Contents lists available at ScienceDirect

Journal of Computational Physics

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

On physics-based preconditioning of the Navier–Stokes equations

HyeongKae Park *, Robert R. Nourgaliev, Richard C. Martineau, Dana A. Knoll

Idaho National Laboratory, Multiphysics Methods Group, 2525 North Fremont Avenue, Idaho Falls, ID 83415-3860, United States

ARTICLE INFO

Article history:
Received 9 February 2009
Received in revised form 11 August 2009
Accepted 14 September 2009
Available online 18 September 2009

Keywords:
Jacobian-free Newton–Krylov
Navier–Stokes equations
Physics-based preconditioning

ABSTRACT

We develop a fully implicit scheme for the Navier–Stokes equations, in conservative form, for low to intermediate Mach number flows. Simulations in this range of flow regime produce stiff wave systems in which slow dynamical (advective) modes coexist with fast acoustic modes. Viscous and thermal diffusion effects in refined boundary layers can also produce stiffness. Implicit schemes allow one to step over the fast wave phenomena (or unresolved viscous time scales), while resolving advective time scales. In this study we employ the Jacobian-free Newton–Krylov (JFNK) method and develop a new physics-based preconditioner. To aid in overcoming numerical stiffness caused by the disparity between acoustic and advective modes, the governing equations are transformed into the primitive-variable form in a preconditioning step. The physics-based preconditioning incorporates traditional semi-implicit and physics-based splitting approaches without a loss of consistency between the original and preconditioned systems. The resulting algorithm is capable of solving low-speed natural circulation problems ($M \sim 10^{-4}$) with significant heat flux as well as intermediate speed ($M \sim 1$) flows efficiently by following dynamical (advective) time scales of the problem.

1. Introduction

There is an increasing interest in developing next generation high-fidelity simulation tools for analysis of nuclear reactor systems. These tools should enhance the predictive capability and will be used for design, optimization and safety assessments of future advanced nuclear reactors. Operating conditions for new reactor concepts can differ significantly from the conventional Light Water Reactors (LWR). For example, high temperature gas-cooled reactors (HTGR) are expected to operate at higher thermal loadings (e.g. a temperature difference of the coolant between inlet and outlet can be as high as ~400 K). Under high-heat flux conditions, density variations of the coolant may become significant even though the flow is relatively low-speed. Under these large entropy changing conditions, incompressible flow simulations with the first-order Boussinesq approximation for buoyancy may cease to be adequate. A deficiency of the incompressible/Boussinesq approximation was demonstrated by the work of Darbandi and Hosseinizadeh [6], where greater than 10% difference in the local Nusselt number has been observed. Therefore, development of more appropriate numerical methods based on the conservative (compressible) form of the Navier–Stokes equations may be necessary to avoid significant modeling errors.

Another motivation for using the conservative form of the Navier–Stokes equations is the ability to model a wide range of flow conditions. For example, two flow regimes are important during hypothetical HTGR accidents: the loss of the forced cooling accident (LOFC) *with* and *without* depressurization [30]. The LOFC without depressurization occurs when a coolant pump fails, leading to a natural convection scenario (passive cooling). In this regime, the flow is driven by the buoyancy forces due to density variations. On the other hand, the LOFC with depressurization occurs when a coolant pipe breaks,

[*] Corresponding author.
E-mail address: Ryosuke.Park@inl.gov (H. Park).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.09.015

causing choked flow at the break due to the large pressure difference. In order to analyze both of these accident scenarios within one framework, an efficient flow algorithm which is valid in the range of very low-Mach number to Mach number approximately unity is required.

One of the challenges in developing algorithms in this flow regime is a numerical stiffness under low-Mach number conditions. With the conservative form of the Navier–Stokes equations, we take into consideration the thermodynamic pressure through an equation of state. This accounts for acoustic modes (as opposed to filtering them out, in the incompressible formulation) and, therefore, introduces a significant time scale separation between fast (acoustic) and slow dynamic (convective) modes. Moreover, time scales of heat conduction and viscous dissipations may be also stiff, especially when one attempts to resolve boundary layers or introduces eddy-diffusivity based turbulence closures.

Several numerical algorithms have been proposed for solving low-Mach number flows. One of the most often used is the "low-Mach number asymptotic" expansion [22,15]. It extends compressible flow algorithms to low-Mach number flows by expanding the flow variables in terms of the Mach number, and rewriting the pressure in terms of spatially constant thermodynamic pressure and spatially-varying local hydrodynamic pressure. For example, a recent work by Liou [19] extends the AUSM$^+$-up scheme to all-speed flows using the low-Mach number asymptotes. Another example is the implicit (stabilized) finite element method developed by Liu and Makhviladze [20]. The large heat release can be treated by these asymptotic expansions because they do not assume solenoidal velocity field $\nabla \cdot \vec{u} = 0$. However, the solution may become inaccurate at intermediate Mach numbers due to the expansion. Therefore, the low-Mach number asymptotic expansion method is often used in low-speed combustion problems.

