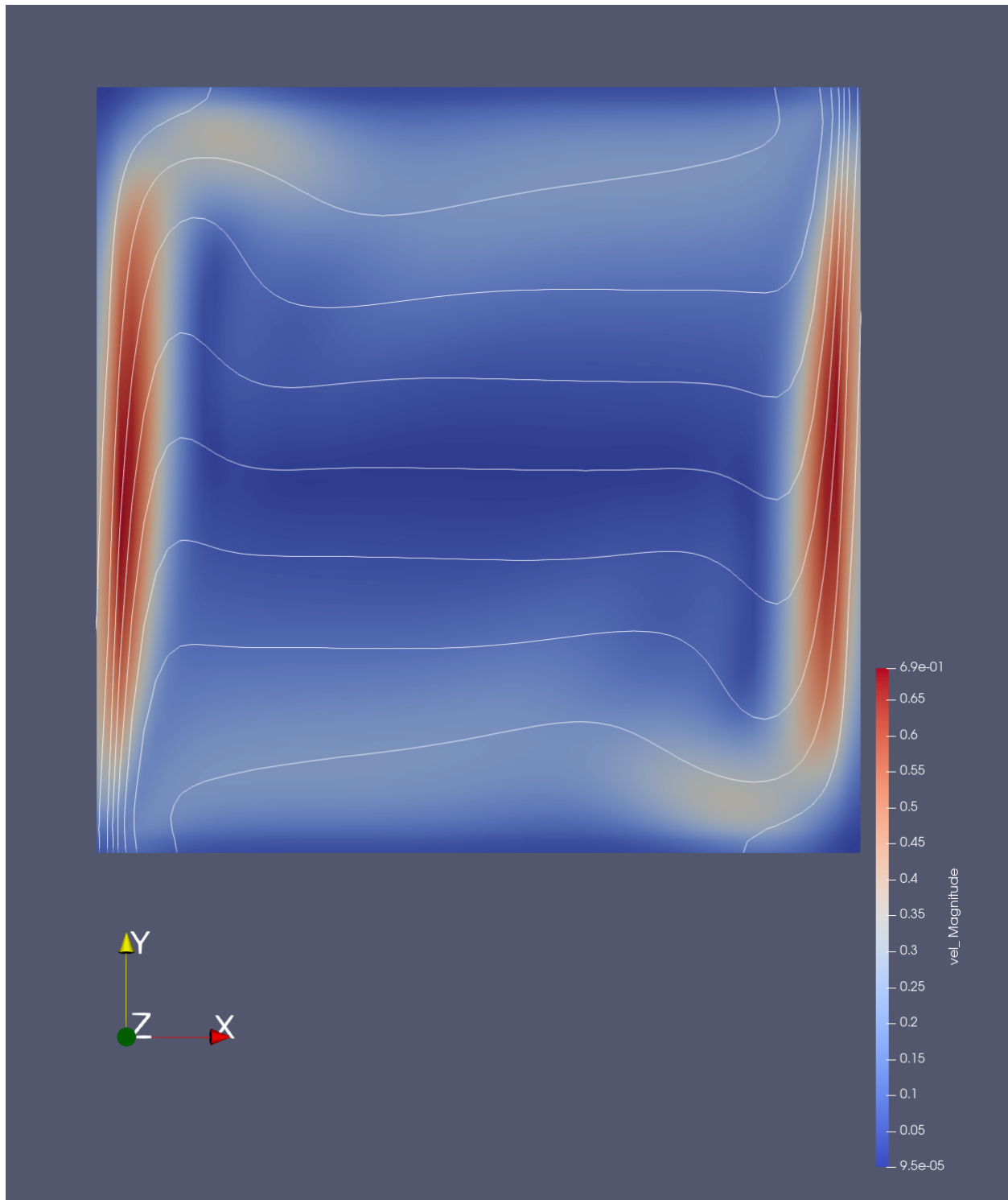
Figure 12: Coloring according to velocity magnitude for a Raleigh number of 1,000,000. Contour lines represent the temperature field

5. Fluoride High Temperature Reactor reference plant

The Fluoride salt cooled High temperature Reactor (FHR) is an advanced reactor technology actively pursued by U.S. nuclear companies. The reactor is composed of hundreds of thousands of fuel pebbles cooled by a Beryllium Lithium Fluoride salt (FLiBe).

The coupled multiphysics FHR core model drove the development of PINSFV for the NEAMS program, as an example of a realistic application of PINSFV to reactor modeling. The core simulation is hosted on the Virtual Test Bed of the National Reactor Innovation Center [6, 22] to advertise the capabilities of NEAMS tools. The reader may consult this report [1] for a similar model in three dimensions.

To demonstrate the new finite volume modeling capabilities for multiphysics reactor simulations, a model of the Mark-1 Pebble-Bed Fluoride-Salt-Cooled Reactor (Mk1-FHR) was built in MOOSE. The model uses SAM, Griffin and Pronghorn [21]—the 1D thermal hydraulics, neutronics and intermediate fidelity flow MOOSE applications, respectively—as well as the MOOSE heat conduction module. This suite of tools is aimed at providing medium-fidelity modeling of the plant—accurate enough so complex behaviors such as accident response or fluid-structure interactions may be studied, but simple enough that simulations can be executed on a desktop computer within a reasonable runtime, usually on the order of minutes to a few hours.

The descriptions of each part of the model, save for the integrated plant analysis using SAM and the general coupling scheme, are adapted from [1]. Results are updated with the current version of the coarse mesh CFD model.

5.1 The Mk1-FHR Plant

We modeled the Mk1 PB-FHR reactor, an open-source design, facilitating collaboration and comparisons with other open literature. The Mk1 PB-FHR is a pre-conceptual design of a fluoride salt-cooled high-temperature reactor. Its objective is to study numerous potential benefits of the FHR technology with regards to efficiency, safety, and construction.

The Mk1 PB-FHR uses an air Brayton cycle to produce 100MWe from 236MWth with the possibility of natural gas co-firing for additional power output. It uses a direct reactor auxiliary cooling system, or DRACS, to passively cool the core in the case of a loss of active cooling. The particulate

fuel is also resistant to higher temperatures than traditional fuel, further improving the performance in loss of cooling accident scenarios.

The Mk1 PB-FHR has an annular cylindrical core. The salt flows upwards and pebbles are buoyant and float at the top of the core. Pebbles enter through a fueling chute, while the fluoride salt coolant enters from both the fueling chute and the inner reflector. Pebbles leave the core through a defueling chute at the top of the core. Their burnup is examined before they are re-inserted or withdrawn from the core. This allows for online refueling, limiting the need for outages and improving the economic performance of the reactor. There is significant coolant bypass flow in the outer reflector

5.2   Model Description

We perform a coupled multiphysics simulation of the Mk1-FHR using coupled applications. Each physics is solved for by the relevant MOOSE application: the neutronics by Griffin, the thermal hydraulics by Pronghorn, the integrated plant analysis with SAM and the fuel performance by the combined use of Pronghorn and the MOOSE heat conduction module. A combined application, such as BlueCRAB, is currently necessary to run the multiphysics coupled problem.

The physics coupling is performed using the MultiApp system. In MOOSE jargon, the neutronics application is the main application; the thermal hydraulics and heat conduction are sub-apps. The coupling scheme is shown in Figure 13. Applications are run successively with Picard fixed-point iterations to converge the multiphysics problem. More advanced fixed point algorithms have been developed previously for the thermal coupling [4], and they may be used in future works. They are said to be tightly coupled, as opposed to loose coupling if the scheme was not iterated, and fully-coupled if a single matrix was used to solve the multiphysics problem.

