INL/CON-23-75464-Revision-0

Development of Segregated Thermal-Hydraulics Solvers in MOOSE

June 2024

Peter German, Alexander D Lindsay, Mauricio Eduardo Tano Retamales, Ramiro Oscar Freile, Sebastian Schunert, Guillaume Louis Giudicelli

Changing the World's Energy Future

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

INL/CON-23-75464-Revision-0

Development of Segregated Thermal-Hydraulics Solvers in MOOSE

Peter German, Alexander D Lindsay, Mauricio Eduardo Tano Retamales, Ramiro Oscar Freile, Sebastian Schunert, Guillaume Louis Giudicelli

June 2024

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

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

Development of Segregated Thermal-Hydraulics Solvers in MOOSE

Péter German[a], Alexander D. Lindsay[a], Mauricio E. Tano[b], Guillaume L. Giudicelli[a], Ramiro O. Freile[c], and Sebastian Schunert[c]

[a]*Computational Frameworks Department, Idaho National Laboratory, 1955 N Fremont Ave, Idaho Falls, ID 83415, peter.german@inlgov, alexander.lindsay@inl.gov, guillaume.giudicelli@inl.gov*

[b]*Thermal Fluid Systems Methods & Analysis, Idaho National Laboratory, 1955 N Fremont Ave, Idaho Falls, ID 83415, mauricio.tanoretamales@inl.gov*

[c]*Reactor Physics Methods & Analysis, Idaho National Laboratory, 1955 N Fremont Ave, Idaho Falls, ID 83415, ramiro.freile@inl.gov, sebastian.scunert@inl.gov*

INTRODUCTION

Simulating fluid flows is an essential part of the design and analysis of nuclear systems. Algorithms that can simulate flows at different fidelity levels are available in the Multiphysics Object-Oriented Simulation Environment (MOOSE) [1, 2] and MOOSE-based applications, such as Pronghorn [3], Pronghorn-Subchannel [4], RELAP-7 [5], and SAM [6]. Currently, significant effort is being invested in the development of coarse-mesh Computational Fluid Dynamics (CFD) capabilities within MOOSE and Pronghorn for simulating Generation IV nuclear reactors. Traditionally, the solution algorithms in MOOSE have relied on Newton or quasi-Newton methods (such as the preconditioned Jacobian-free Newton-Krylov method) where residuals and Jacobians (or approximations thereof) are constructed [7]. Both Newton and quasi-Newton methods require solving a linear system at each nonlinear Newton iteration with the Jacobian as the system matrix. The Jacobian contains blocks originating from all variables in the problem (i.e., for thermal-hydraulics, at least pressure, velocities, and temperature). Due to the formulation of the problem in a general multiphysics setting on an unstructured mesh, creating a good preconditioner for the linear system can be challenging, and thus many fluid applications have utilized direct solver-based methods, such as LU factorization. However, with increasing system size and complexity in multidimensional problems, the direct solution of linear systems becomes computationally expensive both in execution time and memory. For this reason, recent effort has focused on adapting segregated solution algorithms for CFD problems in MOOSE. These algorithms use fixed-point iteration between segregated systems whose assembly and preconditioning are easier than those of the monolithic system.

The first step in this effort has been adding the SIMPLE algorithm [8], which has been widely used by the fluid dynamics community for solving steady-state problems [9, 10]. This algorithm utilizes a Poisson equation for pressure, which can be derived using the original mass and momentum conservation equations. In this paper, we present the first numerical results obtained using the SIMPLE algorithm in MOOSE using two Molten Salt Reactor (MSR) models as test examples. First, a two-dimensional (2D) model serves as a basis for a detailed comparison between the monolithic and segregated solvers in terms of memory usage, solution time, and accuracy. The second example showcases a 3D simulation where the segregated solver significantly outperforms the monolithic approach.

THEORY

The steady-state Navier-Stokes equations describe the momentum and mass conservation of a system:

$$\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \left(\mu \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right)\right) - \nabla p + \rho \boldsymbol{g} - W(\boldsymbol{u}), \quad (1a)$$

$$\nabla \cdot (\rho \boldsymbol{u}) = 0 \,, \quad (1b)$$

where $\boldsymbol{u}$ is the velocity vector field, p is the pressure field, μ is the dynamic viscosity, ρ is the density, $\boldsymbol{g}$ is the gravitational acceleration vector, and $\boldsymbol{W}(\boldsymbol{u})$ is a homogenized, volumetric momentum source that may depend on velocity. In this work, we assume incompressible fluids with constant viscosity, which means that $\nabla \cdot \mu \, (\nabla \boldsymbol{u})^T = 0$.