Semi-implicit methods are another widely used class of all-speed solvers. For example, the Semi-Implicit Method for Pressure Linked Equation (SIMPLE) [25] and low-Mach number projection algorithms [5] solve a velocity–pressure coupling implicitly by efficient iterative schemes. Another example is the Implicit Continuous Eulerian (ICE) method [13], which treats the pressure term in the momentum equation implicitly, eliminating time step restrictions due to fast acoustic modes. The original ICE scheme has been developed primarily for isentropic flows. Thus, it is not well suited for flows with high-heat fluxes. Several non-isentropic improvements have been introduced, including a recently proposed PCICE algorithm [1,21] which minimizes this restriction by incorporating viscous dissipations and heat conduction in a predictor–corrector fashion. However, the semi-implicit nature of the PCICE algorithm can produce additional time integration errors which may become significant at a large CFL number.

Many researchers have attempted to expand the compressible flow framework into an incompressible limit by using "time derivative preconditioning" [3,33,34]. The preconditioned system changes the acoustic speed close to the advection speed. However, similar to artificial compressibility methods [4], it is difficult to achieve time-accurate solutions efficiently; therefore, it is mostly used for steady state computations.

The current approach differs in the following way. First, we solve the conservative form of the Navier–Stokes equations implicitly without any simplifications. Because our discretization does not assume low-Mach number flows, we can solve intermediate Mach number flows without loss of accuracy. The resulting nonlinear system is solved efficiently by the Jacobian-free Newton–Krylov (JFNK) method with physics-based preconditioning (PBP) [16,17,28]. The PBP is employed to accelerate the convergence of the Krylov-based iterative method. In the PBP, the basic ideas of above-mentioned classical semi-implicit, and/or operator-split methods are utilized. The numerical stiffness due to the disparity of the time scales is remedied by identifying and isolating the distinct physical phenomena (i.e. acoustic waves, heat conduction, and viscosity), then preconditioning each physics implicitly.

The current work is closely related to two previous efforts. One of these efforts used the JFNK–PBP approach for application to compressible multiphase flow for hurricane simulation [29,28]. The other effort used the JFNK–PBP approach for application to incompressible flow plus phase-change for melting–freezing applications [10,11]. In both cases, using the semi-implicit method and operator splitting as a preconditioner was shown to provide significant benefit to using the semi-implicit method alone as the solver. In this work we extend the definition of the preconditioning concept to a conservative formulation of the Navier–Stokes equations and establish the algorithm performance on a more generic problem.

The key feature of JFNK is combining Newton's method to solve implicit nonlinear systems with Krylov subspace iterative methods. Krylov methods do not require an explicit form of the Jacobian, which eliminates the computationally expensive step of forming analytical Jacobian matrices required by Newton's method. The required matrix–vector product can be approximated by the numerical differentiation of nonlinear residual functions; therefore, JFNK readily integrates the different physics into one solver framework.

In this study we demonstrate how to efficiently solve the conservative (compressible) form of the Navier–Stokes equations in a wide range of Mach numbers (from 10^{-4} to 1) with second-order spatiotemporal discretization. There are two key ideas beyond JFNK that allow for this. First, we transform the original conservative-variable system into the primitive-variable formulation at the preconditioning stage. This transformation enables separation of different physical phenomena, and allows one to utilize traditional operator-split and semi-implicit algorithms as an effective (physics-based) preconditioner. Second, we have developed a compact form of discrete pressure equation. For this, we employ the Rhie–Chow based interpolation [27,31] to compute fluxing velocities at cell edges. The consistent discretization between the original and the preconditioned systems, together with compact stencils, maximizes the efficiency of the algorithm.

The reminder of this paper is organized as follows. Section 2 introduces the governing equations and the JFNK framework. The spatiotemporal discretization of the governing equation is described in Section 3. We focus here on the Rhie–Chow based interpolation of the advection operator. In Section 4, we describe our physics-based preconditioner for all-speed flows in detail. Efficiency and accuracy of the developed algorithm is demonstrated in Section 5 using two examples of widely varying Mach number. Finally, in Section 6 we conclude this work and outline several future directions.

2. Overview of the solution strategy

2.1. Governing equations

The governing hydrodynamic equations, in the conservative form, are

Conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} = 0$$

The Navier–Stokes equations

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) + \nabla P - \nabla \cdot \underline{\tau} - \rho \vec{g} = 0 \tag{1}$$

Conservation of total energy

$$\frac{\partial E}{\partial t} + \nabla \cdot (\vec{u}(E + P)) - \nabla \cdot (\underline{\tau} \cdot \vec{u}) - \nabla \cdot \vec{q} - \rho \vec{g} \cdot \vec{u} = 0,$$

where $\rho,\ \rho\vec{u},\ E$ are the density, momentum, and total energy of the fluid, $P,\ \underline{\tau},\ \vec{g}$, and $\vec{q} = \kappa \nabla T$ are the pressure, viscous stress tensor, gravity vector, and heat flux, κ and T are the thermal conductivity and temperature, respectively. For a Newtonian fluid with the Stokes hypothesis, the viscous stress tensor components are defined as

$$\underline{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \end{bmatrix} = \begin{bmatrix} \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) & \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \end{bmatrix}. \tag{2}$$

Eq. (1) must be closed with an equation of state. Here, we use the ideal gas law,

$$P = \rho e(\gamma - 1), \tag{3}$$

where $e = c_v T$ and γ are the internal energy and the ratio of specific heats, respectively and c_v is the specific heat at constant volume. Eq. (1) can be written in vector form,

$$\frac{\partial \mathbf{U}}{\partial t} + R(\mathbf{U}) = 0, \tag{4}$$

where $\mathbf{U} = \{\rho, \rho u, \rho v, E\}^T$ is the vector of conservative variables and R is the spatial discrete operators.

