INL/JOU-23-72115-Revision-0



# MOOSE NavierStokes module

July 2023

Alexander D Lindsay, Guillaume Louis Giudicelli, Peter German, Yaqi Wang, Ramiro Oscar Freile, Paolo Balestra, Mauricio Eduardo Tano Retamales, Sebastian Schunert, David Andrs, Rui Hu, Ling Zou, John Peterson, Derek R Gaston, Cody J Permann

*Changing the World's Energy Future*


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

INL/JOU-23-72115-Revision-0

# MOOSE NavierStokes module

**Alexander D Lindsay, Guillaume Louis Giudicelli, Peter German, Yaqi Wang, Ramiro Oscar Freile, Paolo Balestra, Mauricio Eduardo Tano Retamales, Sebastian Schunert, David Andrs, Rui Hu, Ling Zou, John Peterson, Derek R Gaston, Cody J Permann**

**July 2023**

**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, DE-AC07-05ID14517, DE-AC07-05ID14517**

SoftwareX 23 (2023) 101503

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

# MOOSE Navier–Stokes module

Alexander Lindsay [a,*], Guillaume Giudicelli [a], Peter German [a], John Peterson [b], Yaqi Wang [a], Ramiro Freile [a,c], David Andrs [a], Paolo Balestra [a], Mauricio Tano [a], Rui Hu [d], Ling Zou [d], Derek Gaston [a], Cody Permann [a], Sebastian Schunert [a]

[a] Idaho National Laboratory, Idaho Falls, ID, 83415, United States of America
[b] Technology Department, Akselos, Inc., Houston, TX, 77043, United States of America
[c] Department of Nuclear Engineering, Texas A&M University, College Station, TX 77843-3133, United States of America
[d] Argonne National Laboratory, Lemont, IL, 60439, United States of America

ARTICLE INFO

*Article history:*
Received 12 May 2023
Received in revised form 21 July 2023
Accepted 14 August 2023

*Keywords:*
Navier–Stokes
Finite elements
Finite volumes
Porous medium

ABSTRACT

The MOOSE Navier–Stokes module solves mass, momentum, energy, and passive scalar conservation equations in the context of fluid flow. The module supports solution of these equations in both free flow and porous medium contexts and for a range of fluid compressibility. The conservation equations can be discretized in space using continuous Galerkin finite elements or with cell centered finite volumes.



## Code metadata

| | |
|---|---|
| Current code version | v1.0.0 |
| Permanent link to code/repository used for this code version | https://github.com/ElsevierSoftwareX/SOFTX-D-23-00292 |
| Permanent link to Reproducible Capsule | |
| Legal Code License | LGPL 2.1 |
| Code versioning system used | git |
| Software code languages, tools, and services used | C++ |
| Compilation requirements, operating environments & dependencies | gcc or clang compilers, Linux, MacOS, WSL |
| If available Link to developer documentation/manual | https://mooseframework.inl.gov/modules/navier_stokes/index.html |
| Support email for questions | https://github.com/idaholab/moose/discussions/categories/q-a-modules-navier-stokes |

## 1. Motivation and significance

The greatest strength of the MOOSE `navier_stokes` module is its presence within the MOOSE ecosystem [1]. MOOSE is designed to enable complex multiphysics simulations with relatively little work from the user. It is built on top of the finite element library libMesh [2], which supports various solver backends; the solver backend which MOOSE leverages is PETSc [3] which has the capability to interface to other solver packages such as hypre [4]. Through its underlying libraries and the MOOSE framework itself, every MOOSE module and application inherit capabilities such as: parallelism (both MPI and threads), unstructured mesh, adaptivity (both h- and p-refinement), automatic differentiation [5], implicit and explicit time integration, ability to segregate physics occurring on disparate spatial and temporal scales using the `MultiApp` system [6], and advanced (non)linear solver technologies. By building from the MOOSE framework, fluid flow simulation using the `navier_stokes` module can be easily coupled to, for instance, thermomechanical physics at a fluid structure interface, neutronics physics in a nuclear reactor simulation, or to electromagnetic modeling in a magnetohydrodynamic (MHD) drive. This capability to seamlessly interface other physics with fluid flow simulation opened the door to simulations of advanced nuclear reactor types (including molten-salt [7] and high-temperature gas reactors) that were not achievable before.

In addition to its role in advanced multiphysics simulation, the fluid flow physics in `navier_stokes` integrates seamlessly into the native meshing, output, and postprocessing capability of

* Corresponding author.
*E-mail address:* alexander.lindsay@inl.gov (Alexander Lindsay).

https://doi.org/10.1016/j.softx.2023.101503

the MOOSE framework, allowing for an all-in-one approach to modeling and simulation, somewhat akin to commercial products like ANSYS and COMSOL. Whether a user is performing complex multiphysics analysis or setting up a canonical fluid flow case like the lid-driven cavity, the required plain text input for the mesh, module physics, and output is on the order of 50 lines. This allows researchers to rapidly address their domain-area questions. Research works citing `navier_stokes` include: [8–35]. Some application areas from the citing studies include flow in fractures, fluid dynamics during powder bed fusion processes, general study of fluid structure interactions, non-isothermal polymer filling processes, electrohydrodynamic flow generation, and molten salt reactor dynamics.

## 2. Software description

### 2.1. Software architecture

As is the case for the MOOSE framework, the `navier_stokes` module is programmed in C++ with testing infrastructure programmed in Python. Finite element weak forms are implemented with `Kernel` objects for volumetric terms and `IntegratedBC` objects for boundary surface terms. Some boundary condition types (e.g., Dirichlet) may be strongly enforced using `NodalBC` objects. For finite volume discretizations, `FVElementalKernel` objects implement volumetric terms, `FVFluxKernel` objects implement internal face and Dirichlet boundary face fluxes, and `FVFluxBC` objects implement non-Dirichlet boundary face fluxes (e.g., Neumann condition fluxes).

