INL/EXT-21-65482

Improvement of Numerical Methods in Pronghorn

December 2021

Nuclear Energy Advanced Modeling and Simulation M2 Milestone December 2021

Alexander Lindsay[1], Mauricio Tano[2], Guillaume Giudicelli[1], Peter German[1], and Sebastian Schunert[3]

[1]*COMPUTATIONAL FRAMEWORKS*
[2]*THERMAL FLUID SYSTEMS METHODS & ANALYSIS*
[3]*REACTOR PHYSICS METHODS AND ANALYSIS*

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/EXT-21-65482

Improvement of Numerical Methods in Pronghorn

Nuclear Energy Advanced Modeling and Simulation M2 Milestone
December 2021

Alexander Lindsay[1], Mauricio Tano[2], Guillaume Giudicelli[1], Peter German[1], and Sebastian Schunert[3]

[1]**COMPUTATIONAL FRAMEWORKS**
[2]**THERMAL FLUID SYSTEMS METHODS & ANALYSIS**
[3]**REACTOR PHYSICS METHODS AND ANALYSIS**

December 2021

Idaho National Laboratory
Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of _____
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Page intentionally left blank

SUMMARY

During Fiscal Year 2021, the Finite Volume Method (FVM) was deployed in Pronghorn to provide improved efficiency, stability, and accuracy for coarse-mesh, thermal-hydraulics problems. While the main goals of FY 2021 were met, several issues emerged from the early deployment of the FVM in Pronghorn. These issues were:

- The compressible and incompressible formulations that were implemented are inadequate for many nuclear reactor flow problems.
- The omission of terms accounting for the porosity and Darcy-Forchheimer body force discontinuities in the Rhie-Chow interpolation leads to oscillations in pressure and velocity at these discontinuities.
- The FVM lacks a correction for skew for computing accurate pressure gradients, leading to loss of accuracy in regions with skewed elements.
- The FVM currently uses a monolithic solver. Monolithic solvers have issues dealing with the saddle-point nature of the discretized fluid equations. This can result in poor linear solver convergence if direct factorization is not used. Direct factorization incurs a large memory consumption and does not scale efficiently as problems grow in size.

These four issues are addressed in this report. In particular, we report the completion of the following tasks:

- Implementation of a weakly compressible formulation in the MOOSE Navier-Stokes module.
- Implementation of Moukalled's method for including body forces in the Rhie-Chow interpolation. Additionally, we implemented an algorithm to smooth the porosity using repeated interpolation and reconstruction

operations and modified the pressure interpolation to faces in the presence of strong gradients in body forces.

- Implementation of skew correction for internal face values and for Green-Gauss gradient computation.
- Preliminary implementation of a Semi Implicit Method for Pressure Linked Equations (SIMPLE) segregated solver.

Page intentionally left blank

CONTENTS

FIGURES

TABLES

ACRONYMS

FHR	Fluoride-Salt-Cooled High-Temperature Reactor
FV	Finite Volume
FVM	Finite Volume Method
HLLC	Harten-Lax-Van Leer-Contact
KNP	Kurganov-Noelle-Petrova
KT	Kurganov-Tadmor
MMS	Method of Manufactured Solution
MOOSE	Multiphysics Object-Oriented Simulation Environment
RCI	Rhie-Chow interpolation
SIMPLE	Semi Implicit Method for Pressure Linked Equations

Page intentionally left blank

1. Introduction

Pronghorn [1] is a coarse-mesh, thermal-hydraulics code based on the Multiphysics Object-Oriented Simulation Environment (MOOSE) [2]. It focuses on engineering-scale fluid-flow simulations [3] in support of advanced reactor design and transient analysis. The goal is to provide a computationally efficient simulation tool for a multidimensional, transient full-core analysis of a variety of advanced reactors, including pebble-bed reactors (both salt- and gas-cooled), molten-salt reactors, and liquid metal fast reactors.

Pronghorn roughly aligns with the goals of the GenFOAM [4] computer code that is built on OpenFOAM instead of MOOSE. The advantage of Pronghorn is that it natively interacts with the Griffin neutronics (combination of the previous Rattlesnake, Proteus, and MC2-3 capabilities) [5–7] and BISON fuel performance codes [8], both of which are built on MOOSE as well. These tools are supported within the Nuclear Energy Advanced Modeling and Simulation program and used within the U.S. nuclear industry and the Nuclear Regulatory Commission.

Pronghorn is a consumer of capabilities that are implemented within MOOSE and some other applications. The ability to inherit capabilities from *submodules* is a hallmark of the MOOSE framework and constitutes a great advantage because it emphasises code reuse. Pronghorn relies on two major *submodules*: first, it relies on MOOSE's Navier-Stokes module [9–11] for the finite-volume discretization that is the backbone of Pronghorn, and second, it relies on the subchannel *submodule* for subchannel capabilities used mainly for liquid-metal fast reactor analysis.

During Fiscal Year 2021, the FVM was deployed in Pronghorn to provide improved efficiency, stability, and accuracy for coarse-mesh, thermal-hydraulics problems [12]. While the main goals of fiscal year 2021 were met, several issues emerged from the early deployment of the FVM in Pronghorn. These issues were:

- In the initial deployment of the FVM, only fully compressible and incompressible equations were implemented. Neither of these two approaches is adequate for reactor analysis. The incompressible formulation is only valid if density changes remain smaller than a few percent (with the tolerance being dependent on the problem). The density changes are definitely violated in gas-cooled reactors and are most likely violated in salt-cooled reactors. On the other hand, fully compressible formulations perform well at large Mach numbers,

but the vast majority of relevant flows in nuclear reactors have small Mach numbers and are often buoyancy-driven with large Rayleigh numbers. In these flow regimes, weakly compressible (also referred to as low Mach) formulations perform much better.

- The FVM in MOOSE uses a collocated formulation. In the collocated formulation, the Rhie-Chow interpolation (RCI) is used to suppress the pressure checkerboard pattern. The initial FVM implements a standard RCI that does not make accommodations for discontinuities in porosity and body forces (mainly Darcy/Forchheimer drag forces). This omission leads to oscillations in pressure and velocity at discontinuities of porosity and body forces.

- The FVM lacks a correction for skew for computing accurate pressure gradients, leading to loss of accuracy in regions with skewed elements.

- The FVM currently uses a monolithic solver for the Navier-Stokes equations. However, when using primitive variables in incompressible or nearly incompressible flows, the resulting equations are a saddle point problem, making the direct solution very challenging. In addition, the monolithic solver tends to consume more memory and scale poorly with respect to the CPU, making larger models (in excess of 10,000 elements) too costly to be executed on a laptop or workstation.

The four issues identified in the paragraph above are addressed in this report. In particular, we report the completion of the following tasks:

- Implementation of a weakly compressible formulation in the MOOSE Navier-Stokes module.

- Implementation of Moukalled's method [13] for including body forces in the RCI. Additionally, we implemented an algorithm to smooth the porosity using repeated interpolation and reconstruction operations and modified the pressure interpolation to faces in the presence of strong gradients in body forces.

- Implementation of skew correction for internal face values and Green-Gauss gradient computation following Ref. [13].

- Preliminary implementation of a SIMPLE segregated solver.

2. Weakly Compressible Navier-Stokes Equations

The compressible, porous Navier-Stokes equations with Darcy and Forchheimer drag terms are frequently used as a model for the coarse-mesh, thermal-hydraulics analysis of single-phase fluid behavior in reactors. We adopt the set of equations from Ref. [1] but write it in terms of the superficial velocity:

$$\begin{aligned} \gamma\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) &= 0 \\ \frac{\partial\rho\vec{v}}{\partial t} + \nabla\cdot\left(\gamma^{-1}\rho\vec{v}\otimes\vec{v}\right) &= -\gamma\nabla p + \gamma\rho\vec{g} - W\rho\vec{v} \\ \gamma\frac{\partial\rho E}{\partial t} + \nabla\cdot(\rho H\vec{v}) - \nabla\cdot\left(\kappa_f\nabla T\right) + \alpha(T-T_s) &= 0, \end{aligned} \tag{1}$$

where ρ is the fluid density, γ is porosity, t is time, $\vec{v}$ is the superficial velocity, p is pressure, $\vec{g}$ is the vector of gravity, W is the drag coefficient that consists of a Darcy and Forchheimer contribution, E is the total internal energy, H is the total enthalpy, T is the fluid temperature, α is the volumetric heat transfer coefficient between solid and fluid, and T_s is the solid temperature. The solid temperature is determined from a solid heat conduction equation given by:

$$(1-\gamma)\rho_s c_{p,s}\frac{\partial T_s}{\partial t} - \nabla\cdot(k_s\nabla T_s) - \alpha(T-T_s) = \dot{q}''', \tag{2}$$

where ρ_s is the solid density, $c_{p,s}$ is the specific heat capacity of the solid, k_s is the thermal conductivity of the solid, and $\dot{q}'''$ is the distributed heat source. Note, that Equation 1 is referred to as a compressible or fully compressible equation even though it already neglects gravity and viscous work terms in the fluid energy equation. These terms have never been included in Pronghorn problem formulations.

Previous work in Reference [12] used conservative variables $(\rho, \rho\vec{v}, \rho E)$ and the Harten-Lax-Van Leer-Contact (HLLC), Kurganov-Tadmor (KT), or Kurganov-Noelle-Petrova (KNP) discretization method. Collectively, we refer to these methods as the fully compressible approach. The fully compressible approach works well for problems with an appreciable Mach number. However, fully compressible (in particular characteristic-based) approaches may not naturally support body forces and can be overly diffusive at small Mach numbers. Thus, at small Mach

numbers and especially for buoyancy driven flows with large Rayleigh numbers and gradients, the fully compressible approach does not perform adequately.

In addition to the fully compressible approach, an incompressible Finite Volume (FV) discretization with a Boussinesq buoyancy term was implemented in Pronghorn and tested [12]. While incompressible models work well for a wide range of problems, the large temperature differences present in advanced reactors push the incompressibility assumption close to or beyond its range of applicability.

Therefore, we implemented a weakly compressible formulation of the Navier-Stokes equations that allows capturing the most salient effects of density changes while avoiding several pitfalls. The theory of a weakly compressible formulation is detailed in Reference [14], but it is referred to here as a low-Mach approximation. In the remainder of this document, we use *weakly compressible* to refer to this new formulation, but it is understood to be synonymous with low-Mach approximation.

There are three main differences between the fully compressible and weakly compressible equations. First, the low-Mach equations are identical to Equation 1 except that the fluid energy equation is modified to:

$$\gamma \frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho h \vec{v}) - \nabla \cdot \left(\kappa_f \nabla T\right) + \alpha (T - T_s) = 0, \tag{3}$$

where e and h are the internal energy and enthalpy, respectively, and do not include kinetic energy contributions. Second, we solve for primitive variables $(p, \vec{v}, T)$ instead of conservative variables. Third, instead of discretizing the equations using HLLC, KP, or KNP, we use a standard collocated discretization scheme with RCI.

2.1 Differentially Heated Cavity

The weakly compressible implementation is tested on a 2D differentially heated cavity benchmark for several different Rayleigh numbers. The properties of the simulations are summarized in Table 1 and Table 2 where the Rayleigh number is controlled by varying the thermal conductivity. An ideal gas is assumed for the simulated fluid. Velocity magnitude color plots with temperature contours are shown in Figures 1, 2, and 3 for Rayleigh numbers of 10^4, 10^5,

Table 1: Summary of properties kept constant across heated cavity simulations. All units are SI.

Property	Value
Cavity dimension	10
Gravitational acceleration	1
Dynamic viscosity	4×10^{-2}
Constant pressure specific heat	1
Left wall temperature	310
Right wall temperature	300
Mean pressure	10^5
Molar mass	.029

Table 2: Thermal conductivities and resulting Rayleigh numbers for the differentially heated cavity benchmark.