5.2.1   Neutronics model in Griffin

The neutronics calculation computes the power distribution (i.e., the heat source term for the thermal hydraulics simulation). We will briefly summarize the neutronics model and will refer the reader to other reports and publications [23, 24] for additional details on the Mk1-FHR neutronics model.
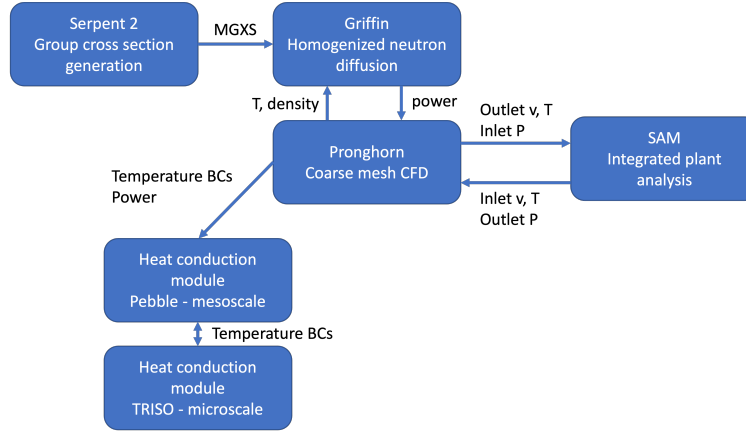
Figure 13: Coupling scheme for the reference plant multiphysics model

Generation of group cross sections with Serpent 2   We generated group cross sections for Griffin using Serpent 2 [25]. Using Monte Carlo methods for group cross section generation allows for accurate representation of complex resonance self-shielding in highly heterogeneous geometries. The use of delta tracking in Serpent 2 limits the computational cost of the simulation in this complex geometry. We based the Serpent 2 model off the model in [26] and improved it to include random pebble packing in the middle of the core. Figure 14 shows an axial cut of the 3D model used. Cross sections are generated for a large range of uniform fuel temperature, coolant temperature, and control rod insertion.

Griffin model   The neutronics model consists of an eight group homogenized diffusion problem without equivalence. The cross section regions each cover a major component of the core: active pebble bed, reflector pebble bed, inner reflector, control rod channel, and outer reflector. The baffle is excluded from the neutronics model as its influence is negligible.

Figures 15 and 16 show the thermal and fast group fluxes and power distribution computed by Griffin in 2D without thermal-hydraulics coupling. The thermal flux peaks are located near both the inner and outer reflector, which thermalize neutrons while absorbing very few of them. The higher energy neutrons making up the fast flux have a longer mean free path, and their distribution is driven by neutron streaming/leakage. The power distribution is mostly driven by the thermal flux as the fuel fission cross sections are higher in the thermal range.
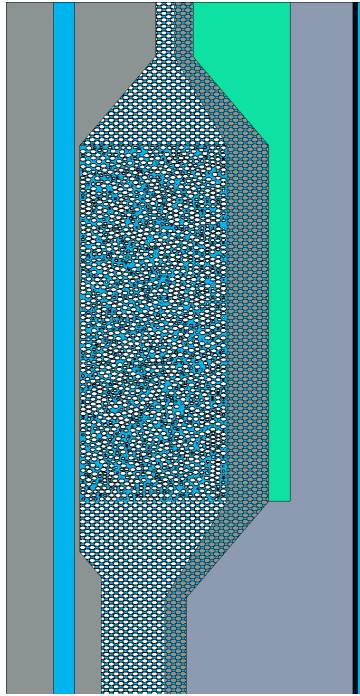
Figure 14: Axial cut of the FHR core in Serpent., from [1]
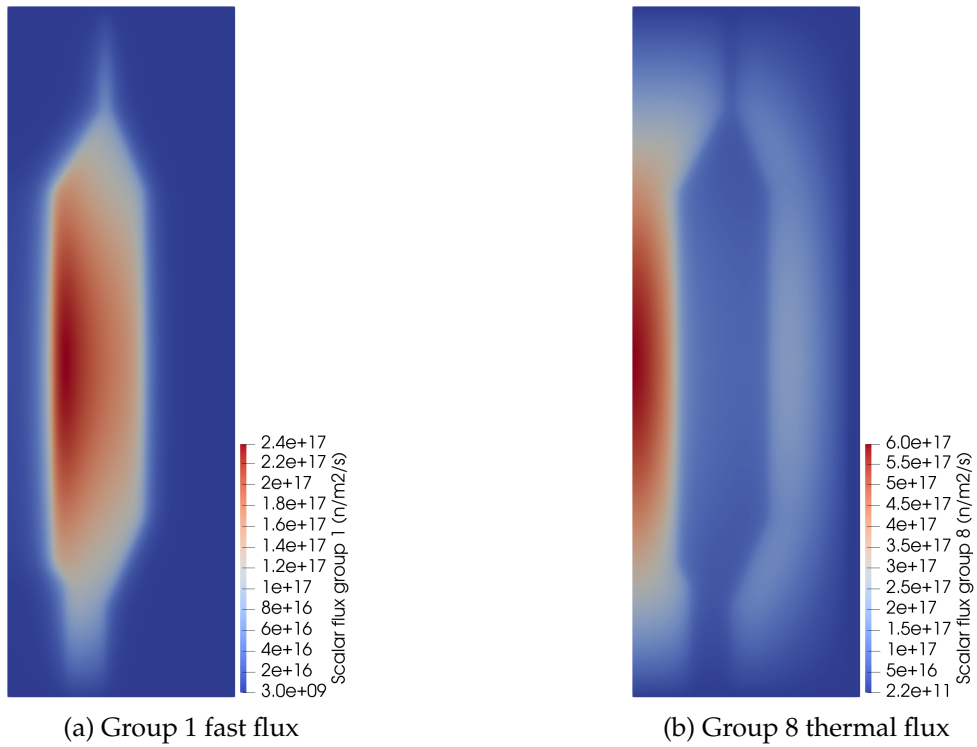
(a) Group 1 fast flux

(b) Group 8 thermal flux

Figure 15: Neutron flux distributions generated by Griffin, from [1]
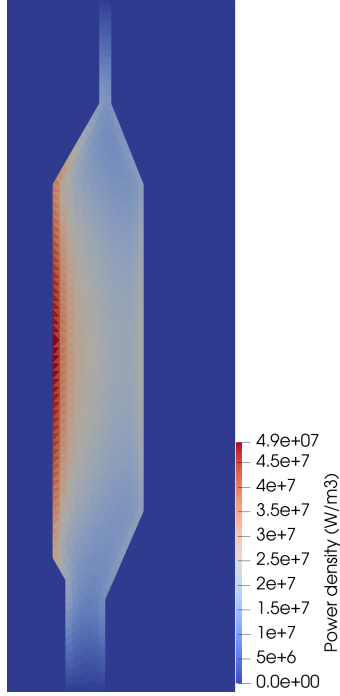
Figure 16: Power distribution generated by Griffin, from [1]

5.2.2 Integrated Plant Analysis using SAM

The SAM model for the Mk1-FHR was developed by Ahmed et al. [27] and coupled to Pronghorn by Daniel Nunez at Argonne National Laboratory as part of the SAM-Pronghorn coupling effort [4]. This effort was supported by the authors of this report. The coupling to Griffin was added in this study.

The SAM model of the Mk1-FHR is composed of the primary loop shown in Figure 17, with the core component replaced by the Pronghorn model. Velocity, temperature, and pressure are exchanged between SAM and Pronghorn at the core inlets and outlets, downstream of each model for velocity and temperature, and upstream for pressure. The flow in the core and the rest of the primary loop is driven by the pump in the SAM model at a fixed mass flow rate of 976 kg/s. For additional details about this model, we refer the reader to Ahmed et al. [27].