In a typical `navier_stokes` simulation, Newton's method is used to solve algebraic systems of equations created through finite element or finite volume discretization of the mass, momentum, and potentially the energy and passive scalar conservation equations. Newton's method requires computation of a residual vector and a description of the derivatives of the residual vector with respect to the degrees of freedom, called the Jacobian. The Jacobian may be provided in a matrix form or its *action* on a vector approximated using finite differencing of residual evaluations. MOOSE allows both Jacobian representations through executioner solve-type choices. Even when the Jacobian action is approximated using finite differencing, efficient solution of the linearized algebraic system of equations via a Krylov method typically requires preconditioning. The preconditioning process is usually constructed from a preconditioning matrix that is generally some approximate representation of the full Jacobian matrix. In MOOSE, residual and Jacobian/preconditioner entries are computed for finite element discretizations using element and node loops, calling `Kernel`, `IntegratedBC`, and `NodalBC` objects to compute elemental or nodal residuals. Finite volume discretizations employ element loops for volumetric terms (`FVElementalKernel` objects) and face loops for face flux terms (`FVFluxKernel` and `FVFluxBC` objects). Element, face, and nodal residuals or Jacobians are summed into or set in global data structures of base type `NumericVector` and `SparseMatrix` respectively held by libMesh `System` objects. For MOOSE simulations, the derived types–`PetscVector` and `PetscMatrix`–are wrappers of PETSc data structures, `Vec` and `Mat`. After computation of the global residual and preconditioning/Jacobian matrix, a linear solve is performed using PETSc `KSP` (linear solver) and `PC` (preconditioning) objects to obtain a Newton update. This represents completion of one nonlinear iteration. Nonlinear iterations proceed until a nonlinear relative or absolute convergence tolerance is reached.

### 2.2. Software functionalities

#### 2.2.1. Governing equations

The MOOSE `navier_stokes` module solves conservation equations, including mass, momentum, energy, and passive scalars, in the context of fluid flow. Here we present equations corresponding to an incompressible or nearly incompressible fluid, which is the set used by most nuclear reactor applications. Conservation of mass is expressed by

$$\frac{\partial \epsilon \rho}{\partial t} + \nabla \cdot \rho \vec{v}_D = 0 \tag{1}$$

where $\epsilon$ is the porosity of the medium, $\rho$ is the fluid velocity, and $\vec{v}_D = \epsilon \vec{v}$ is the superficial velocity. Conservation of momentum is described by

$$\frac{\partial \rho \mathbf{v}_D}{\partial t} + \nabla \cdot \left( \frac{\rho}{\epsilon} \mathbf{v}_D \otimes \mathbf{v}_D \right) = \nabla \cdot \left( \mu \nabla \frac{\mathbf{v}_D}{\epsilon} \right) - \epsilon \nabla p + \epsilon \left( \mathbf{F}_g + \mathbf{F}_f \right) \tag{2}$$

where $\mu$ is the dynamic viscosity, possibly including contributions from turbulence, $p$ is the pressure, and $F_g$ and $F_f$ express body forces due to gravity and friction respectively. Conservation of energy is given by

$$\frac{\partial \epsilon \rho c_p T_f}{\partial t} + \nabla \cdot (\mathbf{v}_D \rho c_{pf} T_f) = \nabla \cdot (\kappa_f \nabla T_f) - \alpha (T_f - T_s) \tag{3}$$

where $c_p$ is the specific heat capacity of the fluid, $T_f$ is the fluid temperature, $\kappa_f$ is the effective thermal conductivity, and *alpha* is the convective heat transfer coefficient between solid and fluid phases in a porous medium simulation. When modeling heat transport in a porous medium, an additional energy equation for the solid is included

$$\frac{\partial (1-\epsilon) \rho c_{ps} T_s}{\partial t} = \nabla \cdot (\kappa_s \nabla T_s) + \alpha (T_f - T_s) + (1-\epsilon) \dot{Q} \tag{4}$$

where $c_{ps}$ is the specific heat capacity of the solid, $\kappa_s$ is the effective thermal conductivity of the solid, and $\dot{Q}$ is a volumetric heating source (or sink); an example heat source could be radiation absorption from neutrons or gammas in a nuclear reactor core. Conservation of a passive scalar is expressed by

$$\frac{\partial \epsilon u}{\partial t} + \nabla \cdot u \vec{v}_D = 0 \tag{5}$$

where $u$ is the concentration of the passive scalar. The fluid energy equation is modified when treating a fluid as fully compressible:

$$\frac{\partial \rho \epsilon e_t}{\partial t} + \nabla \cdot \left( \rho \vec{v}_D \left( e_t + \frac{p}{\rho} \right) \right) = -\alpha \left( T_f - T_s \right) \tag{6}$$

where $e_t = e + v^2/2$ with $e$ the specific internal energy of the fluid. Note that we have presented general equations for simulating fluid flow in a porous medium. Free flow can be obtained by simply setting $\epsilon = 1$ in the above equation set.

Typically, these equations are solved in a single monolithic system using Newton's method. However, the equations can be segregated using the MOOSE `MultiApp` system [1,6]. Naming prefixes of the different objects in the module as well as the fluid flow conditions they can simulate are summarized in Tables 1 and 2. Table 1 describes whether the objects use hand-coded or automatic differentiation (AD) to compute Jacobians for Newton's method, whether the objects are appropriate for incompressible, weakly compressible, or fully compressible flow, and what kind of turbulence models the objects support. Table 2 describes whether the objects are spatially discretized using continuous Galerkin finite elements or finite volumes, what kind of discretization strategy is used for advection terms, and whether the objects can be used to model porous media. For object sets that support porous media, both Darcy and Forchheimer friction

**Table 1**
Part 1, summary of `navier_stokes` capabilities. The Jacobian column denotes whether the object's Jacobians are computed "by-hand" or via automatic differentiation (AD) [5].

| Prefix | Jacobian | Compressibility | Turbulence |
|---|---|---|---|
| INSFE | Hand | None | Mixing length |
| INSAD | AD | None | Smagorinsky |
| PINSFE | Hand | Weak | Mixing length |
| NS | Hand | Full | None |
| INSFV | AD | None | Mixing length/k-$\epsilon$ |
| WCNSFV | AD | Weak | Mixing length/k-$\epsilon$ |
| PINSFV | AD | None | None |
| CNSFVHLLC | AD | Full | None |
| PCNSFVHLLC | AD | Full | None |
| PCNSFVKT | AD | Full | None |