Thermal Conductivity	Rayleigh #
10^{-1}	10^4
10^{-2}	10^5
10^{-3}	10^6

and 10^6, respectively. The results are in good qualitative agreement with the literature [15]. These natural convection results are obtained via a transient march to steady-state in two minutes of wall-time using four processes on a workstation with the weakly compressible implementation. It is impossible to achieve these results in any amount of time using the fully compressible implementation due to the low Mach numbers and high Rayleigh numbers. An even faster wall-time is achieved for the weakly compressible implementation if a steady executioner is used (no time derivatives).

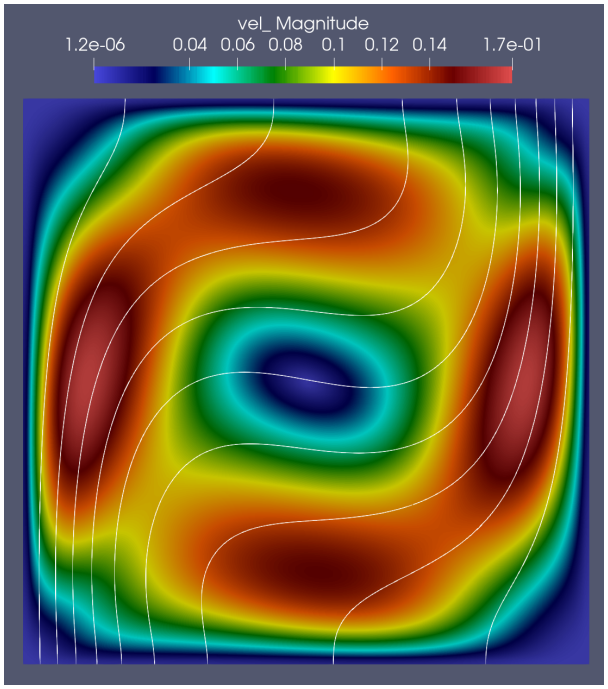

Figure 1: Natural convection benchmark result with a Rayleigh number of 10^4. Coloring is based on the velocity magnitude, and contours correspond to temperature.

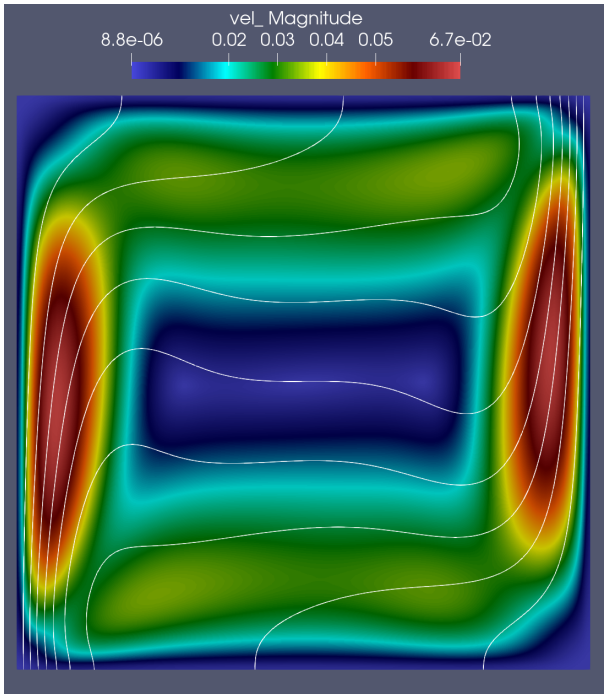

Figure 2: Natural convection benchmark result with a Rayleigh number of 10^5. Coloring is based on the velocity magnitude, and contours correspond to temperature.

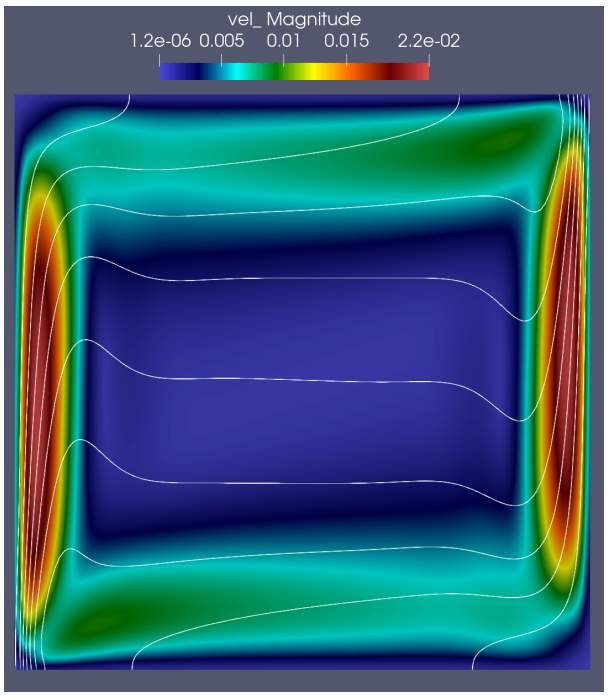

Figure 3: Natural convection benchmark result with a Rayleigh number of 10^6. Coloring is based on the velocity magnitude, and contours correspond to temperature.

3. Monotonic Interpolation at Sharp Interfaces

MOOSE uses a collocated approach to solving the Navier-Stokes equations where pressure and velocities are both defined at the cell centroids [13]. In this setting, the RCI [16] must be used to avoid the checkerboard pattern in the pressure solution that arises from decoupling the pressure in adjacent elements.

Pronghorn is designed for solving coarse-mesh thermal-hydraulics problems, where reactor cores are modeled as largely homogenized, porous zones and drag forces are used to represent the sub-scale detail of the flow in these regions. Therefore, Pronghorn models usually contain discontinuities in porosity that are accompanied by a discontinuity in the drag forces. Using the previously available interpolation scheme, neither pressure nor velocity solutions are monotonic at these interfaces. The non-monotonic solutions at the interface can contaminate large regions of the domain because Pronghorn models typically use a coarse mesh. Over- and undershoots lead to an unreliable prediction of drag and heat transfer correlations; these correlations are typically nonlinear so that an undershoot does not average out an overshoot.

There are two causes for the oscillations at porosity and body force discontinuities. These two reasons are:

- Neglecting Darcy-Forchheimer terms in the RCI
- Linear interpolation results in a poor approximation of the pseudo-momentum equations on faces in the presence of strong discontinuities of porosity and body forces.

These will both be discussed in turn in this section.

This section reports the implementation of an algorithm partially based on a modification of Moukalled's approach to strongly space-dependent body forces that largely eliminates oscillations from Pronghorn solutions. We first introduce the discretized momentum equations and RCI, then we discuss the reason for oscillations, next we describe the algorithm that was implemented in Pronghorn, and finally we present numerical results showing the monotonicity of the algorithm.

3.1 Discretized Momentum Equations

We start the discussion with the momentum conservation equations in Equation 1 by first writing it for component i:

$$\frac{\partial \rho v_i}{\partial t} + \nabla \cdot \left(\gamma^{-1} \rho \vec{v} v_i \right) = -\gamma \nabla_i p + \gamma \rho g_i - W \rho v_i. \tag{4}$$

We integrate over the extent of Cell C:

$$V_C \frac{\rho_C v_{C,i}}{\Delta t} - V_C \frac{\rho_C^o v_{C,i}^o}{\Delta t} + \frac{1}{\gamma_C} \int_{V_C} \nabla \cdot (\rho \vec{v} v_i)\, dV = -\gamma_C V_C \left. \nabla_i p \right|_C + \gamma_C V_C \rho_C g_i - V_C W_C \rho_C v_{C,i}, \tag{5}$$

where V_C is the volume of Cell C, subscript C indicates that the quantity is the cell-centroid value in Cell C, Δt is the timestep, and the superscript o indicates old timestep values. Note that we used an implicit Euler scheme for discretizing the equation, but equivalent equations can be derived for any other time discretization scheme. The γ^{-1} can be pulled out of the divergence by differentiating the term and noting that γ only changes by discrete jumps across two elements, so it is constant within each element and it follows that $\nabla \gamma^{-1} = 0$ in element C. The advection term is rewritten using Gauss' theorem:

$$\frac{1}{\gamma_C} \int_{V_C} \nabla \cdot (\rho \vec{v} v_i)\, dV = \frac{1}{\gamma_C} \sum_{f \in N(C)} v_{i,f} \vec{S}_f \cdot (\rho \vec{v})_f \tag{6}$$

where f indexes faces that separate the neighbor element N from element C and $\vec{S}_f$ is the area of face f multiplied with the outward normal vector of face f pointing from C to N. The face values are obtained by interpolation from the neighboring cell values by:

$$\phi_f \approx g_{C,f} \phi_C + (1 - g_{C,f}) \phi_N, \tag{7}$$

To indicate that we used linear interpolation, we will write:

$$\overline{\phi_f} = g_{C,f} \phi_C + (1 - g_{C,f}) \phi_N, \tag{8}$$

in the remainder of this section. Finally, the discretized momentum equation becomes:

$$\left(V_C\frac{\rho_C}{\Delta t}+\sum_{f\in N(C)}\frac{g_{C,f}\vec{S}_f\cdot(\rho\vec{v})_f}{\gamma_C}\right)v_{C,i}+\sum_{f\in N(C)}\left(\frac{(1-g_{C,f})\vec{S}_f\cdot(\rho\vec{v})_f}{\gamma_C}v_{N,i}\right)$$
$$=-\gamma_C V_C\ \nabla_i p|_C+\gamma_C V_C\rho_C g_i-V_C W_C\rho_C v_{C,i}+V_C\frac{\rho_C^o v_{C,i}^o}{\Delta t}. \tag{9}$$

We introduce the following short-hand notation for the discretized momentum equation:

$$v_{i,C}+H_C\left[v_i\right]=-D_C\ \nabla_i p|_C+b_{C,i}+D_C B_{C,i}, \tag{10}$$

where:

$$\begin{aligned}
a_C&=\left(V_C\frac{\rho_C}{\Delta t}+\sum_{f\in N(C)}\frac{g_{C,f}\vec{S}_f\cdot(\rho\vec{v})_f}{\gamma_C}\right)\\
a_f&=\left(\frac{(1-g_{C,f})\vec{S}_f\cdot(\rho\vec{v})_f}{\gamma_C}\right)\\
H_C\left[v_i\right]&=\sum_{f\in N(C)}\frac{a_f}{a_C}v_{i,N} \qquad (11)\\
b_{C,i}&=\left(\gamma_C V_C\rho_C g_i+V_C\frac{\rho_C^o v_{C,i}^o}{\Delta t}\right)/a_C\\
B_{C,i}&=-W_C\rho_C v_{C,i}\\
D_C&=\frac{\gamma_C V_C}{a_C}. \qquad (12)
\end{aligned}$$

3.2 The Rhie-Chow Interpolation

The RCI suppresses the checkerboard pattern in the pressure solution by coupling pressure in adjacent cells. The basic idea is to emulate the discretization of the momentum equation on a staggered grid where the checkerboard pattern in pressure does not occur. The traditional RCI was derived for the momentum equations without Darcy-Forchheimer terms (i.e., $B_{C,i}=0$), and for the introduction of the RCI, we will follow this approach.