5.2.3 Coarse mesh thermal hydraulics model in Pronghorn

We developed a finite volume thermal hydraulics model of the core in Pronghorn [22, 23]. The underlying kernels and boundary conditions are part of the PINSFV implementation in the

31
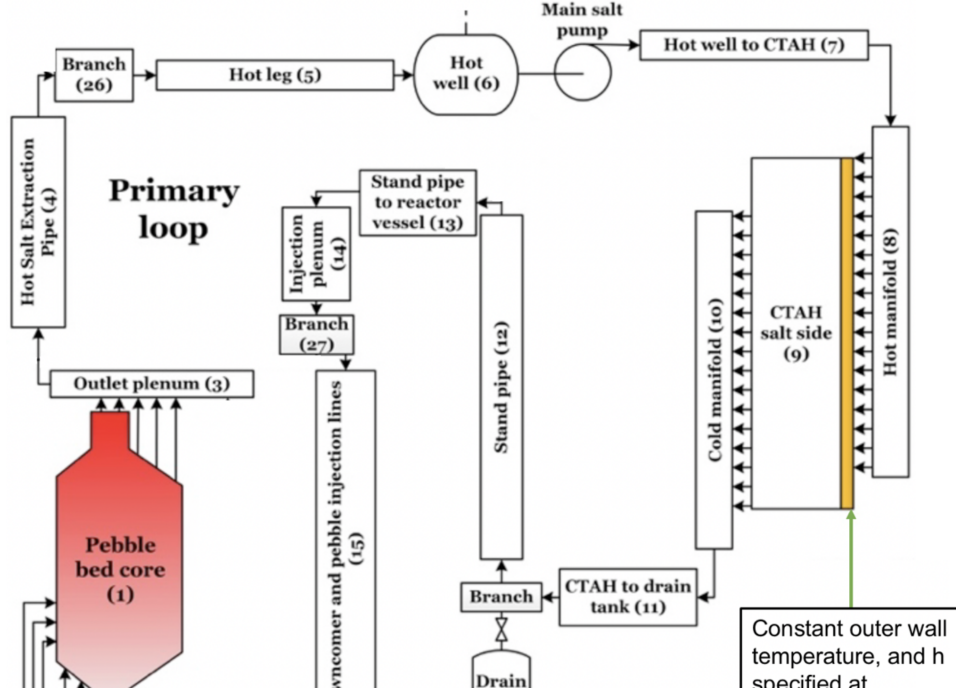
Figure 17: Primary loop for the Mk1-FHR, from [4]

MOOSE Navier-Stokes module. An additional variable material was created in Pronghorn to feed the variable values into the closure relations. We used the same set of closure relations and material properties as the ones selected for an earlier finite element model [5]. We are currently implementing additional closures to better model the influence of the wall on the porosity profile and to have more flexibility in the selection of the drag models for each region.

We present here our first results for a model with a plenum region. The majority of the flow exits the core through the plenum, while only 4% of the flow is bypass flow through the reflector. This leads to much better estimation of the pressure drop than previously [1]. The closures for the drag model in the outer reflector should still be revisited. Some form of tuning, against CFD or experiments, is expected to represent the outer reflector as a porous medium and obtain an appropriate bypass flow fraction. In the same input file, we model the heat transfer in the solid phase in the core and heat conduction in the solid components around the core, for a total of five equations: conservation of mass, x- and y-momentum (in RZ), and fluid and solid energy.

We first examine in Figure 18 the convergence of a few quantities of interest for the 2D-RZ reactor case, namely the pressure drop across the core, the bypass fraction and the maximum fluid temperature. We plot the relative difference to the most refined value. We verify first order

32
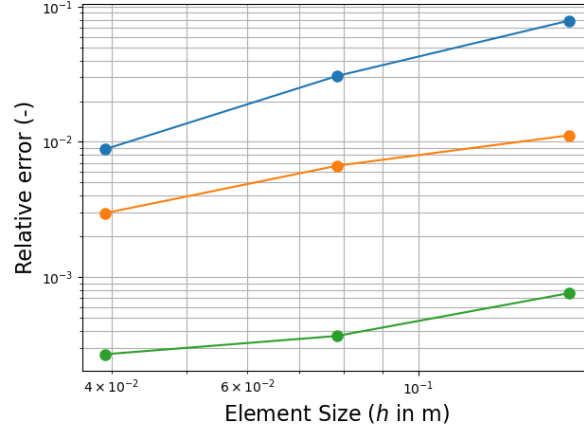
Figure 18: Convergence of the FHR uncoupled conjugate heat transfer simulation with refinement of the mesh, from [1].

convergence with RC interpolation of velocities and upwinding of advected quantities. These simulations are run uncoupled from the neutronics and the fuel performance models.

We show in Figure 19 the pressure field, the velocity magnitude, and the fluid and solid temperature fields for the 2D-RZ most refined uncoupled case. The velocity field shows artefacts at the porosity jump between the core (porosity of 0.4) and the plenum (porosity of 0.5). Such artefacts are also likely to be present between the core and the outer reflector (porosity of 0.14). The temperature profiles do not display artefacts even at porosity interfaces.

5.2.4   Pebble and TRISO particle heat conduction

The fuel temperature is obtained using a multiscale approach described and validated against high-fidelity models in [5]. We summarize the approach here and in Figure 20.

The porous media equations are solved to obtain the solid phase temperature. The solid phase temperature is used as the surface temperature of a 1D spherical pebble model. This 1D pebble model can be generated on every element of the mesh using the MOOSE MultiApp system. The porous media solid temperature is then the outer Dirichlet boundary condition to the 1D pebble model. The center boundary condition is naturally zero-gradient by spherical symmetry. The fuel matrix temperature in this pebble model is itself used as an outer Dirichlet boundary condition in a 1D Tri-structural Isotropic particle fuel (TRISO)+graphite matrix spherical model. A similar approach is used for graphite pebbles in the pebbled reflector in the periphery of the active region,
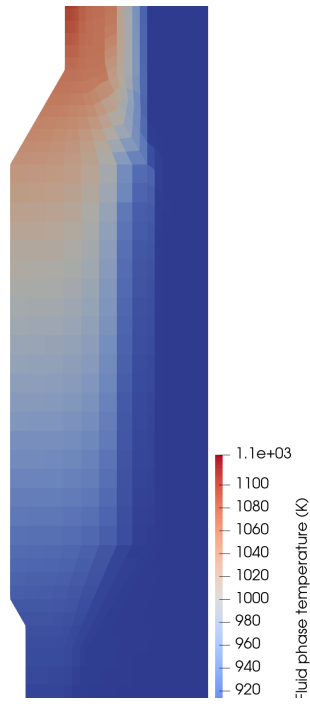
(a) Fluid temperature
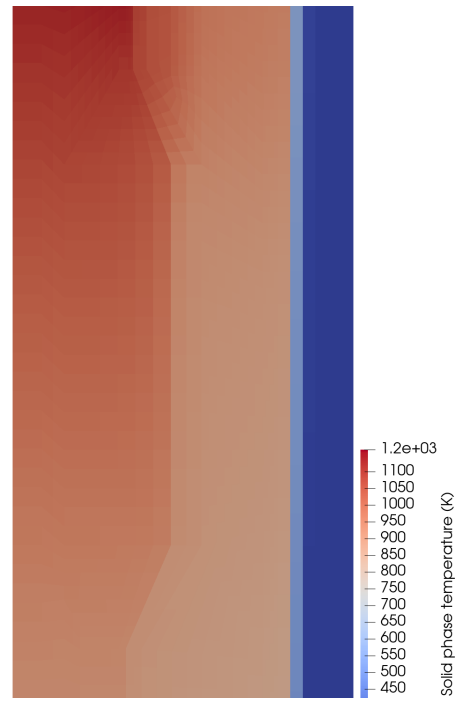
(b) Solid temperature
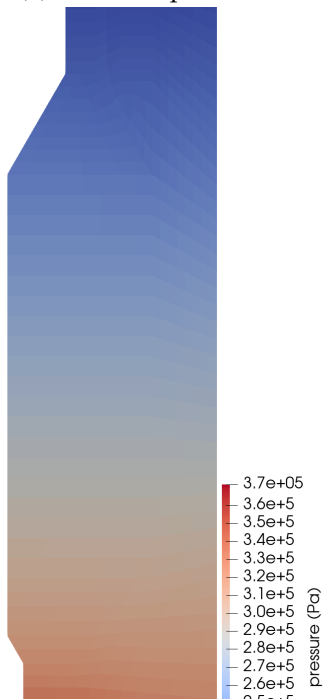
(c) Pressure
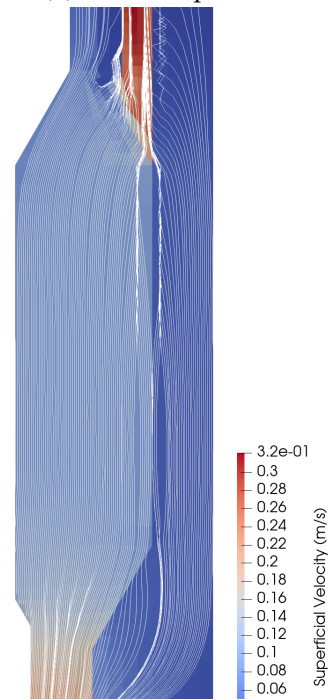
(d) Velocity