2.2. Time discretization

Due to the existence of multiple time scales in low-Mach number flows, we would like to solve Eq. (1) implicitly. An implicit time integration allows one to choose the time step solely based on accuracy, not stability. In the present study, we employ two implicit time discretization schemes, the first-order Backward Euler (BE1),

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + R(\mathbf{U}^{n+1}) = 0, \tag{5}$$

and the second-order Backward Differencing (BDF2),

$$\frac{\frac{3}{2}\mathbf{U}^{n+1} - 2\mathbf{U}^n + \frac{1}{2}\mathbf{U}^{n-1}}{\Delta t} + R(\mathbf{U}^{n+1}) = 0. \tag{6}$$

2.3. Jacobian-free Newton–Krylov method

The Newton–Krylov method is a synergistic combination of the quadratically convergent Newton method and a Krylov subspace iterative method [16]. Newton's method iteratively solves the nonlinear equations of the following form,

$$\text{Res}(\mathbf{U}) = 0, \tag{7}$$

where $\text{Res}(\mathbf{U})$ is the nonlinear residual function. For example, the nonlinear residual function with BE1 can be expressed as

$$\text{Res}(\mathbf{U}^*) = \frac{\mathbf{U}^* - \mathbf{U}^n}{\Delta t} + R(\mathbf{U}^*), \tag{8}$$

where $\mathbf{U}^*$ represents an intermediate solution vector prior to declaration of Newton convergence. At each step of a Newton iteration, the following linear system of equations is solved

$$\mathbb{J}^k \delta \mathbf{U}^k = -\mathrm{Res}(\mathbf{U}^k), \tag{9}$$

where $\mathbb{J}^k \equiv \frac{\partial \mathrm{Res}(\mathbf{U}^k)}{\partial \mathbf{U}^k}$ is the Jacobian matrix of the kth Newton step. Eq. (9) is solved using a Krylov method (e.g. GMRES [32]) and the solution is updated as

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \delta \mathbf{U}^k. \tag{10}$$

JFNK makes use of the fact that Krylov methods involve only a matrix–vector product, hence do not require the explicit form of the Jacobian matrix. The matrix–vector product can then be approximated by the finite difference version of the directional derivative [2] as

$$\mathbb{J}\mathbf{v} \approx \frac{\mathrm{Res}(\mathbf{U} + \epsilon \mathbf{v}) - \mathrm{Res}(\mathbf{U})}{\epsilon}, \tag{11}$$

where ϵ is a small perturbation parameter and $\mathbf{v}$ is a Krylov vector. There are several options for choosing ϵ [16]. In this study, we use the following simple equation suggested in [26] to choose ϵ,

$$\epsilon = \frac{b\sqrt{1 + \|\mathbf{U}\|_2}}{\|\mathbf{v}\|_2}, \tag{12}$$

where b is a small constant related to the square root of machine round-off. Because the matrix–vector product can be approximated by Eq. (11), the nonlinear residual function evaluation is the only necessary component for solving the discrete system equation (9). In practical applications, Eq. (9) is not solved exactly, but in an approximate manner (i.e. inexact Newton method [9]). At each (inexact) Newton step, we solve Eq. (9) with the following convergence criteria

$$\|\mathbb{J}^k \delta \mathbf{U}^k + \mathrm{Res}(\mathbf{U}^k)\|_2 < \eta \|\mathrm{Res}(\mathbf{U}^k)\|_2. \tag{13}$$

The parameter η, is the forcing term. In this study η is set to a constant value of 10^{-2}. For each time (or pseudotime) step, the nonlinear residual is reduced to 10^{-6}. By choosing the ‘‘inexact” Newton iteration method and a constant forcing term, we sacrifice the quadratic convergence rate of a full Newton method. However, the overall computational effort can be reduced because the effort required to solve the linear system becomes smaller.

2.4. Physics-based preconditioner for Navier–Stokes equations

Although the nonlinear system may be solved only with the nonlinear residual evaluation (and a Krylov method), the success of the JFNK method heavily depends upon an efficiency of the GMRES. Preconditioning is commonly employed to improve the convergence of the GMRES. The right-preconditioned system can be expressed as

$$\mathbb{J}^k \mathbb{M}^{-1} (\mathbb{M} \delta \mathbf{U}^k) = -\mathrm{Res}(\mathbf{U}^k), \tag{14}$$

where $\mathbb{M}$ is the preconditioning matrix or the corresponding operations. When solving Eq. (14), the matrix–vector product required on each GMRES iteration can be approximated by

$$\mathbb{J}\mathbb{M}^{-1}\mathbf{v} \approx \frac{\mathrm{Res}(\mathbf{U} + \epsilon \mathbb{M}^{-1}\mathbf{v}) - \mathrm{Res}(\mathbf{U})}{\epsilon}, \tag{15}$$

where $\mathbf{v}$ is a Krylov vector [32]. Physics-based preconditioning (PBP) [16] has been developed with multiphysics problems in mind. A classical operator splitting and semi-implicit algorithm can be utilized as a PBP because a PBP can be viewed as the equivalent procedure to approximating the new time step solution by simpler (and computationally inexpensive) methods. An intelligent choice of a PBP can substantially reduce the number of Krylov iterations, which consequently reduces both the memory requirements and computational effort.