Momentum equations for Cell C and the neighbor P are given by:

$$\begin{aligned} v_{i,C} + H_C\left[v_i\right] &= -D_C\ \nabla_i p|_C + b_{C,i} \\ v_{i,P} + H_P\left[v_i\right] &= -D_C\ \nabla_i p|_P + b_{P,i}. \end{aligned} \tag{13}$$

In addition, we propose that a similar equation describes the velocity on the face f between C and P:

$$v_{i,f} + H_f\left[v_i\right] = -D_f\ \nabla_i p|_f + b_{f,i}. \tag{14}$$

The coefficients in this equation are unknown, but we approximate them by linearly interpolating from the values in Cell C and P:

$$H_f\left[v_i\right] = g_{C,i} H_C\left[v_i\right] + (1 - g_{c,i}) H_P\left[v_i\right] = \overline{H_f\left[v_i\right]}. \tag{15}$$

and similarly for D_f and $b_{f,i}$. We obtain the pseudo momentum equation at the face f:

$$v_{i,f} + \overline{H_f\left[v_i\right]} = -\overline{D_f}\ \nabla_i p|_f + \overline{b_{f,i}}. \tag{16}$$

The next step in the derivation is to replace $\overline{H_f\left[v_i\right]}$ with an expression derived from Equation 13. We solve for $H_C\left[v_i\right]$ and $H_P\left[v_i\right]$ and then compute:

$$\overline{H_f\left[v_i\right]} = g_{C,i} H_C\left[v_i\right] + (1 - g_{C,i}) H_P\left[v_i\right] \approx -\overline{v_{f,i}} - \overline{D_f}\,\overline{\nabla_i p|_f} + \overline{b_{f,i}} + \mathcal{O}(h^2). \tag{17}$$

Substituting Equation 17 into Equation 16 and rearranging gives:

$$v_{f,i} = \overline{v_{f,i}} - \overline{D_f}\left(\nabla_i p|_f - \overline{\nabla_i p|_f}\right). \tag{18}$$

The important difference between ϕ_f and $\overline{\phi}_f$ (where ϕ serves as a placeholder) is that $\overline{\phi_f}$ is simply obtained from the adjacent cell values by linear interpolation, while ϕ_f is the value ϕ assumes on the face. For example, for $\nabla_i p$, $\nabla_i p|_f$ is the differencing formula applied to the pressure on the face, while $\overline{\nabla_i p|_f}$ is the linear interpolation of the pressure gradient in direction i stored at the cell centers of Cells C and P.

3.3 Including Body Force Terms in the Rhie-Chow Interpolation

We follow Reference [13] in the approach to include body-force terms in the RCI. The discretized momentum equation in Cell C that we use as a starting point of the discussion is:

$$v_{i,C} + H_C\left[v_i\right] = -D_C\ \nabla_i p|_C + b_{C,i} + D_C\overline{\overline{B_{C,i}}}, \tag{19}$$

where $\overline{\overline{B_{C,i}}}$ replaces $B_{C,i}$. We define $\overline{\overline{\phi_C}}$ as:

$$\overline{\overline{\phi_C}} = \frac{1}{V_C}\sum_{f\in N(C)}\overline{\phi_f}. \tag{20}$$

Using the definition of $\overline{\overline{\phi_C}}$ and $\overline{\phi_f}$, we can recursively define higher order averaged quantities by consecutively interpolating to all faces and back to the centroids. Replacing $B_{C,i}$ with $\overline{\overline{B_{C,i}}}$ is necessary on collocated grids to ensure that the stencil of the body-forces is similar to that of the pressure gradient [13]. Repeating the RCI derivation, including $\overline{\overline{B_{C,i}}}$, gives:

$$v_{f,i} = \overline{v_{f,i}} - \overline{D_f}\left(\nabla_i p|_f - \overline{\nabla_i p|_f}\right) + \overline{D_f}\left(\overline{B_{f,i}} - \overline{\overline{B_{f,i}}}\right). \tag{21}$$

The MOOSE Navier-Stokes module implementation of the modified RCI allows the repeated application of the smoothing operator to the body forces. However, no large differences in results were found after a single application of the smoothing operator.

3.4 Monotonic Solutions in Staggered Meshes

The oscillations at porous interfaces are produced by inconsistent pressure interpolation at the sharp porous media interfaces. This problem does not appear in staggered grids since pressure is directly solved at the interfaces of velocity cells. A representation of the staggered grid discretization is presented in Figure 4. The shaded blue region corresponds to a low porosity porous media, whereas the white region corresponds to a high porosity one. As observed, there is no artificial aliasing in the pressure field since the pressure at the sharp interface between porous media is resolved. However, this is not the case for collocated grids, as explained in the next section.

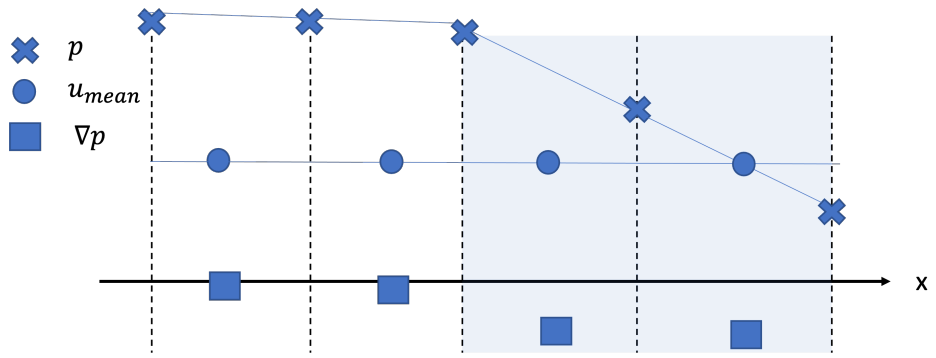

Figure 4: Example of 1D porous media in a staggered mesh. No numerical oscillations are produced when the pressure variables are aligned with the porous interfaces.

3.5 Non-Monotonic Solutions in Collocated Meshes

The process by which numerical oscillations are produced in the velocity and pressure fields for a collocated mesh is presented in Figure 5. The first figure depicts the pressure, velocity, and pressure gradients at the cell centers as they would be computed in the collocated mesh. For reference, the previous interface pressure resolved by staggered grid is kept as a hollow cross symbol for reference. When solving the linear momentum conservation in the Navier Stokes equation, Equation (1), pressure is interpolated to the cell interfaces. The interpolated pressure is presented in the middle panel of Figure 5 with red crosses. As observed, the interpolation yields a pressure at the porous media interface that is lower than the actual pressure resolved by the staggered mesh. Therefore, the pressure gradient in the cell upstream of the porous media interface is larger in magnitude than the actual gradient. Contrarily, the magnitude of the pressure gradient for the cell downstream of the porous interface is smaller than the actual pressure gradient. The computed pressure gradient is indicated with the green squares in the middle panel of Figure 5. This effect results in a net flow acceleration before the porous interface and a net deceleration after the porous interface, which yields the observed oscillations in the superficial velocity field and, consequently, on the pressure field. This is represented by the green circles in the bottom panel of Figure 5.

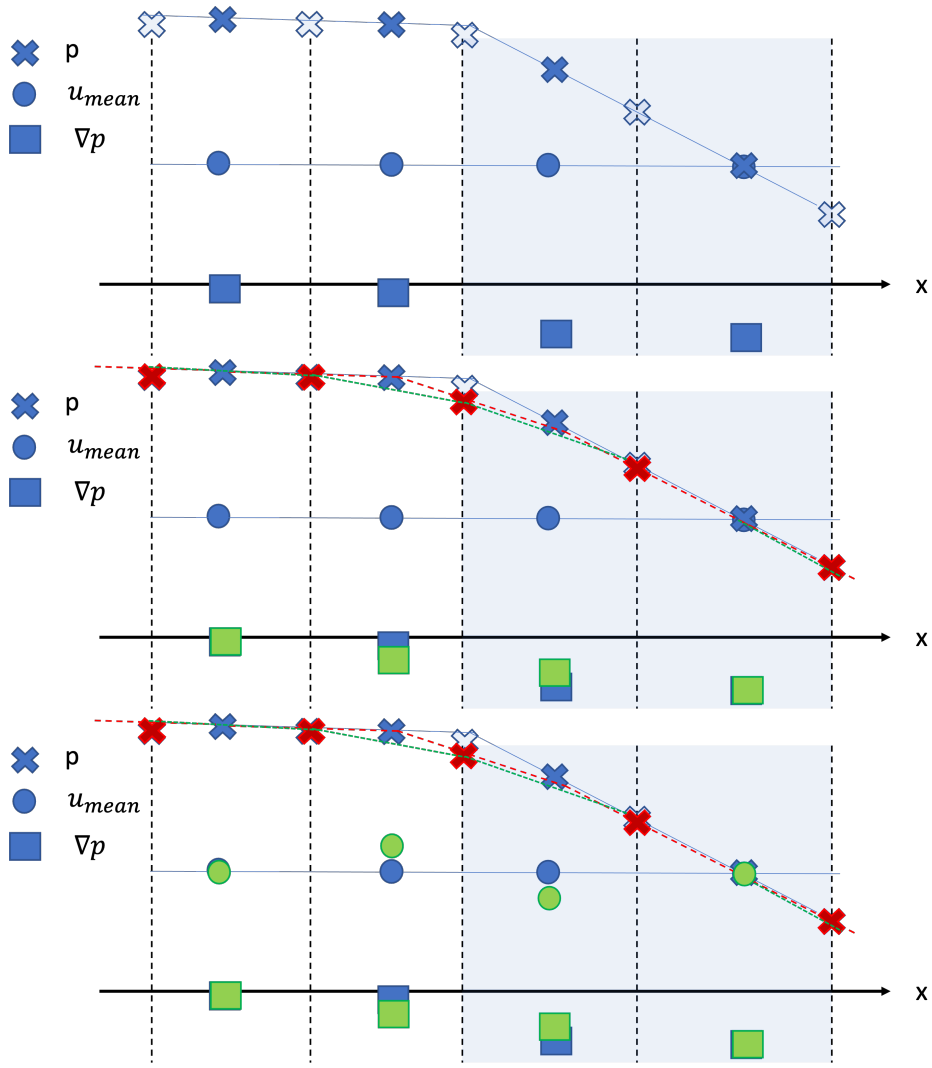

Figure 5: Example of 1D porous media in a collocated grid. *Top*: pressure, velocity, and pressure gradient in the collocated mesh. *Middle*: interpolated pressure at the cell interfaces (red) during the discretization of the Navier Stokes equation. *Bottom*: artificial oscillations in the velocity field (green) due to inconsistent pressure gradients at the cell centers.

If the mesh resolution is fine enough next to the interface, the pressure interpolation error reduces, and only small oscillations are produced, which are damped by physical and numerical viscosity in the vicinity of the interface. However, for coarser mesh resolutions, oscillations may be present far away from the interface due to the non-linear advection term in the linear momentum conservation equation. Large oscillations can ultimately compromise numerical convergence and lead to inaccurate results. In the next section, we propose an approach to reduce the numerical oscillations at the porous media interfaces.