Traditionally, the fluid dynamics solvers in MOOSE and Pronghorn solved these equations using Newton's method:

$$\mathbf{J}(\boldsymbol{x}) \delta \boldsymbol{x} = -\mathbf{R}(\boldsymbol{x}) \quad (2)$$

where $\boldsymbol{x} = [\boldsymbol{u}, p]^T$ is a compound vector. For discretization in space, MOOSE currently supports finite element and cell-centered finite volume methods. We use the latter in this work since it has become the workhorse of MOOSE for multidimensional fluid dynamics simulations. The solution of the momentum-continuity equations in the above-mentioned form, however, poses two significant challenges:

- It introduces a saddle-point problem in the Jacobian due to the continuity equation serving as a constraint, which leads to difficulties when the Newton system is solved. This can be addressed through direct solution-based preconditioners (i.e., LU), which can become computationally prohibitive with large problem sizes.
- Cell-centered finite volume methods may exhibit checkerboarding due to the discretization of the pressure gradient and mass advection terms. This necessitates using the Rhie-Chow interpolation [11], which can cause a significant increase in nonlinearity with advanced advection schemes, leading to convergence issues in the nonlinear iteration.

To alleviate these problems, many fluid dynamics solvers use segregated solution algorithms, which provide a natural solution for both challenges. These segregated approaches can either be introduced at a preconditioning level (Schur-complement-based [12] split preconditioners) or by iteratively

solving segregated linear equations. In this paper, we follow the second approach by adapting the SIMPLE algorithm, which was first described in [8] and discussed in detail in [9, 10]. It can be derived using the linearized momentum equation in a semidiscretized form:

$$\mathbf{A}\left(\boldsymbol{u}^{n-1}\right)\boldsymbol{u}^{\boldsymbol{n}} + \mathbf{H}\left(\boldsymbol{u}^{n-1}\right) = -\boldsymbol{\nabla} p \,, \tag{3}$$

where $\mathbf{A}\left(\boldsymbol{u}^{n-1}\right)$ describes the diagonal terms of the linearized momentum equation computed using the previous velocity iterate $\boldsymbol{u}^{n-1}$, while $\mathbf{H}\left(\boldsymbol{u}^{n-1}\right)$ includes the off-diagonal terms and non-pressure-related source terms on the right-hand side. We can then multiply by the inverse of the diagonal matrix:

$$\boldsymbol{u}^{\boldsymbol{n}} + \mathbf{A}^{-1}\left(\boldsymbol{u}^{n-1}\right)\mathbf{H}\left(\boldsymbol{u}^{n-1}\right) = -\mathbf{A}^{-1}\left(\boldsymbol{u}^{n-1}\right)\boldsymbol{\nabla} p \,, \tag{4}$$

which can then be simplified using the continuity constraint (assuming incompressible fluids $\nabla \cdot \boldsymbol{u}^{n} = 0$):

$$\nabla \cdot \mathbf{A}^{-1}\left(\boldsymbol{u}^{n-1}\right)\mathbf{H}\left(\boldsymbol{u}^{n-1}\right) = -\nabla \cdot \mathbf{A}^{-1}\left(\boldsymbol{u}^{n-1}\right)\boldsymbol{\nabla} p \,, \tag{5}$$

where the inversion of $\boldsymbol{A}$ is trivial, given that it is a diagonal matrix. This results in two equations, one for velocity and another for pressure, which can be solved in a predictor-corrector manner:

- Solve Eq. (3) to predict a velocity, which does not respect the continuity equation. For this, the pressure field is fixed at the previous guess.
- Solve Eq. (5) for the next pressure guess, which can be used to correct the velocity field to respect continuity.
- Correct the velocity field using the rearranged form of Eq. (4). To tackle the checker boarding problems encountered with cell-centered finite volume methods, this correction involves the face velocity instead of the cell velocity:

$$\left(\boldsymbol{u}^{n+1}\right)_f = -\left(\mathbf{A}^{-1}\mathbf{H}\right)_f - \left(\mathbf{A}^{-1}\boldsymbol{\nabla} p\right)_f \,, \tag{6}$$

  This face velocity can then be used in the discretized advection terms directly.

In many cases, this iteration may exhibit convergence issues [8], which can be resolved by relaxing the update in pressure:

$$p^{n+1} = \lambda p^{n+1} + (1-\lambda)p^{n} \,, \tag{7}$$

where $\lambda \in (0, 1]$ is a relaxation parameter. Furthermore, the momentum equation can be relaxed as well to help iterative convergence. For more information on these aspects, see [10].