Figure 19: 2D RZ simulation results with an 80k elements mesh, from [1].
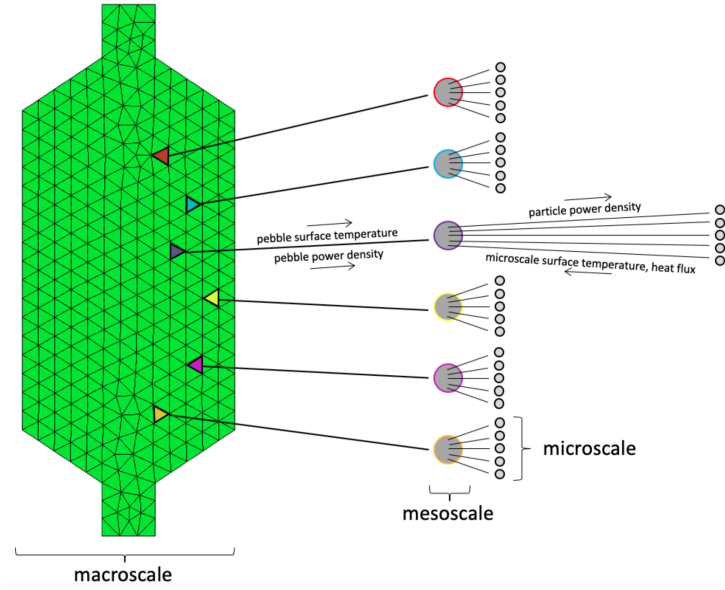
Figure 20: Multiphysics coupling between the thermal hydraulics model and the 1D fuel performance. A fuel performance application is spawned for each element (picture courtesy [5]).

using only a pebble heat conduction model. A coarse mesh reduction is used to limit the computational cost of this approach. It should be pointed out that during transients the use of Dirichlet boundary conditions does not guarantee that the pebble-scale models lose the same amount of heat over their boundary as predicted by the convection term in the porous medium fluid and solid energy equations [24].

5.2.5 Plant model with SAM-Pronghorn-Griffin coupling

All the previously described single physics models are then coupled to build a reference plant model. Griffin is chosen as the master app as it does not require timestepping in steady state solves, and it would be too expensive to run a transient Griffin model on every time step during a relaxation transient. The first level subapp is the Pronghorn core simulation, which has two subapps: one for the heat conduction calculations in the pebble and one for the SAM loop model. The SAM model necessitates shorter time steps than the coarse CFD at the core scale, which is why it is better to have it as a subapp to not constrain the timestep on more expensive solves. The pebble simulations are only coupled in one direction, receiving information from the macroscale solves.

We attempted the same strategy as in [4] to initialize the SAM-Pronghorn coupling, by run-

ning an initially uncoupled relaxation transient and using it as a restart file. The restart system used with three multiapps failed to load a variable for the Pronghorn solves, preventing restarts and forcing a staggering of each solve instead of a restart. Without the option for a restart, we attempted to pre-converge solves by staggering their execution in time. The coupling is currently too fragile and the system code diverged with the modified conditions coming from the coupled core simulation, even when the core simulation is converged beforehand. Relaxing the solves did not remedy the issue. Finally, failing SAM solves were improperly handled by the time stepper, which attempted to continue without running the system solves.

We will investigate these issues in future works. The improvements listed in subsection 6.4 will also improve the FHR reference plant efforts.

6. Molten Salt Fast Reactor reference plant

The MSFR model is based off of the MSFR design created under the Euratom EVOL (Evaluation and Viability of Liquid Fuel Fast Reactor Systems) and ROSATOM MARS (Minor Actinides Recycling in Molten Salt) projects [28]. It is a fast-spectrum reactor that produces 3 GW of thermal power using fuel dissolved in a LiF carrier salt.

Most parameters and material properties of the MSFR models are taken from [28]. That reference specifies a block-style geometry with all 90-degree angles, but we instead use a geometry with curved surfaces that more closely matches "Geometry II" from [29].

The reference plant model is a fully coupled multiphysics simulation of the MSFR including a neutronics solve, a multi-dimensional core model and which will include a primary and secondary loop thermal hydraulics model.

The MSFR models created are based on a 2D RZ simplification of the geometry. The incompressible flow coupled MSFR models were created for the Virtual Test Bed of the National Reactor Innovation Center [30] and served as a basis for this effort.

6.1 Neutronics model

The neutronics of the MSFR are modeled using the diffusion approximation with 6 energy groups. The neutronics solver Griffin is used, with the continuous finite element discretization. The multigroup cross sections were obtained using Serpent on an approximation of the geometry, with square borders.

Neutron precursors in a molten salt reactor travel through the primary, which influences the distribution of the neutron source. This is a source-advection-decay problem, which Pronghorn is well suited to solve. The fission source is passed from Griffin to Pronghorn, and predetermined yields are provided to Pronghorn to compute the precursor source term. After the fluid flow solve, Pronghorn transfers the neutron precursors concentration back to Griffin. Six delayed neutron precursors groups are used.

6.2   Incompressible flow model

The initial model of the MSFR is a closed loop with a volumetric pump and a volumetric heat exchanger. The volumetric coefficients for the heat transfer coefficient, the pump head and the friction loss in the heat exchanger are tuned to match a target temperature differential and a target pressure loss.

This model is hosted on the Virtual Test Bed of the National Reactor Innovation Center [31, 32]. The mesh used to solve both the neutronics and the thermal hydraulics problem is shown in Figure 21.
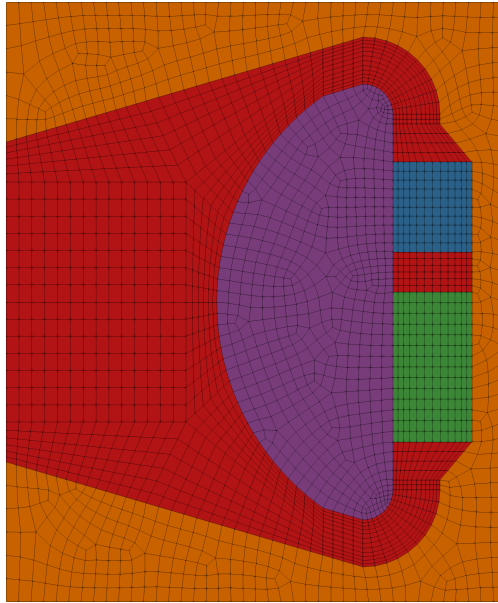
Figure 21: Mesh for the MSFR model

The fluid system includes conservation equations for fluid mass, momentum, and energy as well as the conservation of delayed neutron precursors. The effect of buoyancy is considered by using the Boussinesq approximation.

Practical simulations require modifications to the momentum equations in order to model the effects of turbulence without explicitly resolving the turbulent structures. Here, we will apply the Reynolds-averaging procedure and the Boussinesq hypothesis so that the effect of turbulent momentum transfer is modeled with a term analogous to the viscous shear.

Boundary conditions include standard velocity wall functions at the walls to account for the non-linearity of the velocity in the boundary layer given the coarse mesh of the model and sym-

metry at the center axis of the MSFR.

6.2.1 Algebraic Turbulence model

The Reynolds stress tensor using the RANS formulation is

$$\nabla \cdot \rho \vec{u} u = -\frac{\partial}{\partial x} P + (\mu + \mu_t) \nabla^2 u + f_{\text{fric},x} \tag{12}$$

where μ_t is the turbulent viscosity.