To solve low to intermediate Mach number flow problems with high-heat fluxes, we are interested in the ability to implicitly step over three physical time scales: (a) stiff acoustic waves in low-speed regimes, (b) viscous diffusion, and (c) thermal diffusion in finely gridded boundary layers. The goal of the preconditioning step is to solve these physics implicitly and efficiently by extracting the contribution of these three physics from the original set of equations. These physics are best represented by the primitive variables. Thus, the first step of preconditioning is to transform the conservative variables ($\mathbf{U}$) into primitive variables ($\mathbf{V}$) via the following similarity transformation

$$A^{-1}\mathbb{J}^k A \delta \mathbf{V}^k = -A^{-1}\mathrm{Res}(\mathbf{U}^k), \tag{16}$$

$$\tilde{\mathbb{J}} \delta \mathbf{V}^k = -\widetilde{\mathrm{Res}}(\mathbf{V}^k), \tag{17}$$

where $\mathbf{V} = \{P, u, v, e\}^T$ represents a vector of the primitive variables, while $\tilde{\mathbb{J}} = A^{-1}\mathbb{J}A$ and $\widetilde{\mathrm{Res}}(\mathbf{V}) = A^{-1}\mathrm{Res}(\mathbf{U})$ are the Jacobian matrix and the residual in the primitive variables, respectively. $A = \frac{\partial \mathbf{U}}{\partial \mathbf{V}}$ is the transformation matrix from the conservative to non-conservative variables [14] and the matrix entities can be evaluated from the constitutive relations (e.g. equation of state). This choice of the primitive variables allows us to produce simple elliptic/parabolic systems that target stiff wave, heat conduction, and viscosity.

The transformed Jacobian matrix is simplified by ignoring slow physics, such as advection. The preconditioning matrix is then reduced to a block lower triangular form, with each diagonal block matrix representing a scalar parabolic equation. With the help of the Rhie–Chow interpolation for advection operators, the scalar parabolic systems are readily formed and approximately inverted sequentially with a fixed number of multigrid V-cycles. The approximated correction terms are then transformed back to the conservative variables to complete a preconditioning step. As a result of this preconditioning, we are able to efficiently solve a wide range of flow-speed problems, from extremely low-speed flow (i.e. $M \sim 10^{-4}$) in the nearly incompressible limit to the choked flow conditions (i.e. $M \sim 1$). In the next section, we discuss the detailed description of our spatiotemporal discretization, followed by the physics-based preconditioner.

3. Spatiotemporal discretization and Rhie–Chow interpolation

Discretizing Eq. (1) in two dimensions using finite volume method with the BE1 time integration, results in

$$\frac{\rho_{i,j}^{n+1}-\rho_{i,j}^{n}}{\Delta t}+\frac{\widehat{\rho u}_{i+1/2,j}^{n+1}-\widehat{\rho u}_{i-1/2,j}^{n+1}}{\Delta x}+\frac{\widehat{\rho v}_{i,j+1/2}^{n+1}-\widehat{\rho v}_{i,j-1/2}^{n+1}}{\Delta y}=0, \tag{18}$$

$$\frac{\rho u_{i,j}^{n+1}-\rho u_{i,j}^{n}}{\Delta t}+\frac{(\widehat{\rho u^2+P}-\tau_{xx})_{i+1/2,j}^{n+1}-(\widehat{\rho u^2+P}-\tau_{xx})_{i-1/2,j}^{n+1}}{\Delta x}+\frac{(\widehat{\rho u v}-\tau_{xy})_{i,j+1/2}^{n+1}-(\widehat{\rho u v}-\tau_{xy})_{i,j-1/2}^{n+1}}{\Delta y}=0, \tag{19}$$

$$\frac{\rho v_{i,j}^{n+1}-\rho v_{i,j}^{n}}{\Delta t}+\frac{(\widehat{\rho u v}-\tau_{yx})_{i+1/2,j}^{n+1}-(\widehat{\rho u v}-\tau_{yx})_{i-1/2,j}^{n+1}}{\Delta x}+\frac{(\widehat{\rho v^2+P}-\tau_{yy})_{i,j+1/2}^{n+1}-(\widehat{\rho v^2+P}-\tau_{yy})_{i,j-1/2}^{n+1}}{\Delta y}-\rho_{i,j}^{n+1}g=0, \tag{20}$$

$$\begin{aligned}&\frac{E_{i,j}^{n+1}-E_{i,j}^{n}}{\Delta t}+\frac{\left[u(\widehat{E+P})\right]_{i+1/2,j}^{n+1}-\left[u(\widehat{E+P})\right]_{i-1/2,j}^{n+1}}{\Delta x}+\frac{\left[v(\widehat{E+P})\right]_{i,j+1/2}^{n+1}-\left[v(\widehat{E+P})\right]_{i,j-1/2}^{n+1}}{\Delta y}\\&\quad-\frac{(u\tau_{xx}+v\tau_{xy}+q_x)_{i+1/2,j}^{n+1}-(u\tau_{xx}+v\tau_{xy}+q_x)_{i-1/2,j}^{n+1}}{\Delta x}-\frac{(u\tau_{yx}+v\tau_{yy}+q_y)_{i,j+1/2}^{n+1}-(u\tau_{yx}+v\tau_{yy}+q_y)_{i,j-1/2}^{n+1}}{\Delta y}\\&\quad-\rho v_{i,j}^{n+1}g=0,\end{aligned} \tag{21}$$