RESULTS AND ANALYSIS

In this section, we used two MSR concepts to compare the original, monolithic solution algorithm with LU preconditioning to the new segregated approach.

Two-Dimensional Molten Salt Reactor Test

The first problem is designed to be a verification example to check if the two solution algorithms converge to the same solution with mesh refinement. It showcases a 2D axisymmetric model of the Molten Salt Fast Reactor (MSFR) [13] openly available in the Virtual Test Bed (VTB) [14]. The model consists of 2,068 cells without refinement. Our simulations used an increased dynamic viscosity of μ=20 $Pa \cdot s$, which results in an approximate Reynolds number of Re = 160 in the core cavity. This is due to the lack of more complex turbulence models for segregated solvers at the time of this work. Further simplifications included solving the momentum-continuity system alone and neglecting the friction-based drag forces in the heat exchanger. The reference results for a uniformly refined mesh are presented in Fig. 1.

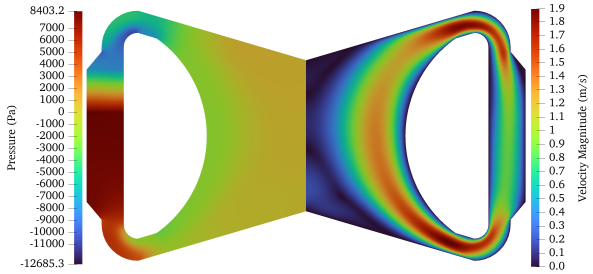

Fig. 1. Reference solutions of velocity and pressure used for the comparison of the two solution approaches.

The convergence of the results through three uniform refinement steps is presented in Fig. 2. We see that the relative error norm decreases between the results from the two different solution approaches, which indirectly verifies the implementation of the segregated velocity-pressure coupling in MOOSE.

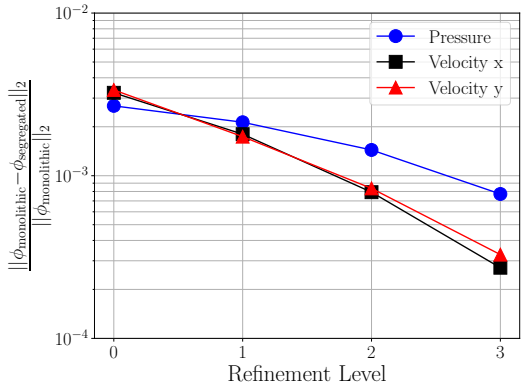

Fig. 2. Diminishing difference in the velocity and pressure fields obtained by the two different solvers ($\phi \in p, u_x, u_y$).

As a next step, the memory consumption of the two implementations is presented in Figure 3. We see that the segregated solver uses only a fraction of the memory required for the monolithic approach. The reason behind this is that the LU-decomposition-based solver fills up the matrix band with entries, resulting in a more dense preconditioner.

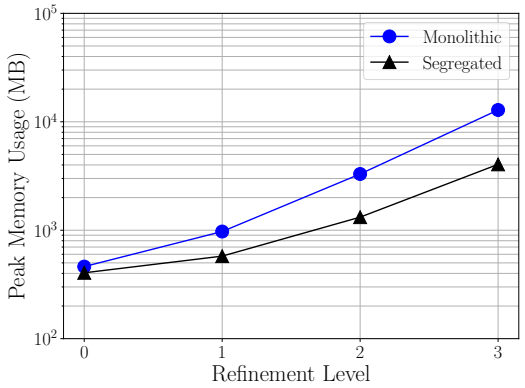

Fig. 3. Memory consumption as function of the refinement for the 2D model of the MSFR.

In terms of computational time, solving this laminar problem is faster with the monolithic solver for all four refinement levels. For the lowest refinement level, the monolithic solver takes 0.61 s to complete one Newton iteration, in contrast to the 0.18 s needed for a SIMPLE iteration. These numbers have been obtained using a single core of an AMD® Ryzen Threadripper 3990x 64-core processor with Ubuntu 20.04.6 operating system. Even though the segregated solver is considerably faster per iteration, it needs a significantly higher number of iterations to reach the same accuracy level. For the lowest refinement level, the monolithic solver needed six Newton iterations to reach a 10^{-5} absolute tolerance on the residual, while the segregated solver needed 104. This discrepancy can get even worse since the number of iterations needed for the segregated solver to converge increases with mesh refinement, whereas the number of iterations for the monolithic solver stays roughly the same. Nevertheless, with increasing complexity and the inclusion of other physics besides the momentum and mass conservation equations, the memory consumption and runtime of the monolithic solver increase even more, making the segregated solver viable.