In this initial effort, a zero-equation model based on the mixing length model is used. In this model the turbulent viscosity is defined as:

$$\mu_t = \rho \cdot l_m{}^2 \cdot |2\overline{\overline{S}} : \overline{\overline{S}}| \tag{13}$$

and

$$\overline{\overline{S}} = 0.5 \cdot \left(\nabla u + \nabla u^t \right) \tag{14}$$

The standard Prandtl's mixing length model dictates that l_m has a linear dependence on the distance to the nearest wall. However, for this simulation we implement a capped mixing length model [33] that defines the mixing length as

$$l_m = \kappa y_d \quad if : \kappa y_d < \kappa_0 \delta \tag{15}$$

$$l_m = \kappa_0 \delta \quad if : \kappa y_d \geq \kappa_0 \delta \tag{16}$$

where $\kappa = 0.41$ is the Von Karman constant, $\kappa_0 = 0.09$ as in Escudier's model and δ has length units and represents the thickness of the velocity boundary layer.

6.2.2 Validation

The validation of the model is performed by comparison to Nek5000 for an open loop case. The flow distribution in the volumetric heat exchanger and pump are not relevant for validation, and the capacity to simply set the same boundary conditions makes the comparison simpler to perform. The Nek5000 simulations were lead by Jun Fang from Argonne National Laboratory.

In the zero equation turbulence model formulation, the mixing length grows linearly up to a distance dependent on a parameter δ, which is representative of the boundary layer thickness. Using Nek5000 data, this parameter is tuned for Reynolds number 1M. This dimensionless number value is of particular interest as it is representative of the operational regime of the MSFR.

The velocity magnitude and pressure fields obtained in Nek5000 and Pronghorn are shown in Figure 22

The parameter δ for this simulation is tuned to 0.1 m. This is the result of a parametric study focused on minimizing the error of the velocity field, specifically at the minimum core radius. The velocity magnitude at the minimum core radius is plotted in Figure 23. The behavior of the velocity fields are similar between both codes. The momentum diffusion at the center of the reactor is slightly higher in Pronghorn.

Pronghorn's total simulation time is 15 minutes using 4 cores. Even though Pronghorn offers a low computation time, it is important to note that in a coarse mesh CFD code like Pronghorn, the tuning done in this section is dependent on the mesh due to the numerical diffusion.

The importance of this numerical diffusion could be reduced in the future with the implementation of second order upwind discretization (currently using linear upwind) and the use of non-orthogonal correctors for meshes with a degree of skewness.

6.2.3 Results

Using the tuned parameter for the turbulence model, the Griffin-Pronghorn model predicts the steady-state velocity and temperature fields in the closed loop MSFR shown in Figure 24.

The maximum temperature of the simulation is around 1280 K. Note that despite some momentum diffusion occurring in the reactor core, the flow at the center is low where the power density has a maximum. This leads to high temperatures in the center region, which in turn drives the flow vertically through buoyancy. We can also observe a low degree of flow separation along the shield wall. The salt here will be in direct contact with the core structure and may challenge the temperature limitations of the structural material.

The main difference between the velocity fields of the previous section and this section is the augmented flow in the core of the reactor. This is a results of the buoyant force coupling.

The Griffin-Pronghorn model also provides the steady-state distributions of the delayed neu-
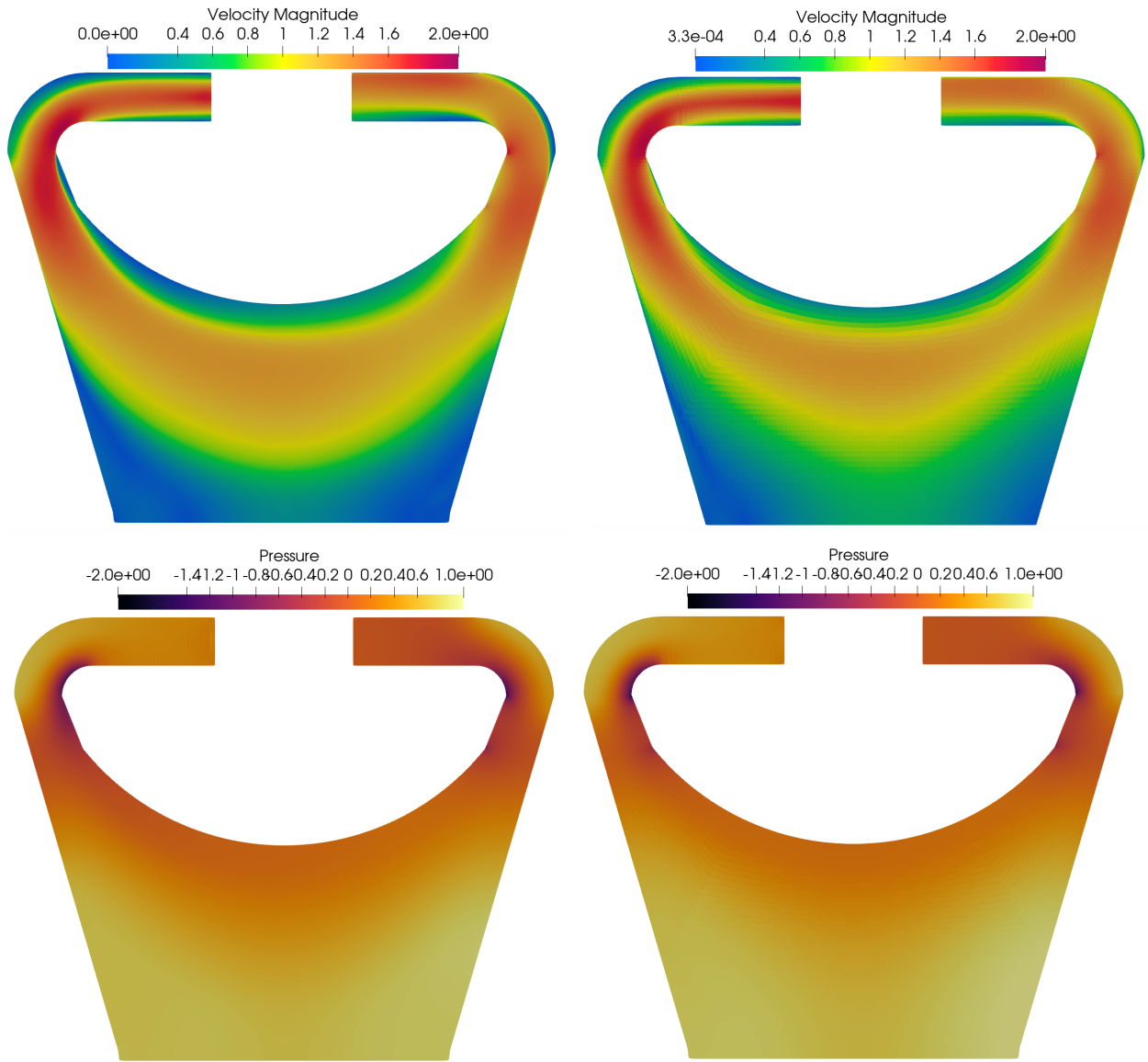
Figure 22: Fluid velocity (top) and pressure (bottom) field for uncoupled open loop MSFR simulations using Nek5000 (left) and Pronghorn (right), from [6]
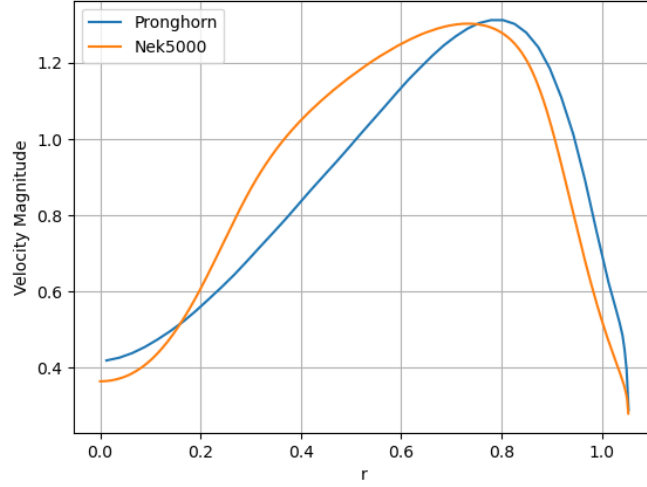
Figure 23: Velocity magnitude at the minimum core radius, from [6]

tron precursors. The distributions of 3 (of the 6) groups are shown in Figure 25. For this plot, the concentrations have been normalized so that the maximum value for each group is unity.