3.6 Consistent Interpolation to Reduce Oscillations in Collocated Meshes

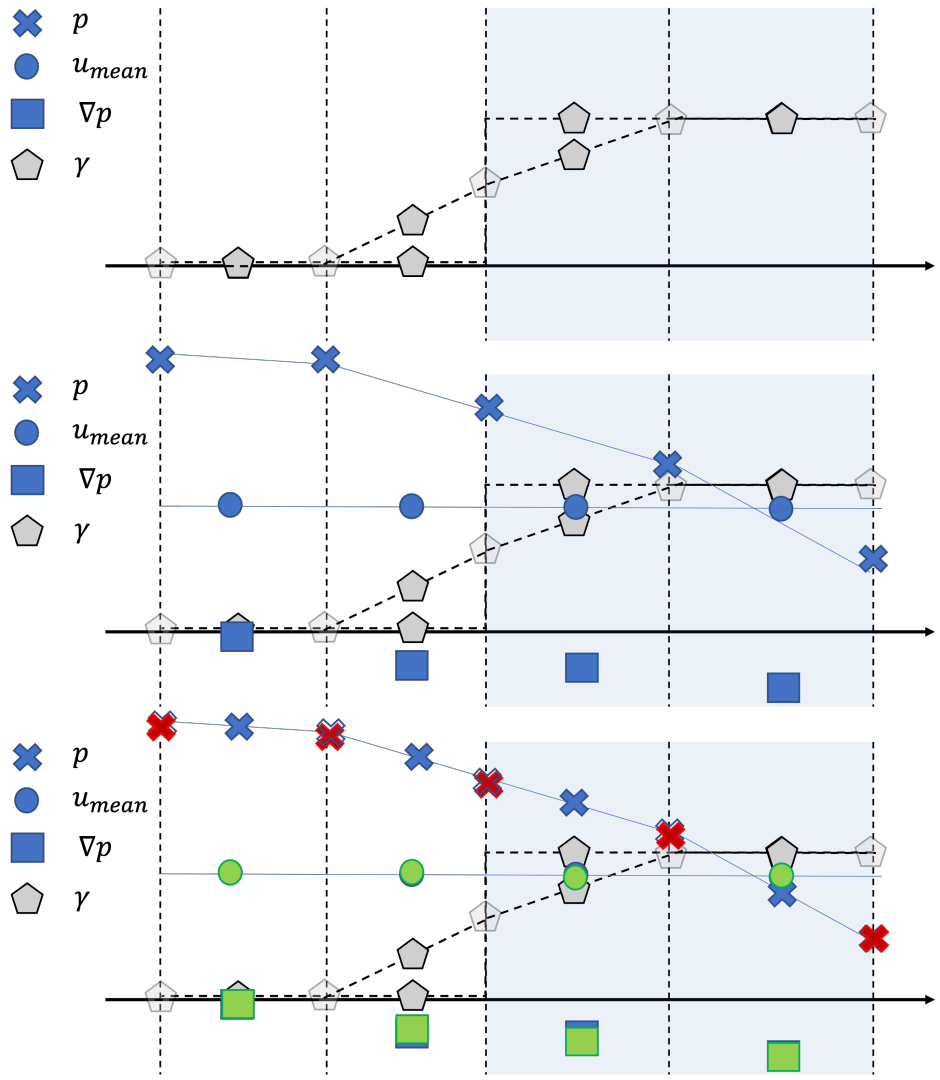

Figure 6: Porous media interpolation to reduce oscillations at interfaces. *Top*: consistent porous media interpolation. *Middle*: modified pressure field with interpolated porous media. *Bottom*: modified pressure gradients and velocities with the consistent interpolation approach.

The schematic description of our consistent interpolation scheme is presented in Figure 6. As depicted in the first panel of this figure, the key step of this approach involves a consistent interpolation of porosity γ_C similar to the body force interpolation scheme in Equation 20. The consistent interpolation scheme linearly interpolates between cell centers and faces and then takes the surface weighted average of the face interpolated values to reconstruct a new cell-centered

porosity as follows:

$$\gamma_{C,1} = \frac{1}{\sum\limits_{f \in N(C)} |\vec{S}_f|} \sum_{f \in N(C)} \overline{\gamma_{C,0}}_f |\vec{S}_f|. \tag{22}$$

Note that the new reconstructed porosity reduces the interpolation error of pressure at the interfaces, as depicted in the bottom panel of Figure 6. However, there is still an error associated with the interpolation through the porosity jump. To progressively reduce this error, we recursively implement the interpolation-reconstruction operation in the porosity field, as defined in Equation 22, where the k interpolation-reconstruction operation can be defined in terms of the $k-1$ operation as follows:

$$\gamma_{C,k} = \frac{1}{\sum\limits_{f \in N(C)} |\vec{S}_f|} \sum_{f \in N(C)} \overline{\gamma_{C,k-1}}_f |\vec{S}_f|. \tag{23}$$

As the number k increases in the interpolation-reconstruction operations, we reduce the pressure interpolation error at the interfaces and, thus, the numerical oscillations in the velocity field. However, high-fidelity resolution at the interfaces is lost as more interpolation-reconstruction operations are applied to porosity. In practice, one should tune the number of interpolation-reconstruction operations to be large enough for velocity oscillations to be reduced to a point at which they can be damped by numerical and physical viscosity. However, it is not recommended to use a very large number of porosity interpolation-reconstruction operations as it will severely reduce the resolution at interfaces.

3.7 Numerical Results

In this section, we provide numerical results demonstrating the effectiveness of the implemented monotonic interpolation method. First, we present results for a three-region pipe with a porous Darcy/Forchheimer region between two free-flow regions, and then we present multidimensional results for a geometry loosely inspired by salt-cooled pebble-bed reactors.

3.7.1 Three-Region Pipe

The first test problem is a three-region pipe inspired by Reference [16]. The problem is depicted in Figure 7. It is essentially a one-dimensional problem modeled in a two-dimensional geometry. The pipe is 15 meters long and 1 meter wide. It is split into three segments of equal length. The first and last segment are free-flow regions with a porosity of 1, while the middle region is a porous flow region with a porosity of 0.4. The Ergun drag coefficient is used in the middle region:

$$-\gamma \frac{\Delta p}{\Delta L} = \frac{150\mu}{D^2}\frac{(1-\gamma)^2}{\gamma^2} v_x + \frac{1.75\rho}{D}\frac{1-\gamma}{\gamma^2}|v_x|v_x, \tag{24}$$

where we set the pebble diameter to $D = 0.03$ m, the dynamic viscosity to 1×10^{-3} Pa-s, and $\rho = 2000$ kg/s. The boundary conditions of the problem are an inflow velocity of $v_x = 0.5$ m/s on the left and an outlet pressure of 0 Pa on the right; the top and bottom boundaries are free-slip boundaries. Incompressible Navier-Stokes equations without heat equation are solved, and no viscous terms are included except the Darcy drag force.

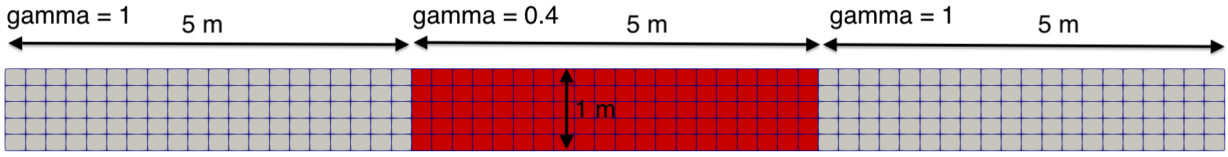

Figure 7: Geometry and mesh of the three-region pipe problem.

The superficial velocity component v_x is plotted along the center line of the pipe in Figure 8 for the monotonic and old non-monotonic interpolation schemes. The exact solution for the x-component of the superficial velocity problem is $u_x = 0.5$ everywhere. The monotonic interpolation method predicts the solution almost exactly (small fluctuations at the porosity are present but not clearly visible in Figure 8) while the non-monotonic interpolation results show large over- and underswings of u_x on the order of 0.2 m/s.

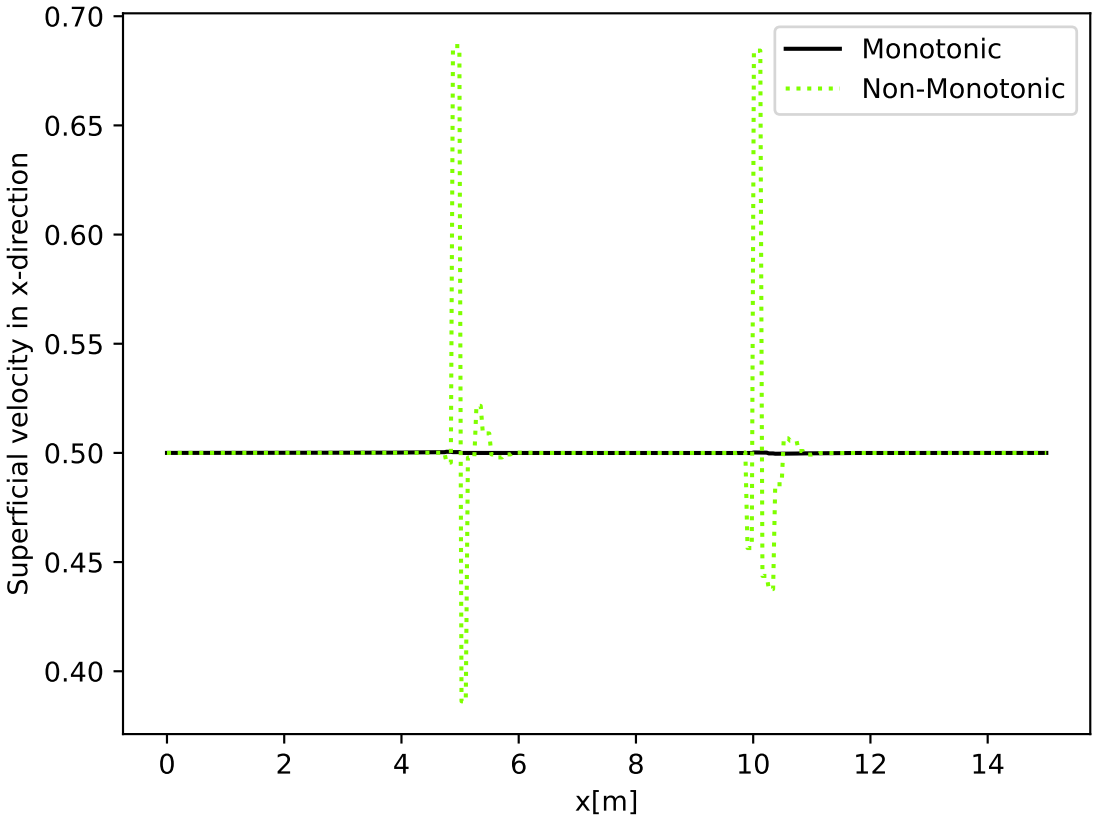

Figure 8: Superficial x-component of the velocity plotted against x along the centerline of the pipe monotonic and old non-monotonic interpolation scheme.

3.7.2 Fluoride-Salt-Cooled High-Temperature Reactor Mockup

We considered two different, two-dimensional geometries shown in Figure 9. These two geometries are loosely inspired by Fluoride-Salt-Cooled High-Temperature Reactor (FHR) reactors but are in no way intended to serve as the basis of any FHR analysis whatsoever (in fact, they lack several important features of FHRs). The mesh used for discretizing the geometries is depicted, and it exemplifies what is meant by coarse-mesh thermal-hydraulics. A total of about $1,000$ elements are used, boundary layers are not resolved, and free slip-boundary conditions are used on all walls. The difference between the two configuration is that the first configuration has only one outlet boundary at the top, while the second configuration on the right of Figure 9, has an additional chimney-like flow outlet.

Table 3: Parameters used for the two-dimensional FHR-inspired test problem.