where subscripts $i,\ j$ are the spatial mesh indices and n and $n+1$ are the current and new times, respectively. Dependent variables are located at cell centers (i.e. subscripts $i,\ j$). However, the finite volume method requires face centered numerical fluxes, denoted by a hat "$\wedge$". In Eqs. (18)–(21), we have assumed that the gravity vector is aligned with the y-axis. To solve this system, the numerical fluxes at cell edges must be defined. Viscous and thermal diffusion operators are discretized by the second-order central differencing. For the advection operator, we employ Rhie–Chow based interpolation [27,31], discussed next.

3.1. Rhie–Chow based interpolation for the advection operator

As mentioned previously, our preconditioner approximately solves a series of parabolic (or elliptic) scalar linear systems using a fixed number of multigrid V-cycles. Compactness of the discretization for numerical fluxes is important for multigrid efficiency. Hence, our goal is to introduce a discretization scheme that results in the compact pressure stencils under the transformation of the original discretization to the primitive variables. In this section, we present a discretization of the advection operator using the following Euler equations (i.e. subset of Eq. (1))

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}+\frac{\partial \mathbf{G}(\mathbf{U})}{\partial y}=0, \tag{22}$$

where

$$\mathbf{F}(\mathbf{U})=\left\{\begin{matrix}\rho u\\ \rho u^2+P\\ \rho u v\\ u(E+P)\end{matrix}\right\},\quad \mathbf{G}(\mathbf{U})=\left\{\begin{matrix}\rho v\\ \rho u v\\ \rho v^2+P\\ v(E+P)\end{matrix}\right\}$$

are the advection operators. By integrating over the control volume, we obtain the following discretized equations

$$\frac{\rho_{i,j}^{n+1}-\rho_{i,j}^{n}}{\Delta t}+\frac{\widehat{\rho u}_{i+1/2,j}^{n+1}-\widehat{\rho u}_{i-1/2,j}^{n+1}}{\Delta x}+\frac{\widehat{\rho v}_{i,j+1/2}^{n+1}-\widehat{\rho v}_{i,j-1/2}^{n+1}}{\Delta y}=0, \tag{23}$$

$$\frac{\rho u_{i,j}^{n+1}-\rho u_{i,j}^{n}}{\Delta t}+\frac{(\widehat{\rho u^2+P})_{i+1/2,j}^{n+1}-(\widehat{\rho u^2+P})_{i-1/2,j}^{n+1}}{\Delta x}+\frac{\widehat{\rho u v}_{i,j+1/2}^{n+1}-\widehat{\rho u v}_{i,j-1/2}^{n+1}}{\Delta y}=0, \tag{24}$$

$$\frac{\rho v^{n+1}_{i,j} - \rho v^{n}_{i,j}}{\Delta t} + \frac{\widehat{\rho u v}^{n+1}_{i+1/2,j} - \widehat{\rho u v}^{n+1}_{i-1/2,j}}{\Delta x} + \frac{(\rho \widehat{v^2 + P})^{n+1}_{i,j+1/2} - (\rho \widehat{v^2 + P})^{n+1}_{i,j-1/2}}{\Delta y} = 0, \tag{25}$$

$$\frac{E^{n+1}_{i,j} - E^{n}_{i,j}}{\Delta t} + \frac{u(\widehat{E+P})^{n+1}_{i+1/2,j} - u(\widehat{E+P})^{n+1}_{i-1/2,j}}{\Delta x} + \frac{v(\widehat{E+P})^{n+1}_{i,j+1/2} - v(\widehat{E+P})^{n+1}_{i,j-1/2}}{\Delta y} = 0. \tag{26}$$

In order to apply the Rhie–Chow interpolation, we first rewrite Eqs. (23)–(26) as

$$\frac{\rho^{n+1}_{i,j} - \rho^{n}_{i,j}}{\Delta t} + \frac{\tilde{u}^{n+1}_{i+1/2,j}\rho^{n+1}_{i+1/2,j} - \tilde{u}^{n+1}_{i-1/2,j}\rho^{n+1}_{i-1/2,j}}{\Delta x} + \frac{\tilde{v}^{n+1}_{i,j+1/2}\rho^{n+1}_{i,j+1/2} - \tilde{v}^{n+1}_{i,j-1/2}\rho^{n+1}_{i,j-1/2}}{\Delta y} = 0, \tag{27}$$

$$\frac{\rho u^{n+1}_{i,j} - \rho u^{n}_{i,j}}{\Delta t} + \frac{\tilde{u}^{n+1}_{i+1/2,j}\rho u^{n+1}_{i+1/2,j} + P^{n+1}_{i+1/2,j} - \tilde{u}^{n+1}_{i-1/2,j}\rho u^{n+1}_{i-1/2,j} - P^{n+1}_{i-1/2,j}}{\Delta x} + \frac{\tilde{v}^{n+1}_{i,j+1/2}\rho u^{n+1}_{i,j+1/2} - \tilde{v}^{n+1}_{i,j-1/2}\rho u^{n+1}_{i,j-1/2}}{\Delta y} = 0, \tag{28}$$