**Table 2**
Part 2, summary of `navier_stokes` capabilities. Summary of acronyms: Continuous Galerkin Finite Elements (CGFE); Finite Volumes (FV); Streamline-Upwind Petrov–Galerkin (SUPG) [36]; Rhie–Chow (RC) [37]; Harten–Lax–van Leer Contact (HLLC) [38]; Kurganov–Tadmor (KT) [39,40].

| Prefix | Method | Advection strategy | Porous |
|---|---|---|---|
| INSFE | CGFE | SUPG | No |
| INSAD | CGFE | SUPG | No |
| PINSFE | CGFE | SUPG | Yes |
| NS | CGFE | SUPG | No |
| INSFV | FV | RC; limiters | No |
| WCNSFV | FV | RC; limiters | No |
| PINSFV | FV | RC; limiters | Yes |
| CNSFVHLLC | FV | HLLC, upwind | No |
| PCNSFVHLLC | FV | HLLC, upwind | Yes |
| PCNSFVKT | FV | KT, limiters | Yes |

relationships are available, and the user can implement the correlations they wish to use to compute the Darcy and Forchheimer factors. For porous media simulations in which the porosity varies from region to region, the incompressible finite volume object set (PINSFV) automatically constrains the velocity and pressure fields in order to satisfy the Bernoulli relation. Other pressure-drop relations at porosity change interfaces will be implemented in the future.

The incompressible and weakly compressible finite volume implementation in `navier_stokes` includes a variety of options for modeling turbulence: a zero equation model based on mixing length and a two-equation method based on the turbulent kinetic energy ($k$) and its dissipation rate ($\epsilon$) [41]. Advection terms can use different limiting schemes for determining the value of advected quantities on faces. First order upwind and second order geometrically weighted averaging are available as well as total-variation diminishing (TVD) [42] schemes, van Leer [43], and min-mod [44], which mix upwind and weighted average interpolations as necessary to maximize accuracy while preventing oscillations in the advected field. The Ladyzhenskaya–Babuska–Brezzi (LBB) [45] condition relevant to incompressible fluid flow is handled for the collocated finite volume implementations using a Rhie–Chow interpolation [37]; for continuous finite elements with equal order velocity and pressure bases, the condition is handled using a Pressure-Stablized Petrov–Galerkin (PSPG) [46] method. For the finite volume implementation handling fully compressible flows, the user can choose two different advection treatments, Harten–Lax–van Leer-Contact (HLLC) [38] or Kurganov–Tadmor (KT) [39,40]. The former does not currently support second-order reconstruction of (sided) face values (e.g., the treatment is piecewise constant) while the latter does.

### 2.2.2. Time integration

A unique feature of the MOOSE `navier_stokes` module is that it allows monolithic fully implicit solutions of the transient Navier–Stokes equations. This eases restrictions on the allowable time-step size that are associated with explicit time integration schemes. For a nonlinear system of equations, calculation of the coefficients for a Rhie–Chow interpolation is not as simple as for a linear system; for a linear system, the $a$ coefficients correspond simply to the diagonals of the Jacobian. To simplify the computation of $a$ coefficients in the nonlinear setting, the MOOSE automatic differentiation system is used [5]. While implicit time integration can be used to avoid the Courant–Friedrichs–Lewy (CFL) [47,48] condition, MOOSE `navier_stokes` does support explicit time integration schemes such as forward Euler and TVD Runge–Kutta schemes.

### 2.2.3. Porosity jump handling

An important strength of `navier_stokes` is its ability to model porous media flow. Perhaps the flagship open-source software for simulating fluid flow, OpenFOAM [49], has porous flow simulation capability; however, it does not contain the oscillation suppression tools around discontinuous porosity changes like MOOSE does. These methods include repeated interpolations and reconstructions of the discontinuous porosity field to create a smooth, continuous porosity field. Additionally, the `PINSFV-MomentumFrictionCorrection` finite volume kernel may be used, which adds a residual equal to the product of a geometric factor, Darcy/Forchheimer coefficient, and superficial velocity gradient. The geometric factor ensures that the friction correction goes to zero as the mesh is refined. Finally, a `BernoulliPressureVariable` class may be used which essentially divides the domain into two subdomains on either side of the discontinuity, and then automatically imposes a pressure drop according to Bernoulli's equation across the interface. This latter treatment is only appropriate for an inviscid or nearly-inviscid fluid. The `BernoulliPressureVariable` treatment has similarities to the baffle treatment in GenFOAM [50].

### 2.2.4. Solution of the discretized equations

Scalable solution of the Navier–Stokes equations is an active area of research. Monolithic multigrid with an appropriate coupled relaxation method, such as the algorithms described by Vanka [51] or Braess and Sarazin [52], has been shown to work well for moderate Reynolds numbers [53]. This approach is used in [54,55]. However, Vanka iteration counts have been observed to increase significantly for high Reynolds numbers [56].