Quantity	Value
Density	$2,000\ \mathrm{kg/m^3}$
Viscosity	3×10^{-2} Pa-s
Pebble diameter	0.03 m
Inlet speed	1 m/s
Drag force correlation (except chimney)	Ergun
Drag force correlation (except chimney)	constant Darcy 0.2
Porosity bed (grey region in Figure 9)	0.4
Porosity chimney region (yellow region Figure 9)	0.9
Porosity everywhere else	0.6

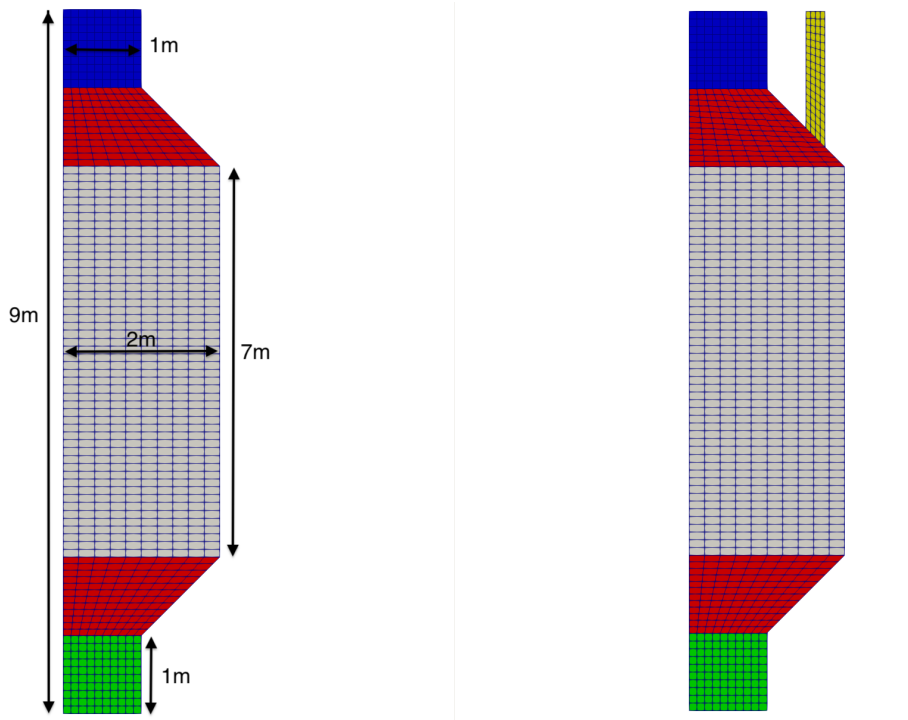

Figure 9: Conceptual two-dimensional test problem loosely inspired by FHR reactors. The bed region is grey.

We use the parameters provided in Table 3 in this test problem. We solve the incompressible mass and momentum equations, while the temperature equation is not solved. Viscous effects, with the exception of the Darcy drag force, are not considered in the equations.

We show a two-dimensional plot of the vertical velocity v_y for Variants A (without chimney) and B (with chimney) in Figure 10; for both Variants A and B, results are shown using the old interpolation method and the new monotonic interpolation method. In addition, a line-out plot

of velocity v_y along the line $x = 0.5$m is shown in Figure 11. Without the improved interpolation and smoothing operations described in this section, strong oscillations in v_y are present at the bed interface. These oscillations mostly disappear for both variants when the monotonic interpolation scheme is used. There appears to be a very small residual under- and over-swing at the upper end of the bed for both variants when using the monotonic interpolation. In addition, the non-monotonic interpolation leads to a significantly different velocity distribution than the monotonic interpolation. The non-monotonic solution for both variants appears to be less physically realistic. In the monotonic interpolation cases, fluid slowly distributes into the additional flow area provided by the bed and then redistributes back into the upper end of the bed. This leads to a roughly parabolic shape of the vertical velocity in the bed, while it is almost flat for the non-monotonic interpolation. In summary, the implemented improvement in interpolation methods reduce oscillations at porosity and body force discontinuities significantly.

The execution time for Variant A on a single processor is about 90 seconds, while it is much longer for the original non-monotonic interpolation scheme (the original interpolation scheme struggles with convergence and is unable to significantly increase the time step to get to the steady-state solution). The application of repeated smoothing, as shown in Equation 23, has a negligible effect on the execution time. The execution time can be reduced on a small workstation to 25–30 seconds when using 12 processors. The resolution/execution time profile of this test case is characteristic for the intended use of Pronghorn as a coarse-mesh thermal-hydraulics code. We expect future improvements in performance when the segregated solver algorithms are fully deployed.

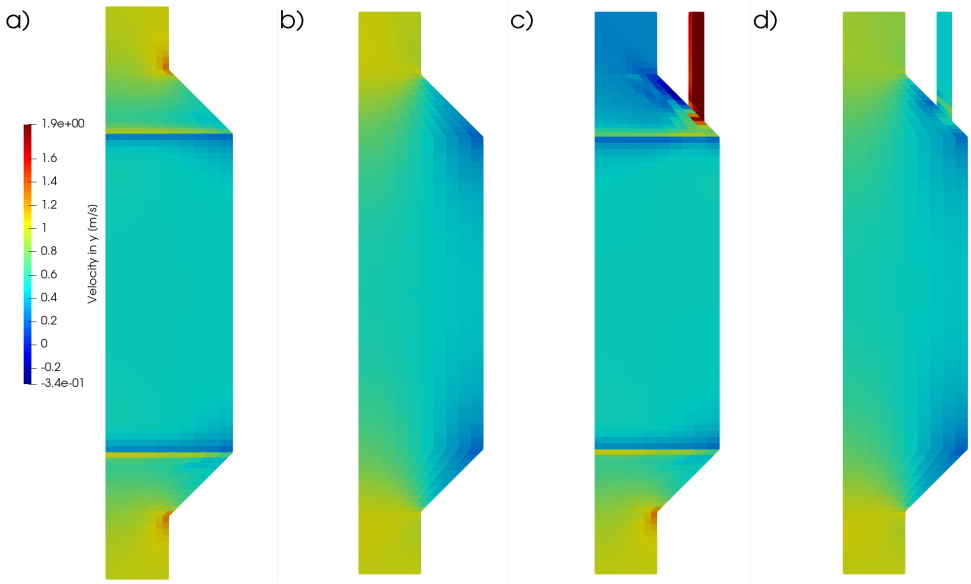

Figure 10: Vertical velocity v_y for a) Variant A without an application of monotonic interpolation, b) Variant A with an application of the monotonic interpolation, c) Variant B without an application of the monotonic interpolation, and d) Variant B with an application of the monotonic interpolation.

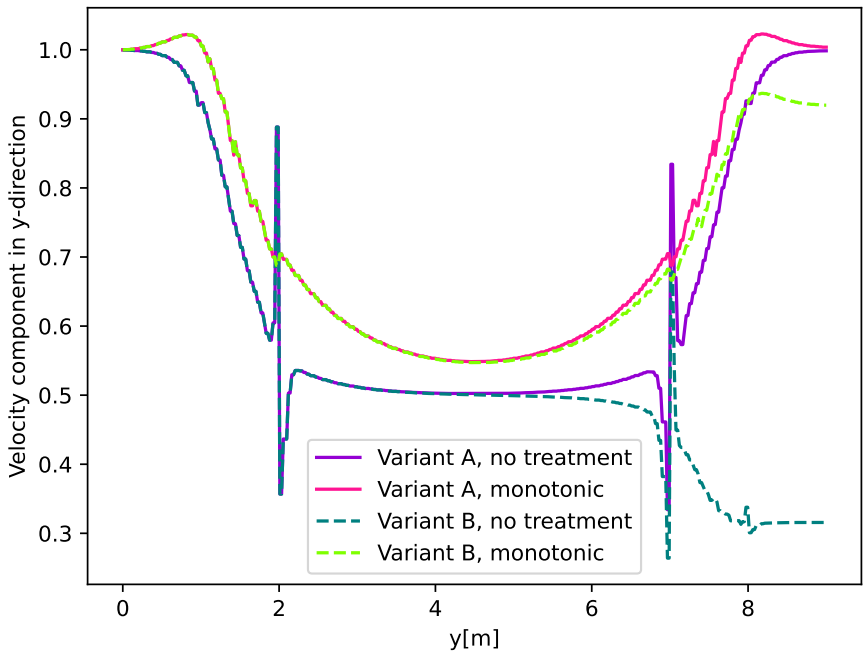

Figure 11: Line-out of vertical velocity v_y along the line $x = 0.5$m for Variants A and B simulated without the treatment described in this section (no treatment) and with the treatment applied in this section (monotonic).

4. Implementation of Skewness Correction in MOOSE

In the FVM, the linear interpolation of face values can be inaccurate on unstructured meshes. These face values are used in the discretized forms of the advection, diffusion, and gradient terms in the Navier-Stokes equations, so an inaccurate face value leads to inaccurate solutions. Figure 12 demonstrates the underlying issue using a cell-pair from a skewed two-dimensional triangulation.

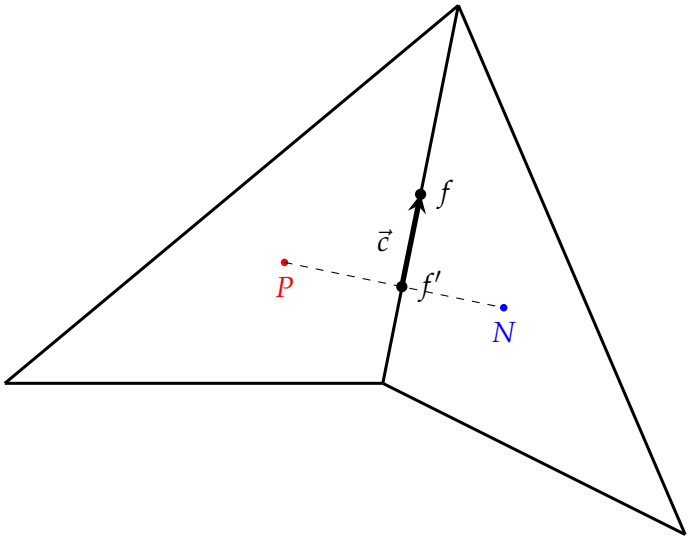

Figure 12: A typical skewed cell-pair in an unstructured mesh (P - cell centroid, N - centroid of neighboring cell, f - face centroid, f' - intersection of face and the line connecting cell centroids, and $\vec{c}$ - correction vector).

As an example, let us take the integral of the gradient of an arbitrary field Φ. This can be discretized in a finite volume setting using the Green-Gauss theorem as follows:

$$\int_{V_P} \nabla \Phi dV \approx \sum_{i}^{N_f} \Phi_{f_i} \vec{S}_{f_i}, \tag{25}$$

where $\Phi_{f_i} = \Phi(f_i)$ is the value of the variable at the center of face $i = 1, ..., N_f$ of Cell P. In the most basic discretization approach, the face value is approximated using the values at the centers of adjacent cells:

$$\Phi_{f_i} \approx g_{c,i} \Phi_P + (1 - g_{c,i}) \Phi_{N,i}, \tag{26}$$

where $g_{c,i}$ is usually computed as $\frac{||\vec{r}_{N_i f_i'}||}{||\vec{r}_{N_i P}||}$ with $\vec{r}_{N_i f'}$ and $\vec{r}_{N_i P}$ denoting vectors connecting N_i with f_i' and P, respectively. If the interpolation weights are computed using the above mentioned

procedure, the value of the variable at the true face centroid f is approximated by the value at f' (i.e., where the line connecting the two cell centers intersects the face). This approach yields a high-order numerical scheme for structured grids where f and f' coincide. However, on unstructured grids, where f' and f are different (see Figure 12), utilizing Equation (26) alone results in loss of accuracy. In this paper, we use the following indicator to quantify the skewness of cell-pairs:

$$\mathbf{s_i} = \frac{||\vec{c}_i||}{||\vec{r}_{N_iP}||} \times 100\%, \tag{27}$$

where $\vec{c}_i$ is a correction vector connecting f and f'.