Group 6 has the shortest half-life (240 ms), and its behavior is dominated by radioactive decay rather than advection. Consequently, its distribution closely matches the fission distribution, with a distinct peak in the middle of the core. Groups 3 and 1 have longer half-lives (6 and 52 s, respectively), so they are more readily advected and diffused by the fluid. In fact, the distribution of the slowest decaying group, group 1, is nearly uniform

6.3 Weakly compressible fluids core model

6.3.1 Updates to the model

The model was adapted for the weakly compressible implementation of the finite volume Navier Stokes equation (WCNSFV). Because the material properties such as viscosity and density are temperature dependent in that model, they can no longer be assumed to be flat over an element. And as the finite volume method requires material property evaluations on model faces, all the fluid material properties were transitioned from constants in the incompressible model to functor material properties.

This required the development of `GeneralFunctorFluidProperties`, which centralizes the definition of functor material properties. This material takes a fluid property object and assigns

(a) Velocity field
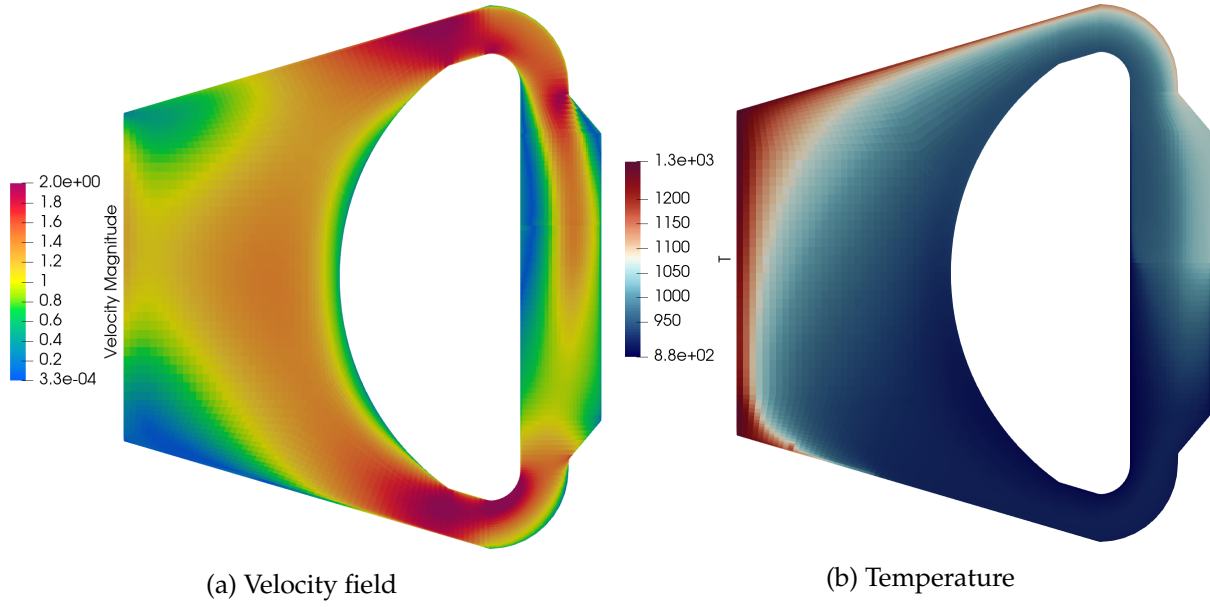
(b) Temperature

Figure 24: Fluid velocity and temperature for the incompressible flow model with coupled neutronics, from [6]
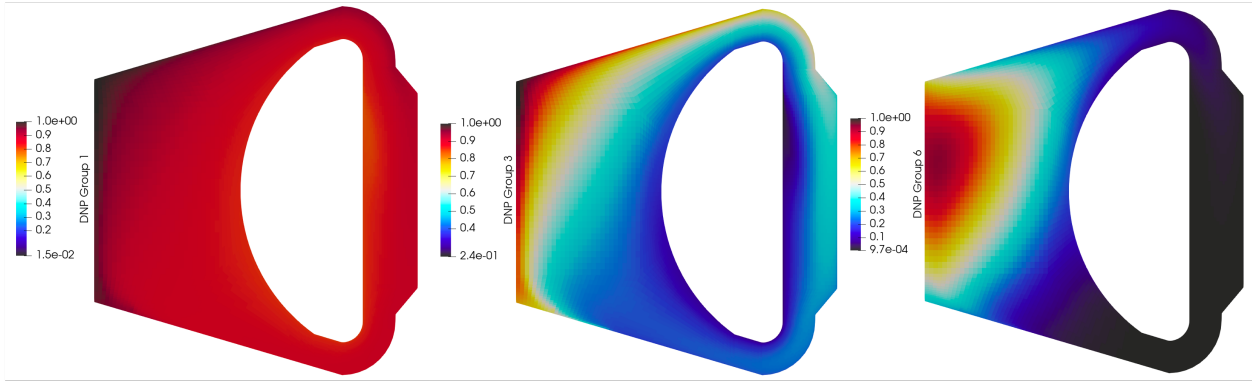
Figure 25: Distribution of neutron precursors in the solution, from [6]

a functor calling this fluid property object's routines, such as `cp_from_v_e(...)`, with the local temperature and pressure.

Other updates to the model include:

- the fluid properties are now computed according to the properties specified in [34]. The viscosity correlation in this reference becomes singular below 762K so a more resilient formulation of these fluid properties should be found. These fluid properties do not include the pressure dependence as well, only the temperature dependence.

- an improvement in computing the temperature differential across the heat exchanger, which

now no longer assumes a uniform temperature and mass flow profile across the surface of the volumetric heat exchanger

6.3.2 Results

We show in Figure 26 the velocity, temperature, pressure and power density fields, as well as the viscosity and density material property. We can see that:

- the mean pressure has drifted in unphysical realms. This is an artefact of the material properties having no pressure dependence and the closed loop model which does not have pressure boundary condition. Adding a constraint on the mean pressure, similar to what is done for the incompressible model, may be required for such cases.

- there are two large recirculation zones at the top of the core and one in the volumetric pump. While the two former is a known potential defect of this design [35], the latter is an artefact of modeling the pumps using a geometric component. 3D analysis of the system should be lead to examine if the recirculation zones are stable in a non-symmetric configuration.

- the core recirculation zone near the top creates a peak temperature, slightly above 1500K. Such a temperature near the wall would challenge materials too much.

- the correlation for viscosity is close or beyond its range of validity in lower temperature regions, leading to a large viscosity in significant parts of the core. This should be fixed by obtaining more robust fluid property correlations for this salt