Three-Dimensional Molten Salt Reactor Problem

The second example showcases a 3D MSR operating with chloride salt. The example has been adapted from [15] and describes a molten salt loop with a reflector attached to the core barrel. The bottom of the reactor core has a homogenized mixing plate modeled as a porous medium with appropriate drag coefficients to regularize the flow field downstream. The parameters of the model are summarized in Table I, while the computational mesh is depicted in Figure 4. The mesh contains 438,072 cells altogether.

The simulated physics included the fluid flow with energy transport, heat exchange between the reflector and salt, and the heat conduction within the reflector. We note that the viscosity of the fluid was artificially increased in this case as well to simulate a laminar fluid flow. This resulted in an approximate Reynolds number of Re = 132 in the core cavity.

Figure 5 shows the flow field together with the fuel salt and reflector temperature distributions. Due to the low power level, the temperature of the fuel salt changes by less than 1 K.

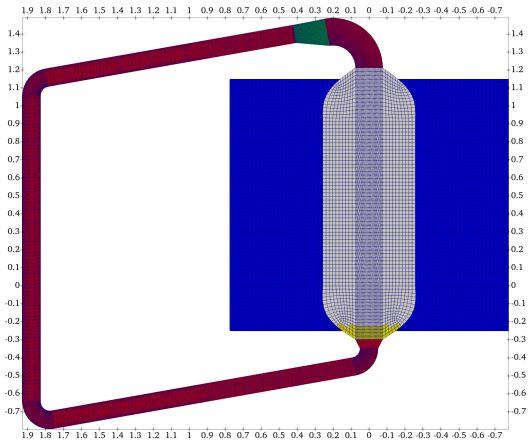

Fig. 4. Geometry and mesh of the 3D MSR concept (blue: reflector, yellow: mixing plate, green: pump, red: pipes, white: reactor core).

On the other hand, due to the low heat exchange coefficient, there is a large temperature gradient between the salt and surrounding structures.

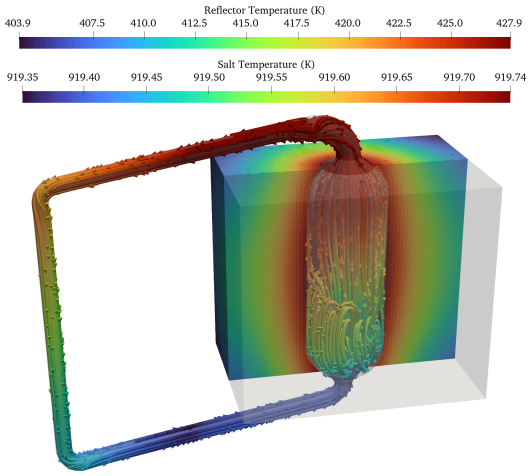

Fig. 5. Temperature and flow fields obtained from a segregated multiphysics computation of the 3D MSR concept.

The results were obtained using five cores of the same processor and same the operating system as in Section . The segregated solver obtained a 10^{-4} tolerance in approximately 169 minutes (257 iterations), using a maximum of 25 GB of system memory. This is equivalent to approximately 39.5 s/iteration. The monolithic solver, due to the implicit coupling of every variable and the resulting, relatively dense preconditioner, however, did not manage to solve the problem due to running out of memory after the first nonlinear iteration on a machine with 500 GB of system memory. The first nonlinear iteration took approximately 95 minutes, which is 150× slower per iteration than the segregated solver. We see that, for more complex problems, the segregated solver results in significant computational gains.

TABLE I. Input parameters used for simulating the molten chloride reactor concept [15].

Param. name	Value	Param. name	Value	Param. name	Value
Salt density	3,279 $\frac{kg}{m^3}$	Mixing plate fric. coeff.	(22, 66, 66) $\frac{kg}{m^4}$	Refl. thermal cond.	30 $\frac{W}{m K}$
Thermal power	1 kW	Salt thermal cond.	0.38 $\frac{W}{m\ K}$	Pipe-air heat ex. coeff.	10 $\frac{W}{m^2 K}$
Salt dynamic viscosity	5 $Pa \cdot s$	Salt specific heat	640 $\frac{J}{kg\ K}$	Pump fric. coeff.	(46.4, 46.4, 46.4) $\frac{kg}{m^4}$
Pump power	1,000 $\frac{kN}{m^3}$	External air temp.	300 K		

CONCLUSIONS

A segregated solution algorithm has been added to the Navier-Stokes module of MOOSE, which is capable of more efficiently solving complex steady-state multiphysics problems by operator splitting. The pressure-velocity coupling is based on the SIMPLE algorithm, which has been commonly utilized in other open-source and commercial software in the fluid dynamics community. Preliminary tests on MSR-based models show that, for small problems, the segregated solver is slower than the monolithic approach, mainly due to the larger number of iterations needed for convergence compared to Newton's method. On the other hand, the segregated solver outperforms the monolithic approach in terms of memory usage, which enables solving considerably larger problems.