In order to increase the accuracy of MOOSE's finite-volume discretization on skewed meshes, a correction for computing face values from cell values is implemented. This correction utilizes the gradient of Φ at f' as follows:

$$\Phi_{f_i} \approx g_{c,i}\Phi_P + (1 - g_{c,i})\Phi_{N,i} + \nabla\Phi_{f_i'} \cdot \vec{c}_i = \Phi_{f_i'} + \nabla\Phi_{f_i'} \cdot \vec{c}_i. \tag{28}$$

This correction greatly increases the convergence rate of the numerical scheme on skewed meshes. However, this comes at the cost of computing $\nabla\Phi_{f_i'}$ yielding an increase in solve times due to an increase in stencil size. In the Navier-Stokes equations, this correction can be used for the determination of the face velocities in the advection term, the skewness correction of the non-orthogonal correction of the face gradients, and the computation of the pressure gradient at cell centers.

4.1 Verification of Skewness Correction using Method of Manufactured Solutions

The skewness correction is tested using a body-force-driven cavity problem. We follow Reference [17] and use the Method of Manufactured Solution (MMS) to perform a convergence study of the velocities and pressure computed by MOOSE. The cavity problem has a size of 1×1 m (the domain is $\Omega = [0,1] \times [0,1]$), the density and viscosity are set to $\rho = 1$ kg/m^3 and

$\mu = 1$ Pa-s. The solution is selected to be:

$$
\begin{aligned}
u &= x^2(1-x)^2(2y - 6y^2 + 4y^3) \\
u &= -y(1-y)^2(2x - 6x^2 + 4x^3) \\
p &= x(1-x), \qquad (29)
\end{aligned}
$$

where u and v are the velocity components in the x and y direction and p denotes pressure. The forcing function that generates this solution can be found in Reference [17]. The boundary conditions for this MMS problem are no-slip walls ($u = v = 0$) on every side of the domain.

The velocity solution is depicted in Figure 13a), and the coarsest and finest mesh are depicted in Figure 13b) and c), respectively. The coarsest mesh is an unstructured triangular mesh. A single refinement step divides each triangle into four smaller triangles; a total of five refinement steps are performed.

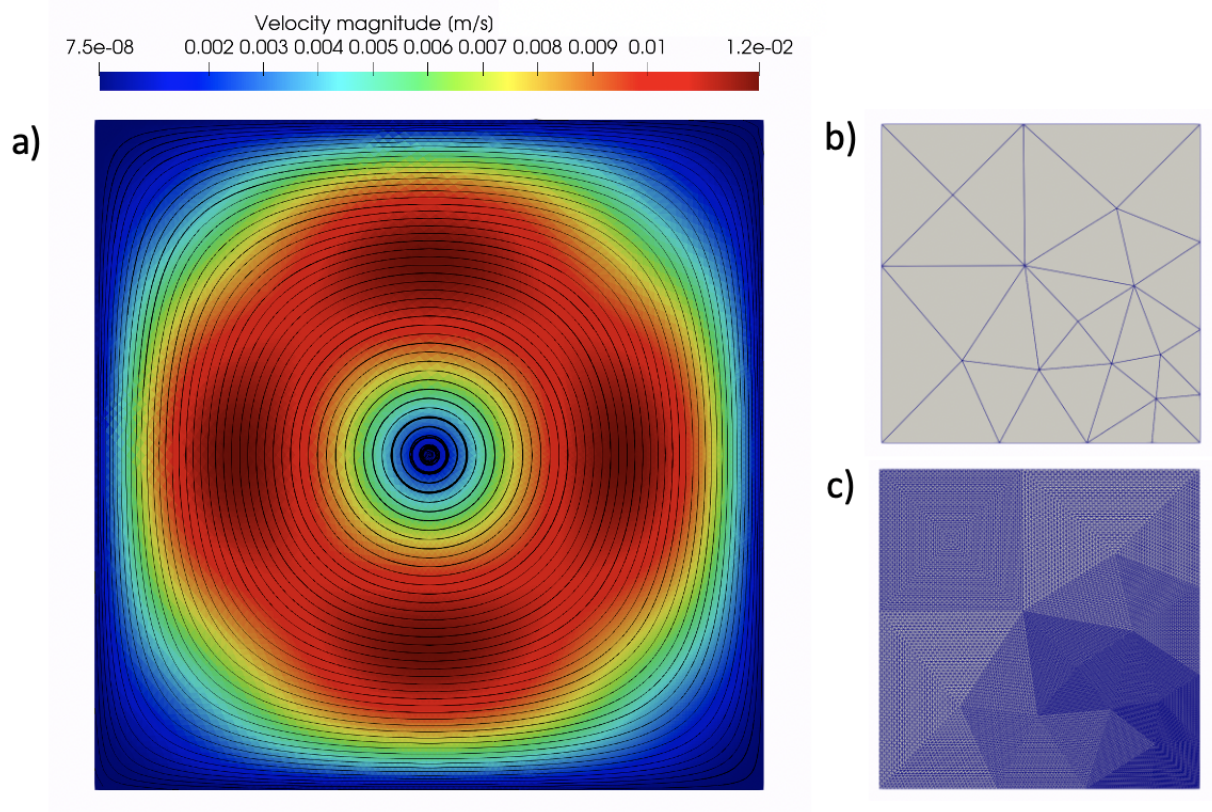

Figure 13: a) Velocity solution of the body-force-driven cavity problem, b) coarsest mesh used for convergence study, and c) finest mesh used for convergence study.

To showcase the quality of the mesh, a skewness histogram (using Equation (27) as a skewness indicator) is presented in Figure 14. Even though the majority of the cell-pairs are only slightly skewed ($\mathbf{s} < 5\%$), there are severely skewed instances where the length of the correction vector is almost 40% of the distance between the cell centers.

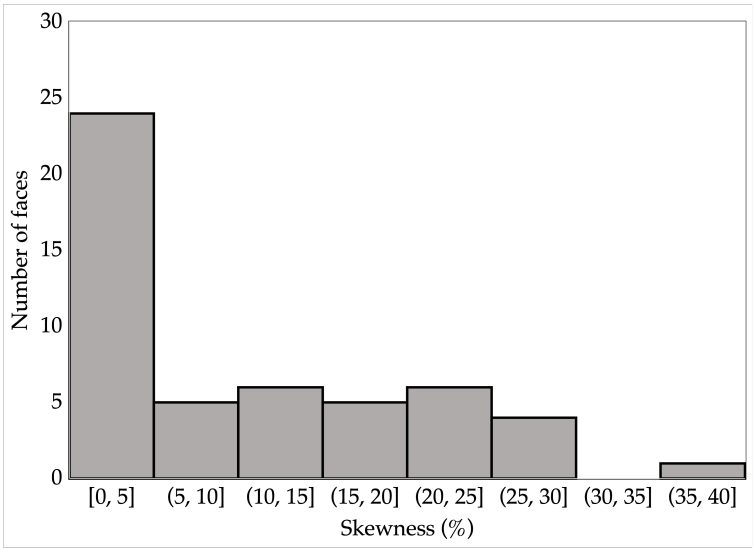

Figure 14: The skewness-distribution of the cell-pairs (faces) in the coarsest mesh.

We first solve the MMS cavity problem on the sequence of meshes without skewness correction. The error measured in the L^2 norm of the velocity components and the pressure are plotted versus the element size in Figure 15. We see that the error between the analytic and numerical solutions decreases like powers with orders of 1 and 0.5 in the case of velocity and pressure, respectively. This is considered to be a poor convergence rate compared to the second-order convergence of orthogonal non-skewed structural meshes.

Next, we solve the MMS cavity problem with skewness correction and plot the resulting error versus the element size in Figure 16. Using the skewness correction, the order of convergence increases to 2 for the velocity components and 1 for the pressure, which is a considerable improvement over the case without skewness correction.

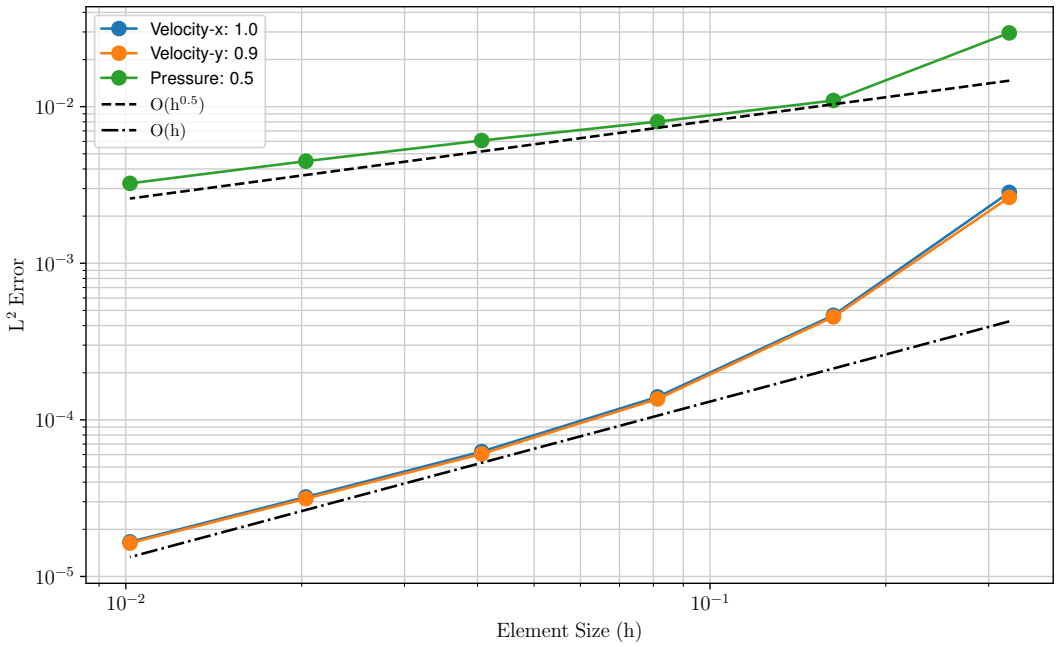

Figure 15: Convergence of velocity and pressure for the MMS test case without using skewness correction.

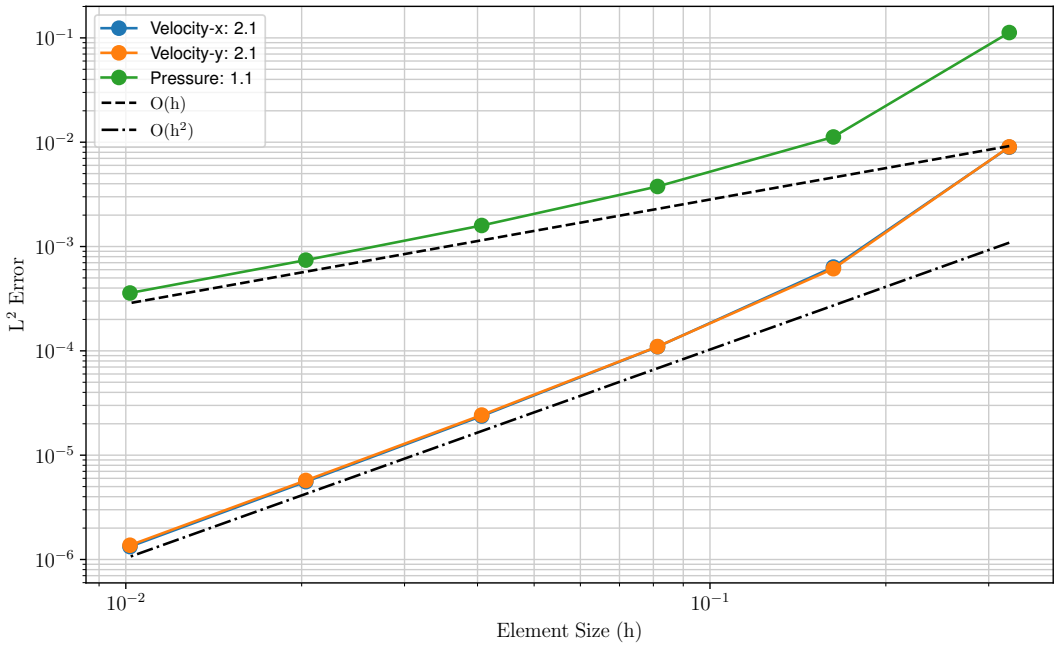

Figure 16: Convergence of velocity and pressure for the MMS test case when using skewness correction.