In Figure 27, we show the precursor concentration for group 1 and 6, the shortest and longest lived precursor groups. As seen previously for the incompressible model, the longer lived precursors are not affected much by the flow distribution, while the shorter lived ones show significant gradients across the core. In particular, the recirculation zone has sufficiently low power density that the shorter lived precursors have a low concentration in that region.
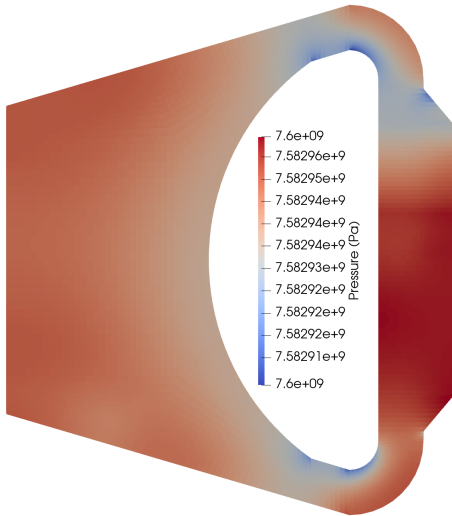
Further efforts will compare these results with the incompressible model to assess the necessity or lack thereof of using a weakly compressible approximation.
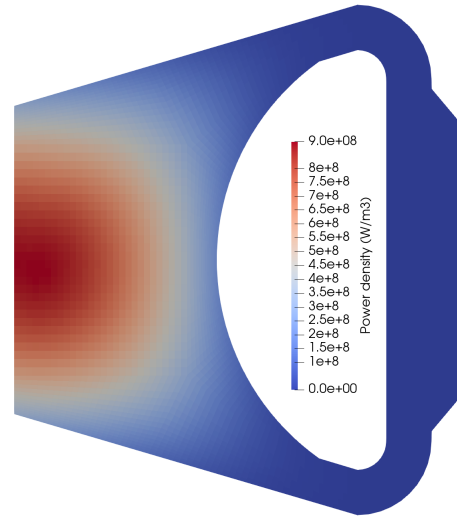
44
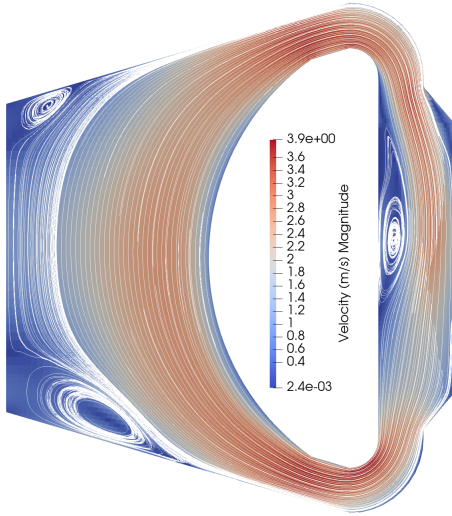
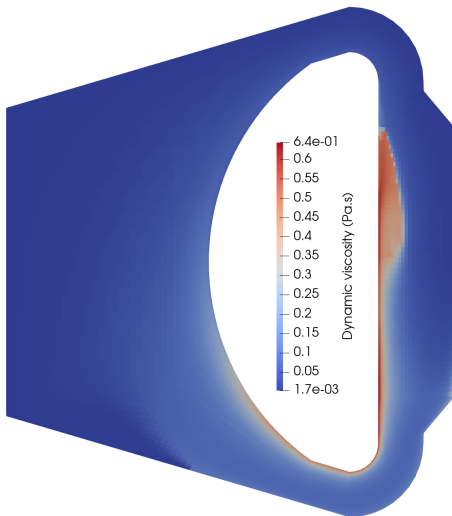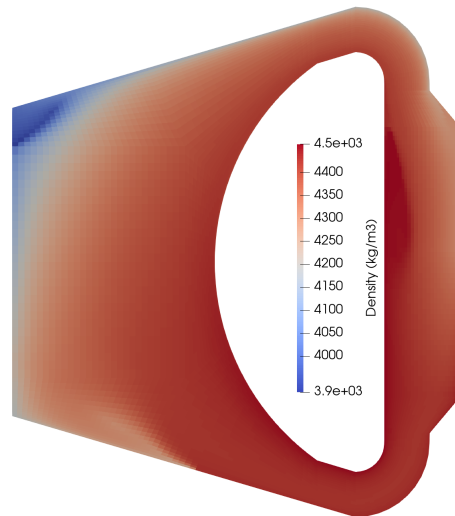(a) Velocity magnitude and streamlines
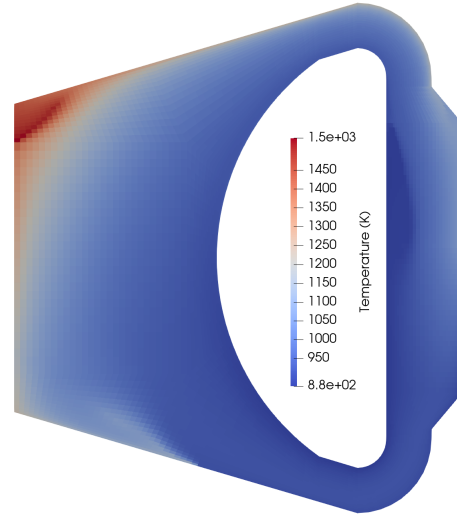
(b) Temperature

(c) Pressure

(d) Power density

(e) Viscosity

(f) Density

Figure 26: MSFR results with temperature dependent material properties

(a) Group 1   (b) Group 6
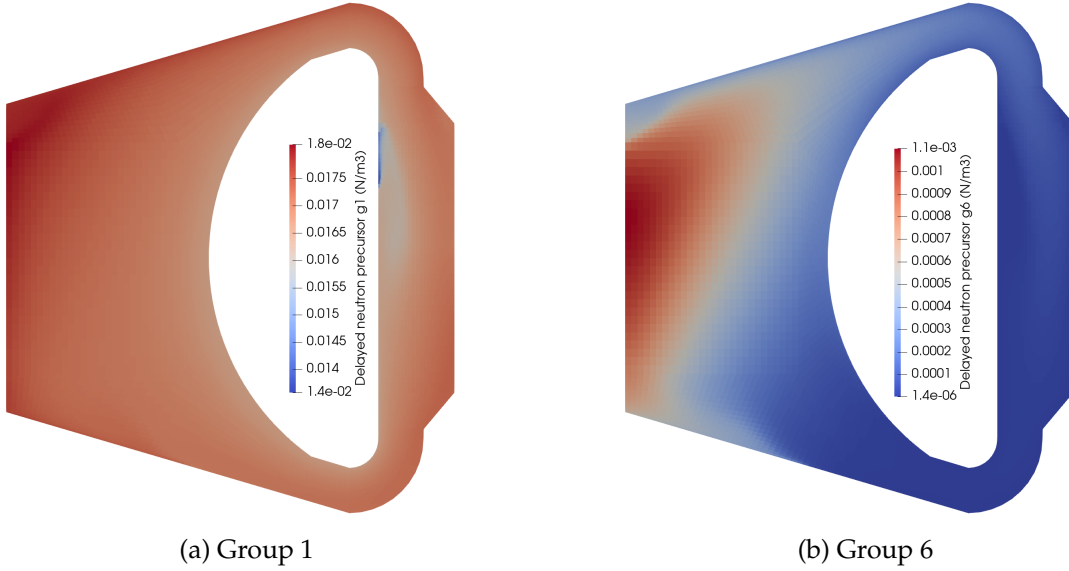
Figure 27: Precursor concentrations in the MSFR coupled model

6.4 Current status of the plant model and SAM-Pronghorn coupling

SAM-Pronghorn coupling was demonstrated for a fluoride high temperature reactor [4]. We currently do not have a MSFR SAM model, and limited efforts to adapt the FHR loop from that model (without precursors) or the MSRE loop model are yet to be successful. However, significant development progress has been made towards that goal.

- The WCNSFV formulation simplifies the coupling at interfaces as the same equation of state is now used in both SAM and Pronghorn. This removes one need for normalization. The coupling should still be done using mass/energy flow rates to be sure to observe proper conservation when the outlet of the core is not fully developed / has a transverse temperature gradient.

- More development should be done on the Pronghorn side to accommodate mass flow and energy flow boundary conditions. An interface to the Dirichlet boundary conditions on velocity and temperature should be created to account for non-uniform pressure and temperature at inlets and for area normalization in 3D models (when the mesh surface is not the analytical surface). An alternative would be to directly use flux boundary conditions for the mass, momentum and energy equations.

46

- The user object for Rhie Chow interpolation will allow postprocessors, which tally the mass and energy flow rates on the edges of the Pronghorn simulation, to use the same velocity interpolation method as the kernels are using. This is an ongoing development and will enable verificiation of the conservation of mass and energy across the coupling.

REFERENCES

[1] S. Schunert, G. Giudicelli, A. Lindsay, PaoloBalestra, S. Harper, R. Freile, M. Tano, and J. Ragusa, "Deployment of the finite volume method in pronghorn for gas- and salt-cooled pebble-bed reactors," Tech. Rep. INL/EXT-21-63189, Idaho National Laboratory, 2021.