Alternative to a monolithic scheme are methods based on block factorizations. Block Gaussian elimination sets up an iterative scheme which requires solution of sub-linear systems involving the velocity block $A$ and the Schur complement $S = D - CA^{-1}B$, often written as $S = D - BA^{-1}B^T$ for Navier–Stokes (for stable discretizations $D = 0$). The Schur complement is dense because of the presence of $A^{-1}$, and so generally $S$ is never explicitly constructed and preconditioning techniques are built from an approximate representation. For the Stokes equations, $S$ is spectrally equivalent to the viscosity-weighted pressure mass matrix [57]. However, use of the pressure mass matrix as the preconditioner leads to rapidly deteriorating convergence as the Reynolds number increases [58,59]. An alternative Schur complement preconditioner is pressure convection diffusion (PCD) [60]. PCD has been shown to have iteration counts independent of mesh size and mildly dependent on Reynolds number [61]; a drawback to the method is that it is not algebraic and requires application of a convection–diffusion operator to the pressure space which is a non-trivial implementation detail. On the other hand the least-squares commutator (LSC), described in [62,63], is algebraic. Originally applicable only to stable discretizations, the method was extended to handle stabilized discretizations ($D \neq 0$) in [64]. In [61] LSC is stated to show some iteration

count dependence on mesh size for general flows as well as $\mathrm{Re}^{-1}$ dependence. They suggest a modification to the commutator that leads to mesh independent scaling for a wider range of Reynolds numbers. While most literature regarding sparse approximate commutators (PCD, LSC) is tied to finite elements, Delcourte [65] studied application of LSC to a finite volume discretization. In that work they examined iteration counts as a function of different scaling choices in the LSC, finding that not scaling or scaling by the reciprocal of the diagonal of $A$ was much more effective at high Reynolds numbers than the seemingly optimal choice for finite elements, which is scaling by the reciprocal of the diagonal of the velocity mass matrix. With the optimal scaling choice, they found iteration counts with mesh dependence between $h^{-1}$ and $h^{-1/2}$ and near $\mathrm{Re}^{-1/2}$.

A third class of solution methodologies is based on an Augmented Lagrange approach, introduced by Benzi and Olshanskii in [56]. The technique introduces a grad-div stabilization which does not change the continuous solution but does modify the Schur complement. For an appreciable grad-div stabilization, the Schur complement is well approximated by the pressure mass matrix. The Augmented Lagrange algorithm shows near independence of the Reynolds number. Originally limited to two dimensions, this method was extended to three dimensions by Farrell [66]. Recent work by Shih [67] has further improved the methodology by extending the supported element types from triangles/tetrahedron to quadrilaterals/hexahedron. Not a member of the three aforementioned preconditioning methodologies, we briefly note recent work by Chen [68] showing very low iteration counts at high Reynolds numbers.

The majority of `navier_stokes` module users are interested in advection-dominated, high Reynolds number applications. Additionally, an ideal preconditioner would be applicable to both finite element and finite volume discretizations, given the availability and use of both discretization options in the module. Finally, the preconditioning strategy chosen should accommodate arbitrary force terms in the momentum equations, including velocity dependent terms such as Darcy/Forchheimer friction. Given these criteria, we have focused to this point on the algebraic LSC preconditioner. LSC has been available in PETSc since 2009, but we have recently added missing scaling operations necessary for more performant mesh scaling as well as the option to commute terms suggested in [61]. The latter capability is useful for finite elements but a poor option for finite volumes. Preconditioning of the $CT^{-1}B$ operator, where $T$ may be the diagonal of the velocity mass matrix, diagonal of $A$, or the identity matrix, is performed using hypre's algebraic multigrid implementation, `boomeramg` [4]. For low Reynolds numbers, `boomeramg` is also used to precondition $A$. For high Reynolds numbers, we currently do a nested symmetric multiplicative split of the velocity components and perform LU factorization on the sub-blocks, which is not scalable but still much more performant than a monolithic LU factorization. Future work will explore choosing good smoothers such that `boomeramg` may still be applied for high Reynolds numbers. Another area of investigation will be approximate ideal restriction (AIR) [69], developed specifically for nonsymmetric multigrid, hooks for which the authors have recently added to PETSc. Usefulness for recirculating flows, however, remains to be seen. We note that additional physics may be added to MOOSE Navier–Stokes inputs and efficiently solved by nesting the velocity–pressure Schur split inside of a larger field split. As an example, conjugate heat transfer may be the second physics solved in a multiplicative (Gauss–Seidel) split with Navier–Stokes solved first.

#### 2.2.5. Other functionality

MOOSE `navier_stokes` supports hybrid memory parallelization, including both distributed memory via the message-passing interface (MPI) as well as shared memory using various thread models including `OpenMP`, `TBB`, and `pthreads`. This ability to run with different memory paradigms allows users to maximize the computational resources available to them. Another important feature of `navier_stokes` is the ability to perform adaptive mesh refinement for both finite volume and continuous finite element implementations. This allows users to concentrate computational expense in regions with steep gradients while leaving regions of smooth solutions relatively coarse. The module also leverages the MooseDocs system, which allows developers to document module objects using markdown with various extensions such as KaTeX for equation rendering and Bibtex for citations. The `MooseDocs` system facilitates the `navier_stokes` module's adherence to MOOSE's NQA-1 standard for software quality assurance [70].

#### 2.2.6. Future research and development

The authors are currently researching the hybrid reconstructed Discontinuous Galerkin continuous Galerkin strategy presented in [71]. This work is of particular interest to users wishing to model multi-phase flow due to its intrinsic LBB stability. Other areas of interest include space–time integration in the vein of [54,55] as well as hydridizable discontinuous Galerkin schemes [72,73] which are more efficient than the classic DG schemes.

## 3. Illustrative examples

### 3.1. 2D flow over cylinder

To start, we present steady two-dimensional flow over a cylindrical obstruction, which is one of the benchmark problems presented in [74]. Results for velocity magnitude and pressure are shown in 1 using a finite volume discretization with 166,266 unknowns. The computed drag and lift coefficients are 5.5742 and .0107 respectively, both within the bounds of the benchmark. On a coarser mesh with 75,138 unknowns, the drag and lift coefficients are 5.5667 and .0107 respectively.

### 3.2. 3D molten salt reactor

The capabilities of the module are showcased using the model of a simplified molten salt reactor. The steady-state simulation was carried out using the parameters presented in Table 3, while the geometry of the reactor is shown in Fig. 2. The modeled system is a closed loop that circulates molten chloride with fissile material dissolved in it. The pump is modeled as a volumetric force uniformly distributed in the blue region. Similarly, the heat exchanger is modeled as a volumetric heat sink in the same region. Through the easy-to-use multiphysics capabilities, the thermal hydraulics model in this paper was coupled with Griffin [8], a MOOSE-based neutron transport code for the determination of the heat source density.