Furthermore, by investigating the computational effort distribution in the segregated solver, we discovered that most of it is spent on assembling the linear systems. This is the consequence of reusing the already existing Jacobian and residual assembly loops. Therefore, future work is targeted at reducing these costs to improve computational efficiency.

ACKNOWLEDGMENTS

This manuscript has been authored by Battelle Energy Alliance, LLC under Contract No. DE-AC07-05ID14517 with the U.S. Department of Energy, with funding from the Nuclear Energy Advanced Modeling and Simulation program. Furthermore, this research made use of the resources of the High-Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

REFERENCES

1. A. D. LINDSAY, D. R. GASTON, C. J. PERMANN, J. M. MILLER, D. ANDRŠ, A. E. SLAUGHTER, F. KONG, J. HANSEL, R. W. CARLSEN, C. ICENHOUR, ET AL., "2.0-MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, **20**, 101202 (2022).
2. A. LINDSAY, G. GIUDICELLI, P. GERMAN, J. PETERSON, Y. WANG, R. FREILE, D. ANDRS, P. BALESTRA, M. TANO, R. HU, ET AL., "MOOSE Navier–Stokes module," *SoftwareX*, **23**, 101503 (2023).
3. A. J. NOVAK, L. ZOU, J. W. PETERSON, D. ANDRS, J. KELLY, R. N. SLAYBAUGH, R. C. MARTINEAU, and H. D. GOUGAR, "Pronghorn theory manual," Tech. rep., Idaho National Lab.(INL), Idaho Falls, ID (United States) (2018).
4. V. KYRIAKOPOULOS, M. E. TANO, and J. C. RAGUSA, "Development of a Single-Phase, Transient, Subchannel Code, within the MOOSE Multi-Physics Computational Framework," *Energies*, **15**, *11*, 3948 (2022).
5. R. A. BERRY, J. W. PETERSON, H. ZHANG, R. C. MARTINEAU, H. ZHAO, L. ZOU, D. ANDRS, and J. HANSEL, "RELAP-7 theory manual," Tech. Rep. INL/EXT-14-31366, Idaho National Lab. (INL), Idaho Falls, ID (United States) (2018).
6. R. HU, L. ZOU, G. HU, D. NUNEZ, T. MUI, and T. FEI, "SAM theory manual," Tech. rep., Argonne National Lab.(ANL), Argonne, IL (United States) (2021).
7. D. A. KNOLL and D. E. KEYES, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, **193**, *2*, 357–397 (2004).
8. S. V. PATANKAR and D. B. SPALDING, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," in "Numerical prediction of flow, heat transfer, turbulence and combustion," Elsevier, pp. 54–73 (1983).
9. H. JASAK, *Error analysis and estimation for the finite volume method with applications to fluid flows.*, Ph.D. thesis, Imperial College London (University of London) (1996).
10. F. JURETIC, *Error analysis in finite volume CFD*, Ph.D. thesis, Imperial College London (University of London) (2005).
11. C. M. RHIE and W.-L. CHOW, "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *AIAA journal*, **21**, *11*, 1525–1532 (1983).
12. D. LOGHIN and A. WATHEN, "Schur complement preconditioners for the Navier–Stokes equations," *International journal for numerical methods in fluids*, **40**, *3-4*, 403–412 (2002).
13. D. HEUER, E. MERLE-LUCOTTE, M. ALLIBERT, M. BROVCHENKO, V. GHETTA, and P. RUBIOLO, "Towards the thorium fuel cycle with molten salt fast reactors," *Annals of Nuclear Energy*, **64**, 421–429 (2014).
14. M. TANO, J. FANG, G. GIUDICELLI, and A. ABOU-JAOUDE, "Coupled Griffin-Prongorn-SAM Model of a Molten Salt Fast Reactor System Transient in the Virtual Test Bed," *Trans. Am. Nucl. Soc*, **127** (2022).
15. M. TANO, R. FREILE, R. DE OLIVEIRA, S. WALKER, and A. ABOU-JAOUDE, "Coupled steady-state, precursor distribution, and transient studies for open-core flow MCFRs," Tech. Rep. INL/RPT-23-73982, Idaho National Lab.(INL), Idaho Falls, ID (United States) (2023).