5. Development of a SIMPLE Segregated Solver

Traditionally, MOOSE has used a monolithic approach to solving the nonlinear problem arising from the Navier-Stokes equations at each timestep. MOOSE is designed to solve multiphysics problems in a fully coupled approach, making it the solver algorithm of choice for the initial development of thermal-hydraulics capabilities. However, the standard solver approach for incompressible and mildly compressible fluid problems is a segregated solver. These solvers separate the momentum and pressure solves and iterate between them in a fixed-point fashion. Segregated solvers offer the following advantages over monolithic solvers:

- For incompressible problems, the system of mass and momentum equations is a saddle-point problem, making its direct solution difficult. In general, the monolithic system of mass and momentum equations is hard to solve using Newton's method or a quasi-Newton method, because good linear solvers for the full Jacobian method (or an approximation thereof) are not readily available. Lower-upper factorizations work for two-dimensional problems that have a sufficiently small number of degrees of freedom.
- RCI was developed for segregated solvers and must be specifically adapted for monolithic solvers.
- Each portion of the segregated solve (momentum and then pressure solve) are well defined linear problems that are relatively easy to solve.
- A large body of knowledge exists for segregated solvers, including under-relaxation, the inclusion of higher order differencing methods, RCI, etc.

Therefore, segregated solver algorithms are currently developed in the MOOSE Navier-Stokes module. We report the implementation and initial testing of the SIMPLE algorithm [13] in this section.

We first introduce the SIMPLE algorithm that is implemented in MOOSE. The algorithm itself is a variation of the algorithm described in Reference [13] in how the body force terms are treated. We use the general notation introduced in Section 3.

For each dimension $i = x, y, z$, the momentum equations in Equation 9 are linearized by evaluating the coefficients of $v_{C,i}$ and $v_{N,i}$ and the drag coefficient W_C at the last iterate of the

velocity. In addition, the pressure term for the current iteration is expressed as $p^{(n)} = p^{(n-1)} + p'$, where $p^{(n-1)}$ is the pressure lagged by one iteration. We solve the momentum predictor equation using $p^{(n-1)}$ and enforce incompressibility via p'. We collect all terms multiplying velocities into the matrix $\mathbf{M}$ with the exception of the Darcy and Forchheimer drag terms, which are collected into the matrix $\mathbf{F}$. Dividing by γV_C, we obtain:

$$\left(\mathbf{M}_i + \mathbf{F}_i\right) v_i^* = -\nabla_i p^{(n-1)} + f_i + \frac{\rho_C^o v_i}{\gamma \Delta t} \tag{30}$$

where we denote iteration indices with a superscript (n), v_i^* and $\nabla_i p^{(n-1)}$ are column vectors comprising one entry for each element, and f_i collects anything that does not depend on velocity, except for the terms that are explicitly shown (including boundary conditions). In the presence of discontinuities in γ and the Darcy-Forchheimer forces, $\mathbf{F}$ can lead to oscillations. Therefore, we smooth $\mathbf{F}$ by repeated application of the $\overline{\overline{\mathbf{F}}}$ operator and denote the modified force term by $\hat{\mathbf{F}}$:

$$\left(\mathbf{M}_i + \hat{\mathbf{F}}_i\right) v_i^* = -\nabla_i p^{(n-1)} + f_i + \frac{\rho_C^o v_i}{\gamma \Delta t} \tag{31}$$

For convenience, we define $\mathbf{K}_i = \mathbf{M}_i + \hat{\mathbf{F}}_i$. The momentum predictor is usually under-relaxed by factor λ_v to ensure stability:

$$\left(\mathbf{K}_i + \mathrm{diag}(\mathbf{K}_i)\frac{1-\lambda_v}{\lambda_v}\right) v_i^* = -\nabla_i p^{(n-1)} + \mathrm{diag}(\mathbf{K}_i)\frac{1-\lambda_v}{\lambda_v} v_i^{*,(n-1)} + f_i + \frac{\rho_C^o v_i}{\gamma \Delta t} \tag{32}$$

After solving the momentum predictor equations, we split the matrix $\mathbf{M}$ into a diagonal portion and an off-diagonal portion:

$$\begin{aligned} \mathbf{A}_i &= \mathrm{diag}(\mathbf{M}_i) \\ \mathbf{H}_i &= \mathbf{M}_i - \mathrm{diag}(\mathbf{M}_i). \end{aligned} \tag{33}$$

Then we compute $\hat{H}_i$ as:

$$\hat{H}_i = -\mathbf{H}_i v_i^* - \nabla_i p^{(n-1)} + f_i + \mathrm{diag}(\mathbf{K}_i)\frac{1-\lambda_v}{\lambda_v} v_i^{*,(n-1)} + \frac{\rho_C^o v_i}{\gamma \Delta t}. \tag{34}$$

Note that, in the actual momentum predictor equation (i.e., not lagging the pressure gradient), we would have:

$$\left(\mathbf{A}_i + \hat{\mathbf{F}}_i\right) v_i^* = \hat{H}_i - \nabla_i p', \tag{35}$$

which is the proper splitting of the momentum predictor. And at convergence, we have:

$$v_i = \left(1 + \mathbf{A}_i^{-1}\hat{\mathbf{F}}_i\right)^{-1} \mathbf{A}_i^{-1}\hat{H}_i - \left(1 + \mathbf{A}_i^{-1}\hat{\mathbf{F}}_i\right)^{-1} \mathbf{A}_i^{-1}\nabla_i p', \tag{36}$$

where we used the identity $\mathbf{A}_i^{-1}\left(1 + \mathbf{A}_i^{-1}\hat{\mathbf{F}}_i\right)^{-1} = \left(\mathbf{A}_i + \hat{\mathbf{F}}_i\right)^{-1}$. We define the matrix $\mathbf{T}_i = 1 + \mathbf{A}_i^{-1}\hat{\mathbf{F}}_i$. We can now compute p' by requiring the divergence of the converged velocity field to be zero by the discrete divergence operator $\mathbf{D}$[1]. This yields an equation for the pressure predictor as follows:

$$\mathbf{D}\left(\mathbf{T}_i\right)^{-1}\mathbf{A}_i^{-1}\nabla_i p' = \mathbf{D}\left(\mathbf{T}_i\right)^{-1}\mathbf{A}_i^{-1}\hat{H}_i. \tag{37}$$

Finally, during the correction steps, the pressure field is then updated by:

$$p^{(n)} = p^{(n-1)} + \gamma_p p', \tag{38}$$

where γ_p is a relaxation factor for the porosity field, and the velocity field by:

$$v_i^{(n)} = \mathbf{T}_i^{-1}\mathbf{A}_i^{-1}\left(\hat{H}_i + \nabla_i p'\right). \tag{39}$$

The implementation of the SIMPLE algorithm implemented in MOOSE is summarized in Algorithm 1.

5.1 Implementation of the Segregated Solver in MOOSE

The SIMPLE algorithm is currently implemented using MOOSE's *MultiApp* framework. The details of the *MultiApp* system are detailed in Reference [18]. In summary, the *MultiApp* system allows a primary application to create a secondary application underneath it and exchange information with it via *MultiAppTransfers*. MOOSE supports fixed-point iterations between the primary and secondary applications. The *MultiApp* system can be used to split the system of

[1]This is exactly true only for incompressible equations but works for mildly compressible equations as well

Algorithm 1 SIMPLE Algorithm. Note the difference in velocity and advecting velocity

Use initial guess or last timestep values to initialize variables.
Iteration count $n = 0$.
while not converged **do**
  $n \leftarrow n + 1$
  # Begin momentum predictor
  for i = x,y,z **do**
    Assemble the momentum predictor and solve Equation 32 for v_i^*.
    Compute and store $\mathbf{H}v_i^*$, $\mathbf{A}$, $\mathbf{T}_i$
  end for
  Solve the pressure predictor Equation 37 for p'.
  Update the pressure using Equation 38.
  Correct the velocity using Equation 39.
end while
Move to next timestep.

equations into the momentum components and pressure and solve each of them separately.

The following split is performed:

- The primary application has the pressure correction p' as the variable to solve for.
- The momentum predictor components are solved as secondary applications. These secondary applications are executed within the fixed point iteration before the solution of the primary application for p'.
- The corrector for pressure and velocities are *AuxiliaryKernels* that are executed after the completion of the primary application's solve for p'.

Currently, the optimal under-relaxation of the momentum predictor and under-relaxation of the velocity corrector in steady and transient modes are supported in this implementation.

5.2 Backward-Facing Step

The geometry for this test case is depicted in Figure 17. It consists of a backward-facing step with an expansion ratio of 1:4. An incompressible flow develops in the inlet pipe of the backward-facing step until a sudden expansion where the tripping of the inferior boundary layer is produced. This yields a recirculation bubble downstream with a large shear strain rate region until the boundary layer reattachment. This problem was chosen to compare the monolithic and

the segregated solvers because the varying velocities over the domain makes the solve challenging for monolithic solves, which often involve several nonlinear iterations.

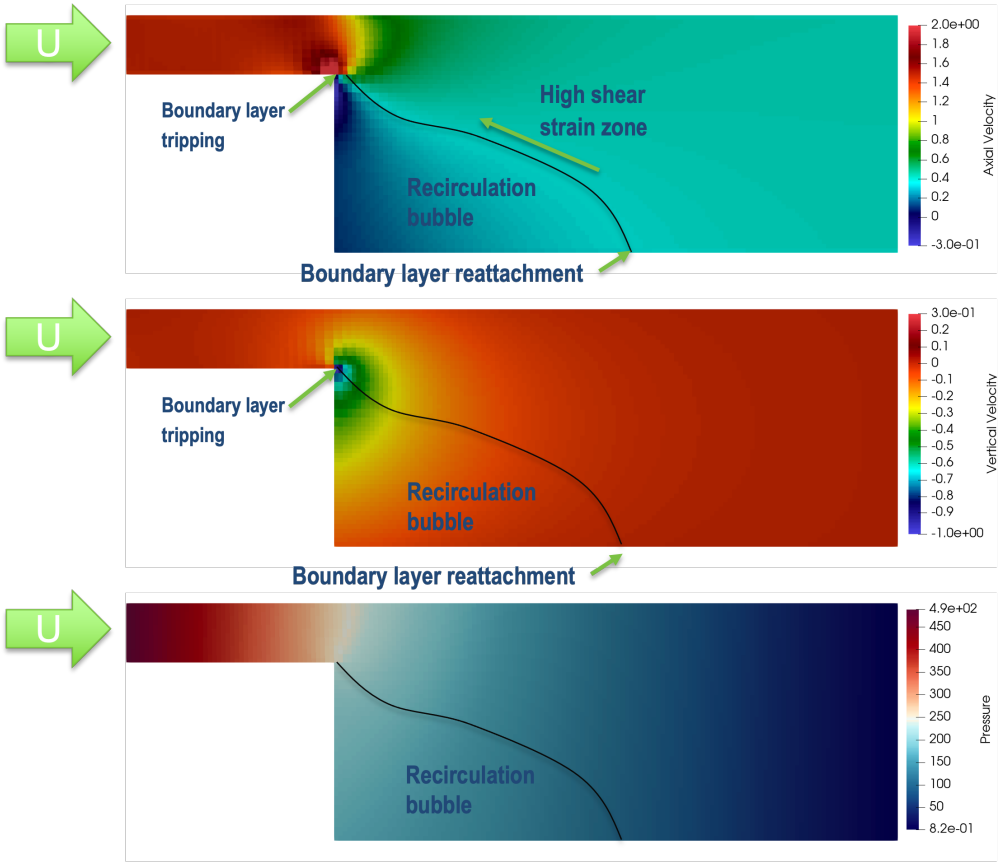

Figure 17: Example of the flow field for a backward-facing step with a Reynolds number of 20,000 at the inlet.