The fluid flow in the reactor is modeled as an incompressible fluid with the addition of the Boussinesq approximation for the treatment of buoyancy. These assumptions can be justified by the low power density and consequently, minimal temperature change in the reactor. Fig. 2 also shows the velocity field using stream lines. We see a recirculation region close to the inlet (bottom) of the reactor core.

The temperature and delayed neutron precursor (group 5) fields are presented in Fig. 3. We see an asymmetric temperature distribution in the core which is the result of the secondary flow (recirculation region) close to the reactor inlet. This layering,

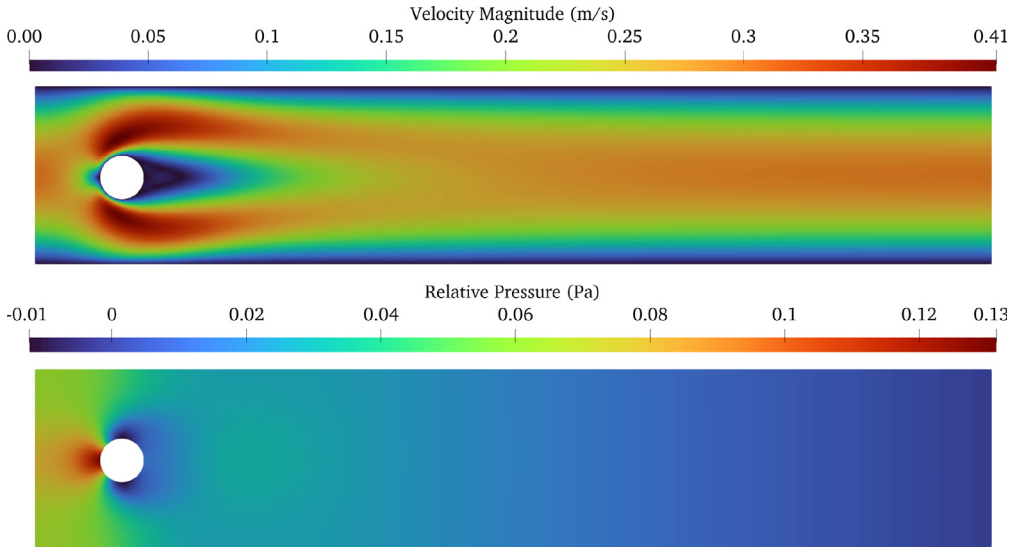


**Fig. 1.** Results for two-dimensional steady flow over a cylindrical obstruction with a Reynolds number of 20. The top figure shows the velocity magnitude and the bottom figure shows the pressure.

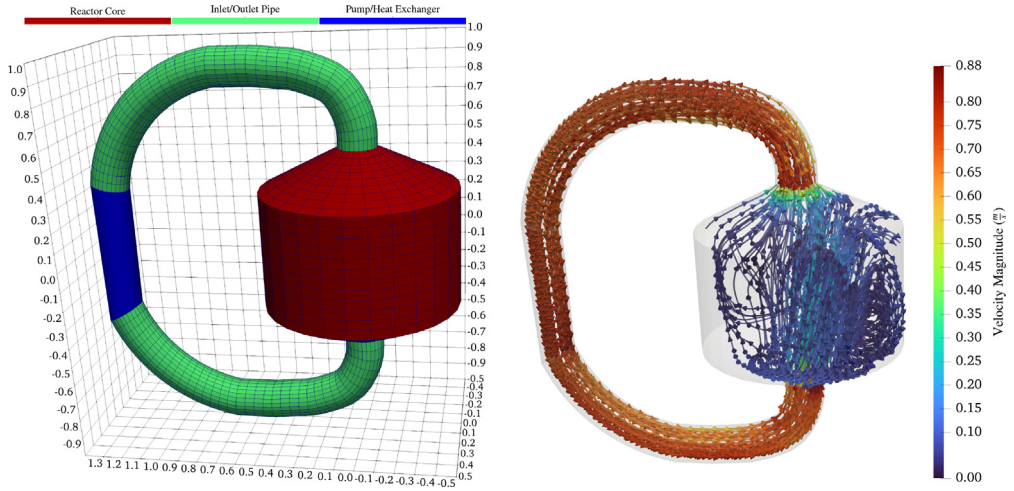


**Fig. 2.** The geometry (left) of the molten chloride reactor together with the velocity field (right) obtained by a steady-state simulation.

**Table 3**
The parameters used for the simulation of the molten salt reactor.

| Parameter | Value |
|---|---|
| Density | 1550 $\left[\frac{\text{kg}}{\text{m}^3}\right]$ |
| Dynamic viscosity | 0.0166 [Pa s] |
| Thermal expansion coefficient | $5.64 \times 10^{-5}$ $\left[\frac{1}{\text{K}}\right]$ |
| Volumetric pump force | 25 $\left[\frac{\text{kN}}{\text{m}^3}\right]$ |
| Specific heat | 1594 $\left[\frac{\text{J}}{\text{kg K}}\right]$ |
| Thermal conductivity | 1.7 $\left[\frac{\text{W}}{\text{m K}}\right]$ |
| External temperature in heat exchanger | 873.15 K |
| Volumetric heat exchange coefficient | 50 $\frac{\text{MW}}{\text{m}^3}$ |
| Reactor power | 1 $MW_{\text{th}}$ |

however, is not present in the precursor concentration due to the additional sink and source terms coming from the radioactive decay of the delayed neutron precursors (sink) and the fission of the nuclear fuel (source).

## 4. Impact

Because `navier_stokes` derives from the MOOSE framework, its physics can be easily integrated into MOOSE multiphysics simulations. For instance `navier_stokes` is coupled with the `tensor_mechanics` module in the `fsi` module to model transmission of stresses and material displacement across fluid–structure boundaries. Additionally, the Virtual Test Bed (VTB) [75,76] contains a litany of nuclear reactor models which couple neutronics and thermal hydraulics (powered by the `navier_stokes` module) through the MOOSE `MultiApp` system. Indeed, all of the next generation of nuclear reactor types will depend on an ability to understand the interactions of neutronics with coolant fluid dynamics; consequently, the ability to perform fluid dynamics simulations within a multiphysics setting is critical.