[2] S. King, E. Kappes, M. Marciniak, D. Nguyen, Y. Hassan, and V. Ugaz, "Pressure Drop Measurements in a Versatile Experimental Facility of Packed Spheres," *American Nuclear Society*, vol. 117, no. 1, pp. 1713–1716, 2017.

[3] J. Lee, S. King, Y. A. Hassan, T. Nguyen, P. Balestra, S. Schunert, and G. Giudicelli, "Pronghorn Porous Media Model Validation with Pressure Drop Measurements," *NURETH-19*, 2022 [under review].

[4] R. Hu, D. Nunez, G. Hu, L. Zou, G. Giudicelli, D. Andrs, and S. Schunert, "Development of an integrated system- and engineering-scale thermal fluids analysis capability based on sam and pronghorn," Tech. Rep. ANL/NSE-21/36, Argonne National Laboratory, 2021.

[5] A. Novak, *Multiscale Thermal-Hydraulic Methods for Pebble Bed Reactors*. PhD thesis, University of California Berkeley, 2020.

[6] National Reactor Innovation Center, "Virtual test bed documentation." `https://mooseframework.inl.gov/virtual_test_bed/`, 2021.

[7] M. Quintard, "Introduction to Heat and Mass Transport in Porous Media," Tech. Rep. STO-EN-AVT-261, Von Karman Institute for Fluid Dynamics, Rhode Saint Genese, Belgium, 2015.

[8] J. Bear, *Dynamics of Fluids in Porous Media*. Courier Corporation, 2013.

[9] D. Takhanov, "Forchheimer Model for Non-Darcy Flow in Porous Media and Fractures," tech. rep., Imperial College London, London, United Kingdom, 2011.

[10] A. E. Slaughter, C. J.Permann, J. M. Miller, B. K. Alger, and S. R. Novascone, "Continuous integration, in-code documentation, and automation for nuclear quality assurance conformance," *Nuclear Technology*, vol. 0, no. 0, pp. 1–8, 2021.

[11] A. J. Novak, S. Schunert, R. W. Carlsen, P. Balestra, D. Andrs, J. Lee, J. Kelly, R. N. Slaybaugh, R. C. Martineau, and H. Gougar, "Pronghorn Theory Manual," Tech. Rep. INL/EXT-18-44453-Rev001, Idaho National Laboratory, Idaho Falls, ID, USA, 2020.

[12] K. A. Maupin, L. P. Swiler, and N. W. Porter, "Validation Metrics for Deterministic and Probabilistic Data," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 3, no. 3, p. 031002, 2018. https://doi.org/10.1115/1.4042443.

[13] A. Montillet, E. Akkari, and J. Comiti, "About a Correlating Equation for Predicting Pressure Drops through Packed Beds of Spheres in a Large Range of Reynolds Numbers," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 4, pp. 329–333, 2007. https://doi.org/10.1016/j.cep.2006.07.002.

[14] G. Cumming, F. Fidler, and D. Vaux, "Error Bars in Experimental Biology," *Journal of Cell Biology*, vol. 177, no. 1, pp. 7–11, 2007. https://doi.org/10.1083/jcb.200611141.

[15] G. Cumming, "Inference by Eye: Reading the Overlap of Independent Confidence Intervals," *Statistics in Medicine*, vol. 28, no. 2, pp. 205–220, 2009. https://doi.org/10.1002/sim.3471.

[16] D. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer, *Introduction to Statistics*. Citeseer, 2017.

[17] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, vol. 6. Springer, 2016.

[18] S. Zhang, X. Zhao, and S. Bayyuk, "Generalized formulations for the rhie–chow interpolation," *Journal of Computational Physics*, vol. 258, pp. 880–914, 2014.

[19] M. Nordlund, M. Stanic, A. Kuczaj, E. Frederix, and B. Geurts, "Improved piso algorithms for modeling density varying flow in conjugate fluid–porous domains," *Journal of Computational Physics*, vol. 306, pp. 199–215, 2016.

[20] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, "Gen-foam: a novel openfoam® based multi-physics solver for 2d/3d transient analysis of nuclear reactors," *Nuclear Engineering and Design*, vol. 294, pp. 24–37, 2015.

[21] A. Novak *et al.*, "Pronghorn theory manual," Tech. Rep. INL/EXT-18-44453, Idaho National Laboratory, 2018.

[22] A. J. N. Guillaume Giudicelli, Abdalla Abou-Jaoude, "Coupled multiphysics simulations of the mk1-fhr in the virtual test bed," in *Transactions of the American Nuclear Society*, vol. 125, 2021.

[23] G. Giudicelli, A. Lindsay, P. Balestra, R. Carlsen, J. Ortensi, D. Gaston, M. DeHart, A. Abou-Jaoude, and A. J. Novak, "Coupled multi-physics transient simulations of the mk1-fhr reactor using the moose framework," in *International Conference on Mathematics and Computation for Nuclear Engineering*, 2021.

[24] S. Schunert, G. Giudicelli, P. Balestra, JavierOrtensi, R. Freile, and L. Harbour, "Nrc multi-physics analysiscapability deployment fy2021 - part 1," Tech. Rep. INL/EXT-20-60777, Idaho National Laboratory, 2021.

[25] J. Leppänen,, *Development of a New Monte Carlo Reactor Physics Code*. PhD thesis, Helsinki University of Technology, 2007.

[26] X. Wang, *Coupled neutronics and thermal-hydraulics modeling for pebble-bed Fluoride-Salt-Cooled, High-Temperature Reactor*. PhD thesis, University of California Berkeley, 2018.

[27] K. Ahmed, R. Scarlat, and R. Hu, "Benchmark simulation of natural circulation cooling systems with salt working fluid using sam," tech. rep., Argonne National Laboratory, 2017.

[28] Brovchenko, Mariya, Kloosterman, Jan-Leen, Luzzi, Lelio, Merle, Elsa, Heuer, Daniel, Laureau, Axel, Feynberg, Olga, Ignatiev, Victor, Aufiero, Manuele, Cammi, Antonio, Fiorina, Carlo, Alcaro, Fabio, Dulla, Sandra, Ravetto, Piero, Frima, Lodewijk, Lathouwers, Danny, and Merk, Bruno, "Neutronic benchmark of the molten salt fast reactor in the frame of the evol and mars collaborative projects," *EPJ Nuclear Sci. Technol.*, vol. 5, p. 2, 2019.

[29] H. Rouch, O. Geoffroy, P. Rubiolo, A. Laureau, M. Brovchenko, D. Heuer, and E. Merle-Lucotte, "Preliminary thermal–hydraulic core design of the molten salt fast reactor (msfr)," *Annals of Nuclear Energy*, vol. 64, pp. 449–456, 2014.

[30] A. Abou-Jaoude, G. Giudicelli, D. Gaston, R. Freile, S. Balderrama, and C. Permann, "Fy21 key technical developments in advanced reactor modeling," Tech. Rep. M4RC-21IN0206024, National Reactor Innovation Center, 2021.

[31] R. Freile, S. Harper, G. Giudicelli, and A. Abou-Jaoude, "Coupled griffin and pronghorn simulation of the molten salt fast reactor (msfr) for the virtual test bed," in *Transactions of the American Nuclear Society*, vol. 125, 2021.

[32] A. Abou-Jaoude, D. Gaston, G. Giudicelli, B. Feng, and C. Permann, "Overview of the virtual test bed," in *Transactions of the American Nuclear Society*, vol. 125, 2021.

[33] M. Escudier, *The Distribution of Mixing-Length in Turbulent Flows Near Walls*. PhD thesis, Imperial College, 1966.

[34] C. Tripodo, A. Di Ronco, S. Lorenzi, and A. Cammi, "Development of a control-oriented power plant simulator for the molten salt fast reactor," *EPJ Nuclear Sciences & Technologies*, vol. 5, p. 13, 01 2019.

[35] B. Deng, Y. Cui, and J. e. a. Chen, "Core and blanket thermal–hydraulic analysis of a molten salt fast reactor based on coupling of openmc and openfoam," *NUCL SCI TECH*, vol. 85, 2020.