$$\frac{\rho v^{n+1}_{i,j} - \rho v^{n}_{i,j}}{\Delta t} + \frac{\tilde{u}^{n+1}_{i+1/2,j}\rho v^{n+1}_{i+1/2,j} - \tilde{u}^{n+1}_{i-1/2,j}\rho v^{n+1}_{i-1/2,j}}{\Delta x} + \frac{\tilde{v}^{n+1}_{i,j+1/2}\rho v^{n+1}_{i,j+1/2} + P^{n+1}_{i,j+1/2} - \tilde{v}^{n+1}_{i,j-1/2}\rho v^{n+1}_{i,j-1/2} - P^{n+1}_{i,j-1/2}}{\Delta y} = 0, \tag{29}$$

$$\frac{E^{n+1}_{i,j} - E^{n}_{i,j}}{\Delta t} + \frac{\tilde{u}^{n+1}_{i+1/2,j}(E+P)^{n+1}_{i+1/2,j} - \tilde{u}^{n+1}_{i-1/2,j}(E+P)^{n+1}_{i-1/2,j}}{\Delta x} + \frac{\tilde{v}^{n+1}_{i,j+1/2}(E+P)^{n+1}_{i,j+1/2} - \tilde{v}^{n+1}_{i,j-1/2}(E+P)^{n+1}_{i,j-1/2}}{\Delta y} = 0, \tag{30}$$

where $\tilde{u}_{i\pm1/2,j}$ and $\tilde{v}_{i,j\pm1/2}$ denote the x- and y-components of the Rhie–Chow fluxing velocities. These velocities are defined by discretizing the momentum (the Navier–Stokes) equations at cell edges in the following manner

$$\tilde{u}^{n+1}_{i\pm1/2,j} = \tilde{u}^{n}_{i\pm1/2,j} \mp \frac{\Delta t}{\rho^{n+1}_{i\pm1/2,j}} \frac{P^{n+1}_{i\pm1,j} - P^{n+1}_{i,j}}{\Delta x} \mp \frac{\Delta t}{\rho^{n+1}_{i\pm1/2,j}} \frac{H^{n+1}_{i\pm1,j} + H^{n+1}_{i,j}}{2}, \tag{31}$$

where H is the operator which includes the convective flux, viscous stress, and body force terms. Eq. (31) resembles an average of the two momentum equations at $i,\ i\pm1$; however, the averaging of two pressure gradients is replaced by the compact pressure gradient. This equation is equivalent to averaging the cell centered momentum equation to within a second-order truncation error. Moreover, in order to make the fluxing velocity more consistent with the original momentum equations, we define the time level n velocity as

$$\tilde{u}^{n}_{i\pm1/2,j} = \frac{1}{2}\left(\frac{\rho u^{n}_{i,j}}{\rho^{n}_{i,j}} + \frac{\rho u^{n}_{i\pm1,j}}{\rho^{n}_{i\pm1,j}}\right). \tag{32}$$

In our approach, the advected quantities and other cell edge quantities in Eqs. (27)–(30) are computed by the second- or third-order upwinding (i.e. QUICK or PPM [18,35]) and advected by the Rhie–Chow fluxing velocities. Using this discretization, our resulting elliptic pressure equation has a compact stencil.

4. Physics-based preconditioner for Navier–Stokes equations

4.1. Primitive-variable based physics-based preconditioner

Having discussed the spatiotemporal discretization, we now present our PBP in a primitive-variable form. This preconditioner allows us to isolate the implicit treatment of acoustic waves by a single variable, the pressure P. The original Jacobian matrix[1] can be written in the following block form

$$\mathbb{J} = \begin{pmatrix} \mathbb{J}_{\rho,\rho} & \mathbb{J}_{\rho,\rho u} & \mathbb{J}_{\rho,\rho v} & \mathbb{J}_{\rho,E} \\ \mathbb{J}_{\rho u,\rho} & \mathbb{J}_{\rho u,\rho u} & \mathbb{J}_{\rho u,\rho v} & \mathbb{J}_{\rho u,E} \\ \mathbb{J}_{\rho v,\rho} & \mathbb{J}_{\rho v,\rho u} & \mathbb{J}_{\rho v,\rho v} & \mathbb{J}_{\rho v,E} \\ \mathbb{J}_{E,\rho} & \mathbb{J}_{E,\rho u} & \mathbb{J}_{E,\rho v} & \mathbb{J}_{E,E} \end{pmatrix}. \tag{33}$$

Each block element of Eq. (33) represents a coupling of conservative variables. In low to intermediate Mach number flows with high-heat flux, we are interested in the ability to step over three distinct time scales: (a) stiff acoustic waves in low-speed regions, (b) viscous diffusion, and (c) heat conduction in finely gridded boundary layers. The goal of our algorithm is to develop an implicit second-order in time method which will closely follow advection time scales, but not be stability limited by the advection CFL.