Beyond its essential role in modeling advanced carbon-free nuclear systems, the `navier_stokes` module is an excellent playground for learning about different discretization techniques for computation fluid dynamics. To the authors' knowledge, no other open-source software offers both finite element and finite

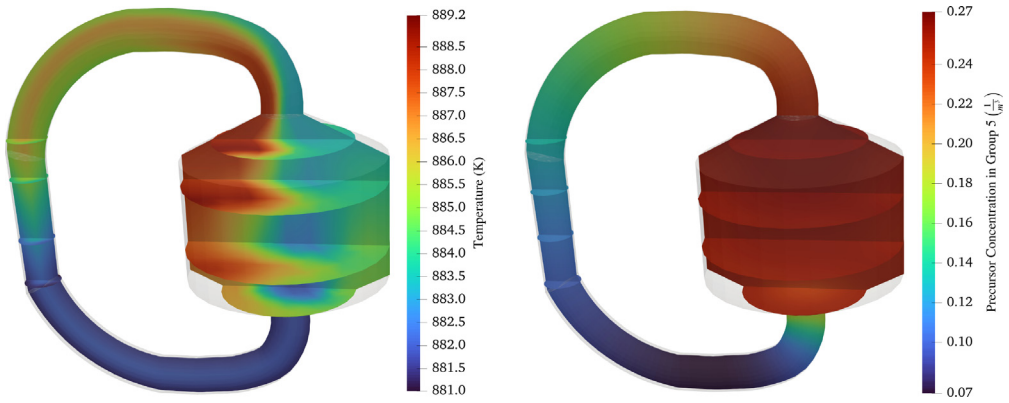


**Fig. 3.** The steady-state temperature (left) and precursors concentration (right) fields in the molten chloride reactor.

volume discretizations of the Navier–Stokes equations. Having the two discretizations side-by-side allows users to rapidly test and learn what strategies work best for different kinds of flow conditions. Additionally any algorithmic improvement made in one discretization technique can often inform algorithmic improvements in the other.

## 5. Conclusions

The `navier_stokes` module plays a critical role in modeling advanced nuclear energy reactor concepts. Through its dependence on the MOOSE framework, libMesh, and PETSc, the module has access to features such as mesh adaptivity, mesh distribution, MPI and thread parallelism, and state-of-the-art preconditioners and linear solvers. The module offers finite element and finite volume discretizations, free-flow and porous implementations of the Navier–Stokes equations, pressure and high-Reynolds number stabilization techniques, and various turbulence models. Because of these features, its ease of incorporation in multiphysics models, and `MooseDocs` documentation, use of the module is expanding, not just within the nuclear community, but in the broader computational fluid dynamics community as well.

## Declaration of competing interest

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

## Data availability

Data will be made available on request

## Acknowledgments

We acknowledge contributions from Sterling Harper, David Andrs, Nolan Macdonald, Andrew Slaughter, Daniel Schwen, Adam Cahill, Gavin Ridley, Joshua Hansel, Wen Jiang, Fande Kong, Albert Patrick, Logan Harbour, Brian Alger, Yidong Xia, Zach Prince, Nathaniel Peat, Jason Hales, Casey Icenhour, April Novak, Topher Matthews, Yeon Sang Jung, Roy Stogner, and Martin Lesueur. We also appreciate conversations with Pierre Jolivet on approaches to preconditioning the linearized Navier–Stokes equations. We also wish to acknowledge the United States Department of Energy Nuclear Energy Advanced Modeling and Simulation (NEAMS) program, which has been the principal funding source of module development over the years. Battelle Energy Alliance, LLC operates Idaho National Laboratory under contract number DEAC0705ID14517 from the U.S. Department of Energy. UChicago Argonne, LLC operates Argonne National Laboratory under U.S. Department of Energy contract number DEAC0206CH 11357.

## References

[1] Lindsay AD, Gaston DR, Permann CJ, Miller JM, Andrš D, Slaughter AE, et al. 2.0 - MOOSE: Enabling massively parallel multiphysics simulation. SoftwareX 2022;20:101202. http://dx.doi.org/10.1016/j.softx.2022.101202, URL https://www.sciencedirect.com/science/article/pii/S2352711022001200.

[2] Kirk BS, Peterson JW, Stogner RH, Carey GF. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. Eng Comput 2006;22:237–54.

[3] Balay S, Abhyankar S, Adams MF, Benson S, Brown J, Brune P, et al. PETSc/TAO users manual. Tech. rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory; 2022.

[4] Falgout RD, Yang UM. hypre: A library of high performance preconditioners. In: International conference on computational science. Springer; 2002, p. 632–41.

[5] Lindsay A, Stogner R, Gaston D, Schwen D, Matthews C, Jiang W, et al. Automatic differentiation in MetaPhysicL and its applications in MOOSE. Nucl Technol 2021;1–18.

[6] Gaston DR, Permann CJ, Peterson JW, Slaughter AE, Andrš D, Wang Y, et al. Physics-based multiscale coupling for full core nuclear reactor simulation. Ann Nucl Energy 2015;84:45–54.

[7] Abou-Jaoude A, Harper S, Giudicelli G, Balestra P, Schunert S, Martin N, et al. A workflow leveraging MOOSE transient multiphysics simulations to evaluate the impact of thermophysical property uncertainties on molten-salt reactors. Ann Nucl Energy 2021;163:108546.

[8] Wang Y, Schunert S, Ortensi J, Laboure V, DeHart M, Prince Z, et al. Rattlesnake: A MOOSE-based multiphysics multischeme radiation transport application. Nucl Technol 2021;207(7):1047–72.

[9] Egert R, Nitschke F, Gholami Korzani M, Kohl T. Stochastic 3D Navier-Stokes flow in self-affine fracture geometries controlled by anisotropy and channeling. Geophys Res Lett 2021;48(9):e2020GL092138.

[10] Yang Y, Do nate-Buendía C, Oyedeji TD, Gökce B, Xu B-X. Nanoparticle tracing during laser powder bed fusion of oxide dispersion strengthened steels. Materials 2021;14(13):3463.