Two geometries are analyzed, a 2D and 3D. The baseline meshes for each of these geometries is presented in Figure 18. Uniform refinements are added on top of these meshes to compare the scaling of the monolithic and segregated solvers. In all cases, inlet velocity and viscosity are set so that the Reynolds number at the inlet, using the distance between the top and bottom walls as a characteristic distance, is equal to 20,000.

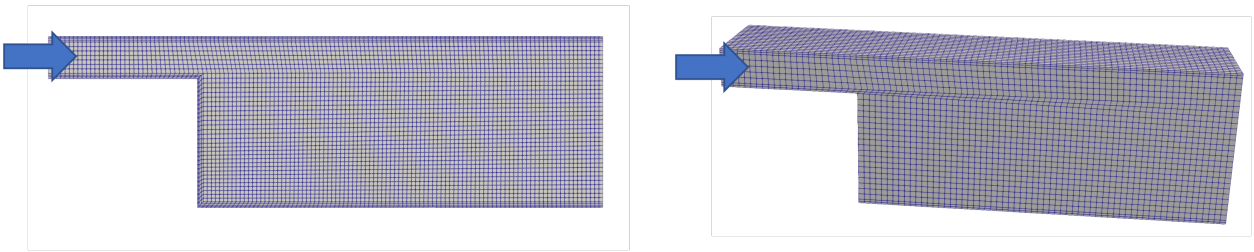

Figure 18: Baseline meshes used for the 2D and 3D backward-facing step cases. *Left*: 2D case. *Right*: 3D case.

The speedup and relative mean-squared difference in the velocity magnitude field for the 2D and 3D backward-facing steps as a function of the number of uniform refinements is presented in Figure 19. In every case, we obtain a positive speedup as measured by wall time. This means that the segregated solve simulations always take less time than the monolithic ones. Nonetheless, we observe that the speedup decreases as more uniform refinements are performed on the mesh, which increases the number of degrees of freedom. This is mainly because the previously mentioned *MultiAppTransfers* become progressively more expensive as we increase the number of degrees of freedom in the mesh. The speedup is expected to be closer to uniform after eliminating the need for *MultiAppTransfers* for executing segregated solvers. Finally, we observe that the relative mean-squared differences are in the order of 10^{-3}. These differences are expected since they are in the order of the error incurred in enforcing incompressibility as a constraint via the pressure field. No systematic bias is observed between the segregated and monolithic solves with increasing mesh density.

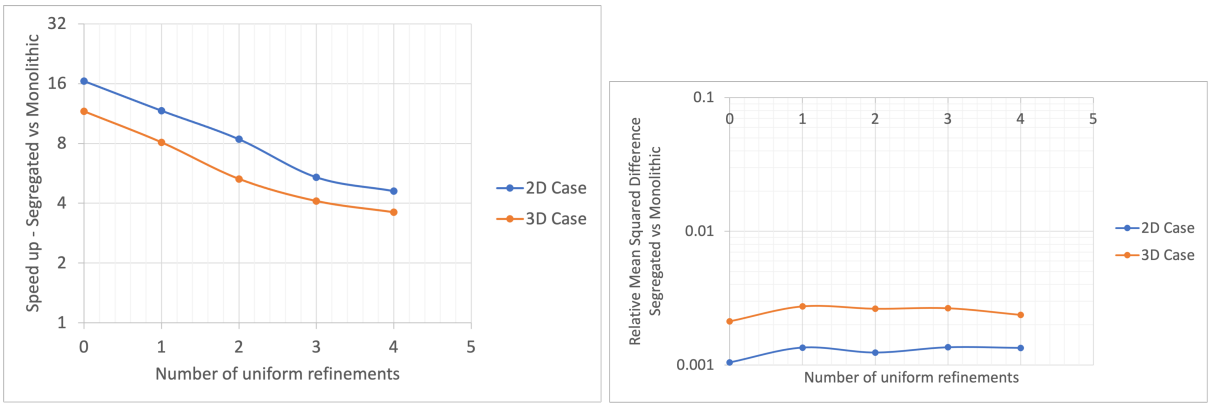

Figure 19: *Left*: wall-time speedup of solving the problem with a segregated vs a monolithic solve, positive speedups indicate that the segregated solver is faster. *Right*: relative mean-squared difference between segregated and monolithic solve for the velocity magnitude fields, similar differences were obtained for pressure.

6. Summary

This report details significant algorithmic improvements in MOOSE's finite-volume capabilities. Four tasks were completed: implementation of a weakly compressible finite-volume formulation in MOOSE's Navier-Stokes module, implementation of a monotonic interpolation scheme at discontiuous interfaces of body forces (with a focus on Darcy/Forchheimer-type body forces) and porosities, implementation of a skew correction for cell-to-face value interpolation, and implementation of a SIMPLE segregated solver in MOOSE.

The implementation of a weakly compressible formulation in MOOSE's Navier-Stokes module was completed. The weakly compressible implementation applies almost without restrictions to advanced reactor flow problems and supports efficient solutions in low-flow, buoyancy-driven flow regimes. The performance of the weakly compressible formulation was demonstrated for a differentially heated cavity with a large Rayleigh number. Previous attempts at solving this problem using the fully compressible formulation were unsuccessful.

Coarse-mesh thermal-hydraulics uses porosity and Darcy/Forchheimer drag forces to model nuclear reactors on the engineering scale. Previously, MOOSE's finite-volume interpolation methods showed oscillations at interfaces where porosity and/or body forces are discontinuous. These oscillations compromise accuracy, stability, and reliability of the coarse-mesh thermal-hydraulics models in Pronghorn. In this report, we identified the problems leading to the non-monotonic interpolation as:

- Linear interpolation of pressure to faces is inaccurate at the described discontinuities, leading to an overestimation of pressure gradients in the cell before a reduction in porosity and an underestimation in the cell right after the reduction. This leads to an nonphysical acceleration before and deceleration right after the interface.
- Body forces were not included in the RCI.

The following methods were implemented to achieve monotonic solutions at interfaces:

- Smoothing of porosity and body forces using Moukalled's cell-face-cell interpolation [13].
- Inclusion of body forces in the RCI.

- Modification of pressure interpolation at the interface by including body force terms.

These modifications lead to a practically monotonic interpolation scheme.

We implemented a skew correction for the interpolation of face values from cell values. For significantly skewed meshes, the skew correction improves the order of convergence of MOOSE's FVM significantly, as shown by a MMS test problem. The method is instrumental for an accurate interpolation of the face values of velocity and computation of cell gradients of pressure via the Green-Gauss method.

Currently, the standard solution algorithm in MOOSE's fluid capabilities is a monolithic solve. While monolithic solves can improve stability in certain applications, they are generally less efficient and stable than segregated solvers. For improving MOOSE's and Pronghorn's efficiency, we implemented a SIMPLE solution method in MOOSE and tested it for a backward-facing step problem. We observed a reduction in run time of a factor of 10 or more for coarse meshes.

REFERENCES

[1] A. J. Novak, R. W. Carlsen, S. Schunert, P. Balestra, D. Reger, R. N. Slaybaugh, and R. C. Martineau, "Pronghorn: A multidimensional coarse-mesh application for advanced reactor thermal hydraulics," *Nuclear Technology*, vol. 207, no. 7, pp. 1015–1046, 2021.

[2] C. J. Permann, D. R. Gaston, D. Andrš, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, and R. C. Martineau, "MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, vol. 11, p. 100430, 2020.

[3] E. Merzari, H. Yuan, M. Min, D. Shaver, R. Rahaman, P. Shriwise, P. Romano, A. Talamo, Y.-H. Lan, D. Gaston, R. Martineau, P. Fischer, and Y. Hassan, "Cardinal: A lower-length-scale multiphysics simulator for pebble-bed reactors," *Nuclear Technology*, vol. 207, no. 7, pp. 1118–1141, 2021.

[4] 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.

[5] Y. Wang, S. Schunert, J. Ortensi, V. Laboure, M. DeHart, Z. Prince, F. Kong, J. Harter, P. Balestra, and F. Gleicher, "Rattlesnake: A moose-based multiphysics multischeme radiation transport application," *Nuclear Technology*, vol. 207, no. 7, pp. 1047–1072, 2021.

[6] Shemon, E.R. and et al, "Proteus-sn user manual revision 1.2," Tech. Rep. ANL/NE-14/6, Argonne National Laboratory, Nov. 2014.

[7] Lee, C. H. and et al, "Mc2-3: Multigroup cross section generation code for fast reactor analysis," Tech. Rep. ANL/NE-11/41 (Rev. 3), Argonne National Laboratory, Aug., 2018.

[8] R. L. Williamson, J. D. Hales, S. R. Novascone, G. Pastore, K. A. Gamble, B. W. Spencer, W. Jiang, S. A. Pitts, A. Casagranda, D. Schwen, A. X. Zabriskie, A. Toptan, R. Gardner, C. Matthews, W. Liu, and H. Chen, "Bison: A flexible code for advanced simulation of the performance of multiple nuclear fuel forms," *Nuclear Technology*, vol. 207, no. 7, pp. 954–980, 2021.

[9] R. F. J. L. Guillaume L. Giudicelli, Alexander D. Lindsay, "Neams-th-crab," Tech. Rep. INL/EXT-21-62895, Idaho National Laboratory, 2021.

[10] P. B. R. C. J. O. D. G. M. D. A. A.-J. A. J. N. Guillaume Giudicelli, Alexander Lindsay, "Coupled multiphysics transient simulations of the mk1-fhr reactor using the finite volume capabilities of the moose framework," in *Mathematics and Computation for Nuclear Science and Engineering*, American Nuclear Society, 2021.

[11] J. W. Peterson, A. D. Lindsay, and F. Kong, "Overview of the incompressible navier–stokes simulation capabilities in the moose framework," *Advances in Engineering Software*, vol. 119, pp. 68–92, 2018.

[12] S. Schunert, G. Giudicelli, A. Lindsay, P. Balestra, 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," External report INL/EXT-21-63189, Idaho National Laboratory, 2021.

[13] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, vol. 113. Springer, 2016.

[14] S. Domino, "Sierra low mach module: Nalu theory manual 1.0," *SAND2015-3107W, Sandia National Laboratories Unclassified Unlimited Release (UUR)*, vol. 30, 2015.

[15] G. de Vahl Davis, "Natural convection of air in a square cavity: a bench mark numerical solution," *International Journal for numerical methods in fluids*, vol. 3, no. 3, pp. 249–264, 1983.

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

[17] A. Montlaur, S. Mendez, J. Peraire, and A. Huerta, "Discontinuous galerkin methods for the navier–stokes equations using solenoidal approximations," *International Journal for Numerical Methods in Fluids*, vol. 64, 10 2009.

[18] D. R. Gaston, C. J. Permann, J. W. Peterson, A. E. Slaughter, D. Andrš, Y. Wang, M. P. Short, D. M. Perez, M. R. Tonks, J. Ortensi, L. Zou, and R. C. Martineau, "Physics-based multiscale coupling for full core nuclear reactor simulation," *Annals of Nuclear Energy*, vol. 84, pp. 45–54, 2015.