The goal of the preconditioning step is to solve the potentially stiff physics implicitly and efficiently by extracting the contribution of these three physics from the original set of equations. First, we transform Eq. (33) using the similarity transformation (16) to get the primitive-variable system (17). The resulting primitive-variable Jacobian matrix $\widetilde{\mathbb{J}}$ has the following form

[1] Which is never explicitly formed in the code.

$$\widetilde{\mathbb{J}} = \begin{pmatrix} \mathbb{J}_{P,P} & \mathbb{J}_{P,u} & \mathbb{J}_{P,v} & \mathbb{J}_{P,e} \\ \mathbb{J}_{u,P} & \mathbb{J}_{u,u} & \mathbb{J}_{u,v} & \mathbb{J}_{u,e} \\ \mathbb{J}_{v,P} & \mathbb{J}_{v,u} & \mathbb{J}_{v,v} & \mathbb{J}_{v,e} \\ \mathbb{J}_{e,P} & \mathbb{J}_{e,u} & \mathbb{J}_{e,v} & \mathbb{J}_{e,e} \end{pmatrix}. \tag{34}$$

A and A^{-1} are the transformation matrices of the form

$$A = \begin{pmatrix} \frac{\partial \rho}{\partial P} & \frac{\partial \rho}{\partial u} & \frac{\partial \rho}{\partial v} & \frac{\partial \rho}{\partial e} \\ \frac{\partial \rho u}{\partial P} & \frac{\partial \rho u}{\partial u} & \frac{\partial \rho u}{\partial v} & \frac{\partial \rho u}{\partial e} \\ \frac{\partial \rho v}{\partial P} & \frac{\partial \rho v}{\partial u} & \frac{\partial \rho v}{\partial v} & \frac{\partial \rho v}{\partial e} \\ \frac{\partial E}{\partial P} & \frac{\partial E}{\partial u} & \frac{\partial E}{\partial v} & \frac{\partial E}{\partial e} \end{pmatrix}, \tag{35}$$

$$A^{-1} = \begin{pmatrix} \frac{\partial P}{\partial \rho} & \frac{\partial P}{\partial \rho u} & \frac{\partial P}{\partial \rho v} & \frac{\partial P}{\partial E} \\ \frac{\partial u}{\partial \rho} & \frac{\partial u}{\partial \rho u} & \frac{\partial u}{\partial \rho v} & \frac{\partial u}{\partial E} \\ \frac{\partial v}{\partial \rho} & \frac{\partial v}{\partial \rho u} & \frac{\partial v}{\partial \rho v} & \frac{\partial v}{\partial E} \\ \frac{\partial e}{\partial \rho} & \frac{\partial e}{\partial \rho u} & \frac{\partial e}{\partial \rho v} & \frac{\partial e}{\partial E} \end{pmatrix}. \tag{36}$$

For preconditioning, we simplify the transformed Jacobian matrix (Eq. (34)) by ignoring (i.e. linearizing at *k*th Newton step): (a) advection, (b) compressible work, and (c) viscous heating. With these simplifications, Picard linearization, and operator splitting (with some reordering), the preconditioning matrix becomes

$$\mathbb{M} = \begin{pmatrix} \mathbb{A}_{e,e} & 0 & 0 & 0 \\ \mathbb{A}_{P,e} & \mathbb{A}_{P,P} & 0 & 0 \\ 0 & \mathbb{A}_{u,P} & \mathbb{A}_{u,u} & 0 \\ 0 & \mathbb{A}_{v,P} & 0 & \mathbb{A}_{v,v} \end{pmatrix}. \tag{37}$$

The preconditioning process approximately solves

$$\mathbb{M}\delta\mathbf{V} = -\widetilde{\mathrm{Res}}(\mathbf{V}). \tag{38}$$

To approximately invert the preconditioning matrix, we employ operator splitting. Because the preconditioning matrix is block lower triangular, Eq. (38) can be solved in the following steps:

$$\begin{aligned} &(1) \quad \mathbb{A}_{e,e}\delta e = -\widetilde{\mathrm{Res}}_e; \\ &(2) \quad \mathbb{A}_{P,P}\delta P = -\widetilde{\mathrm{Res}}_P - \mathbb{A}_{P,e}\delta e; \\ &(3) \quad \mathbb{A}_{u,u}\delta u = -\widetilde{\mathrm{Res}}_u - \mathbb{A}_{u,P}\delta P; \\ &(4) \quad \mathbb{A}_{v,v}\delta v = -\widetilde{\mathrm{Res}}_v - \mathbb{A}_{v,P}\delta P. \end{aligned} \tag{39}$$

Each scalar parabolic or elliptic system is approximately inverted with a fixed number of multigrid V-cycles. Next, we provide the explicit form of the discrete operator $\mathbb{M}$ and compare with the PDEs.

(*Step 1*) *Transformation to the primitive variables:*
The first step is to transform the residual RHS vector in the primitive-variable form as follows

$$\widetilde{\mathrm{Res}}(\mathbf{V}) = A^{-1}\mathrm{Res}(\mathbf{U}). \tag{40}$$

Note that since this is a local transformation, and A^{-1} has an explicit form (Eq. (36)), this operation is computationally inexpensive.