[11] Dhulipala SL, Shields MD, Spencer BW, Bolisetti C, Slaughter AE, Labouré VM, et al. Active learning with multifidelity modeling for efficient rare event simulation. J Comput Phys 2022;468:111506.

[12] Taddei T, Zhang L. A discretize-then-map approach for the treatment of parameterized geometries in model order reduction. Comput Methods Appl Mech Engrg 2021;384:113956.

[13] Yang Y, Kühn P, Yi M, Egger H, Xu B-X. Non-isothermal phase-field modeling of heat–melt–microstructure-coupled processes during powder bed fusion. JOM 2020;72(4):1719–33.

[14] Chen X, Williams DM. Versatile mixed methods for the incompressible Navier–Stokes equations. Comput Math Appl 2020;80(6):1555–77.

[15] Blöcher G, Kluge C, Milsch H, Cacace M, Jacquey AB, Schmittbuhl J. Permeability of matrix-fracture systems under mechanical loading–constraints from laboratory experiments and 3-D numerical modelling. Adv Geosci 2019;49:95–104.

[16] Kadapa C, Dettmer WG, Perić D. Accurate iteration-free mixed-stabilised formulation for laminar incompressible Navier–Stokes: Applications to fluid–structure interaction. J Fluids Struct 2020;97:103077.

[17] Lesueur M, Rattez H, Colomés O. $\mu$CT scans permeability computation with an unfitted boundary method to improve coarsening accuracy. Comput Geosci 2022;105118.

[18] von Planta C, Vogler D, Chen X, Nestola MG, Saar MO, Krause R. Modelling of hydro-mechanical processes in heterogeneous fracture intersections using a fictitious domain method with variational transfer operators. Comput Geosci 2020;24(5):1799–814.
[19] Gao P. Three dimensional finite element computation of the non-isothermal polymer filling process by the phase field model. Adv Eng Softw 2022;172:103207.
[20] Gałek R, Strzelczyk P. Velocity profiles of an electrohydrodynamic flow generator: CFD and experiment. J Electrost 2019;99:19–30.
[21] He Y, Niu Y, Xiang F, Wu Y, Zhang J, Su G, et al. Preliminary development of a multi-physics coupled fuel performance code for annular fuel analysis under normal conditions. Nucl Eng Des 2022;393:111810.
[22] Park SM, Munk M. Verification of moltres for multiphysics simulations of fast-spectrum molten salt reactors. Ann Nucl Energy 2022;173:109111.
[23] Stewart R, Balestra P, Reger D, Merzari E. Generation of localized reactor point kinetics parameters using coupled neutronic and thermal fluid models for pebble-bed reactor transient analysis. Ann Nucl Energy 2022;174:109143.
[24] Nicolas A, Messner MC, Sham T-L. A method for predicting failure statistics for steady state elevated temperature structural components. Int J Press Vessels Pip 2021;192:104363.
[25] Deng Q, Blöcher G, Cacace M, Schmittbuhl J. Hydraulic diffusivity of a partially open rough fracture. Rock Mech Rock Eng 2021;54(10):5493–515.
[26] Weiss AG, Abdoelatef MG, Bani Ahmad MT, Ahmed K, Kimber ML. A preliminary evaluation of the computational fluid dynamics capabilities in MOOSE. In: International conference on nuclear engineering, vol. 85260. American Society of Mechanical Engineers; 2021, V003T08A016.
[27] Xie Z, Jiang W, Wang C, Wu X. Bayesian inverse uncertainty quantification of a MOOSE-based melt pool model for additive manufacturing using experimental data. Ann Nucl Energy 2022;165:108782.
[28] Nicolas A, Messner MC, Sham T-L. Initial development of a high temperature life prediction method directly accounting for variability in material properties. Tech. rep., Argonne, IL (United States): Argonne National Lab.(ANL); 2020.
[29] Gaston DR. Parallel, asynchronous ray-tracing for scalable, 3D [Ph.D. thesis], Massachusetts Institute of Technology; 2020, full-core method of characteristics neutron transport on unstructured mesh.
[30] Weiss AG, Zaidan LJ, Bani Ahmad MT, Abdoelatef MG, Peterson JW, Lindsay AD, et al. Characterization of the finite element computational fluid dynamics capabilities in the multiphysics object oriented simulation environment. J Nucl Eng Radiat Sci 2022;9(2):021402.
[31] Kong F. Neutron transport criticality calculations using a parallel monolithic multilevel Schwarz preconditioner together with a nonlinear diffusion acceleration method. Ann Nucl Energy 2020;141:107342.
[32] Kong F, Wang Y, Gaston DR, Lindsay AD, Permann CJ, Martineau RC. A scalable multilevel domain decomposition preconditioner with a subspace-based coarsening algorithm for the neutron transport calculations. 2019, arXiv preprint arXiv:1906.07743.
[33] Hu R, Fang J, Nunez D, Tano M, Giudicelli G, Salko R. Development of integrated thermal fluids modeling capability for MSRs. Tech. rep., Argonne, IL (United States): Argonne National Lab.(ANL); 2022.
[34] Guo J, Zhang S, Yang C, Wang J, Huang S, Wang K. Preliminary verification of incompressible Navier-Stokes equations solved by the Newton method. Int J Adv Nucl React Des Technol 2020;2:69–85.
[35] Gałek R, Wilk J. Numerical simulation of air flow in needle-to-cylinder electrohydrodynamic device. Prog Comput Fluid Dyn Int J 2020;20(2):84–92.
[36] Brooks A, Tjr H. Streamline-upwind/petrov-galerkin methods for advection dominated flows. 1980.
[37] Rhie CM, Chow W-L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J 1983;21(11):1525–32.
[38] Toro EF. Riemann solvers and numerical methods for fluid dynamics: A practical introduction. 3rd ed. Springer Science & Business Media; 2009.
[39] Kurganov A, Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. J Comput Phys 2000;160(1):241–82.
[40] Kurganov A, Noelle S, Petrova G. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations. SIAM J Sci Comput 2001;23(3):707–40.
[41] Mohammadi B, Pironneau O. Analysis of the k-epsilon turbulence model. 1993.
[42] Moukalled F, Mangani L, Darwish M. The finite volume method. In: The finite volume method in computational fluid dynamics. Springer; 2016, p. 103–35.
[43] Van Leer B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. J Comput Phys 1977;23(3):276–99.
[44] Sweby PK. High resolution TVD schemes using flux limiters. In: Large-scale computations in fluid mechanics. 1985.
[45] Brezzi F. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. Publ Math Inf Rennes 1974;(S4):1–26.
[46] Tezduyar TE, Mittal S, Ray S, Shih R. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comput Methods Appl Mech Engrg 1992;95(2):221–42.
[47] Courant R, Friedrichs K, Lewy H. Über die partiellen Differenzengleichungen der mathematischen physik. Math Ann 1928;100(1):32–74.
[48] De Moura CA, Kubrusly CS. The Courant–Friedrichs–Lewy (CFL) condition. AMC 2013;10(12).
[49] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620–31.
[50] Fiorina C, Clifford I, Aufiero M, Mikityuk K. GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors. Nucl Eng Des 2015;294:24–37.
[51] Vanka SP. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. J Comput Phys 1986;65(1):138–58.
[52] Braess D, Sarazin R. An efficient smoother for the Stokes problem. Appl Numer Math 1997;23(1):3–19.
[53] Turek S. Efficient solvers for incompressible flow problems: an algorithmic and computational approache, vol. 6. Springer Science & Business Media; 1999.
[54] Anselmann M, Bause M. A geometric multigrid method for space-time finite element discretizations of the Navier–Stokes equations and its application to 3d flow simulation. ACM Trans Math Software 2023;49(1):1–25.
[55] Hussain S, Schieweck F, Turek S. An efficient and stable finite element solver of higher order in space and time for nonstationary incompressible flow. Internat J Numer Methods Fluids 2013;73(11):927–52.
[56] Benzi M, Olshanskii MA. An augmented Lagrangian-based approach to the Oseen problem. SIAM J Sci Comput 2006;28(6):2095–113.
[57] Silvester D, Wathen A. Fast iterative solution of stabilised Stokes systems part II: Using general block preconditioners. SIAM J Numer Anal 1994;31(5):1352–67.
[58] Elman H, Silvester D. Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations. SIAM J Sci Comput 1996;17(1):33–46.
[59] Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press; 2014.
[60] Kay D, Loghin D, Wathen A. A preconditioner for the steady-state Navier–Stokes equations. SIAM J Sci Comput 2002;24(1):237–56.
[61] Olshanskii MA, Vassilevski YV. Pressure Schur complement preconditioners for the discrete Oseen problem. SIAM J Sci Comput 2007;29(6):2686–704.
[62] Elman HC. Preconditioning for the steady-state Navier–Stokes equations with low viscosity. SIAM J Sci Comput 1999;20(4):1299–316.
[63] Elman H, Howle VE, Shadid J, Shuttleworth R, Tuminaro R. Block preconditioners based on approximate commutators. SIAM J Sci Comput 2006;27(5):1651–68.
[64] Elman H, Howle VE, Shadid J, Silvester D, Tuminaro R. Least squares preconditioners for stabilized discretizations of the Navier–Stokes equations. SIAM J Sci Comput 2008;30(1):290–311.
[65] Delcourte S, Jennequin D. Saddle point preconditioners for linearized Navier–Stokes equations discretized by a finite volume method. Appl Numer Math 2010;60(11):1054–66.
[66] Farrell PE, Mitchell L, Wechsung F. An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier–Stokes equations at high Reynolds number. SIAM J Sci Comput 2019;41(5):A3073–96.
[67] Shih Y-h, Stadler G, Wechsung F. Robust multigrid techniques for augmented Lagrangian preconditioning of incompressible Stokes equations with extreme viscosity variations. SIAM J Sci Comput 2022;S27–53.
[68] Chen Q, Jiao X, Yang O. Robust and efficient multilevel-ILU preconditioning of hybrid Newton–GMRES for incompressible Navier–Stokes equations. Internat J Numer Methods Fluids 2021;93(12):3405–23.
[69] Manteuffel TA, Ruge J, Southworth BS. Nonsymmetric algebraic multigrid based on local approximate ideal restriction (ℓAIR). SIAM J Sci Comput 2018;40(6):A4105–30.
[70] Slaughter AE, Permann CJ, Miller JM, Alger BK, Novascone SR. Continuous integration, in-code documentation, and automation for nuclear quality assurance conformance. Nucl Technol 2021;207(7):923–30.

[71] Pandare AK, Luo H. A hybrid reconstructed discontinuous Galerkin and continuous Galerkin finite element method for incompressible flows on unstructured grids. J Comput Phys 2016;322:491–510.
[72] Cockburn B. The hybridizable discontinuous Galerkin methods. In: Proceedings of the international congress of mathematicians 2010 (ICM 2010) (in 4 volumes) vol. I: Plenary lectures and ceremonies vols. II–IV: invited lectures. World Scientific; 2010, p. 2749–75.
[73] Lehrenfeld C. Hybrid discontinuous Galerkin methods for solving incompressible flow problems. Rheinisch Westfalischen Tech Hochschule Aachen 2010;111.
[74] Schäfer M, Turek S, Durst F, Krause E, Rannacher R. Benchmark computations of laminar flow around a cylinder. Springer; 1996.
[75] Abou-Jaoude A, Gaston D, Giudicelli G, Feng B, Permann C. The virtual test bed repository : A library of multiphysics reference reactor models using NEAMS tools. In: Transactions of the American nuclear society. 2021.
[76] Giudicelli GL, Abou-Jaoude A, Novak AJ, Abdelhameed A, Balestra P, Charlot L, Fang J, Feng B, Folk T, Freile R, et al. The virtual test bed (VTB) repository: a library of reference reactor models using NEAMS tools. Nuclear Science and Engineering 2023;1–17.