(*Step 2*) *Heat conduction* $\mathbb{A}_{e,e}\delta e = -\widetilde{\mathrm{Res}}_e$*:*
Once the RHS is transformed, we precondition the internal energy equation targeting heat conduction. To obtain a simplified internal energy equation for preconditioning, we acknowledge that in our target flow regime (1) the time scale of advection is much slower than that of heat conduction, (2) viscous heating is insignificant, and (3) time rate of change in kinetic energy is small compared to internal energy. Then, the advection term $\nabla \cdot (u(E+P))$ and the viscous heating term $\nabla \cdot (\underline{\tau} \cdot \vec{u})$ can be ignored. The corresponding PDE which defines $\mathbb{A}_{e,e}$ step may be written as

$$\frac{\partial e}{\partial t} - \frac{1}{\rho}\nabla \cdot \left(\frac{\kappa}{c_v}\nabla e\right) = 0, \tag{41}$$

and the discrete equation as

$$\frac{e_{i,j}^{n+1} - e_{i,j}^{n}}{\Delta t} - \frac{\kappa}{c_v \rho_{i,j}^{n+1}}\left(\frac{e_{i+1,j}^{n+1} - 2e_{i,j}^{n+1} + e_{i-1,j}^{n+1}}{\Delta x^2} + \frac{e_{i,j+1}^{n+1} - 2e_{i,j}^{n+1} + e_{i,j-1}^{n+1}}{\Delta y^2}\right) = 0. \tag{42}$$

Note that Eq. (42) is decoupled from other primitive variables, leading to the first row of Eq. (37). The outer Newton iteration is solving for a delta (or an update) and thus we cast our preconditioning step in the same "delta" form

$$\frac{\delta e_{i,j}}{\Delta t} - \frac{\kappa}{c_v \rho^k_{i,j}} \left(\frac{\delta e_{i+1,j} - 2\delta e_{i,j} + \delta e_{i-1,j}}{\Delta x^2} + \frac{\delta e_{i,j+1} - 2\delta e_{i,j} + \delta e_{i,j-1}}{\Delta y^2} \right) = -\widetilde{\mathrm{Res}}_e, \tag{43}$$

or

$$\mathbb{A}_{e,e}\delta e = -\widetilde{\mathrm{Res}}_e, \tag{44}$$

where $\delta e = e^{k+1} - e^k$. For example, the local matrix component for the internal energy equation then becomes

$$\mathbb{A}^{i,j}_{e,e} = \begin{bmatrix} \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i-1,j+1}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i,j+1}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i+1,j+1}} \\ \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i-1,j}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i,j}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i+1,j}} \\ \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i-1,j-1}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i,j-1}} & \frac{\partial \widetilde{\mathrm{Res}}^{i,j}_e}{\partial e_{i+1,j-1}} \end{bmatrix} = \frac{\kappa}{c_v \rho^k_{i,j}} \begin{bmatrix} 0 & -\frac{1}{\Delta y^2} & 0 \\ -\frac{1}{\Delta x^2} & \left(\frac{c_v \rho^k_{i,j}}{\kappa \Delta t} + \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right) & -\frac{1}{\Delta x^2} \\ 0 & -\frac{1}{\Delta y^2} & 0 \end{bmatrix}. \tag{45}$$

Note that density is linearized at the current *k*th Newton step.

(*Step 3*) *Pressure waves* $\mathbb{A}_{P,P}\delta P = -\widetilde{Res}_P - \mathbb{A}_{P,e}\delta e$*:*

The next step is preconditioning of the stiff acoustic waves. To accomplish this, we will approximately invert a pressure Poisson equation. In order to produce the pressure Poisson equation, traditional pressure-based semi-implicit schemes employ the substitution of the momentum equation (Eqs. (24) and (25)), ignoring advection, viscosity and gravity, into the continuity equations (Eq. (23)). This simplification allows one to decouple the pressure Poisson equation from the velocity equations, leading to the second row of Eq. (37). Then, the linearized equation of state is utilized to transform density into pressure [13,21]. However, if ∇P and $\nabla \cdot \rho\vec{u}$ are both evaluated with standard second-order central difference formulas, the resulting pressure Poisson equation will have non-compact stencils. Non-compact stencils can lead to a degraded performance of the multigrid based preconditioner. As discussed in Section 3.1, we employ Rhie–Chow discretization instead. Substitution of Eq. (31) into Eq. (27), together with Rhie–Chow based interpolation, results in the following:

$$\begin{aligned} &\frac{\rho^{n+1}_{i,j} - \rho^n_{i,j}}{\Delta t} + \frac{\tilde{u}^n_{i+\frac{1}{2},j}\rho^{n+1}_{i+\frac{1}{2},j} - \tilde{u}^n_{i-\frac{1}{2},j}\rho^{n+1}_{i-\frac{1}{2},j}}{\Delta x} - \Delta t \frac{H^{n+1}_{i+1,j} - H^{n+1}_{i-1,j}}{2\Delta x} + \frac{\tilde{v}^n_{i,j+\frac{1}{2}}\rho^{n+1}_{i,j+\frac{1}{2}} - \tilde{v}^n_{i,j-\frac{1}{2}}\rho^{n+1}_{i,j-\frac{1}{2}}}{\Delta y} \\ &\quad - \Delta t \frac{H^{n+1-H^{n+1}_{i,j-1}}_{i,j+1}}{2\Delta y} - \Delta t \frac{P^{n+1}_{i+1,j} - 2P^{n+1}_{i,j} + P^{n+1}_{i-1,j}}{(\Delta x)^2} - \Delta t \frac{P^{n+1}_{i,j+1} - 2P^{n+1}_{i,j} + P^{n+1}_{i,j-1}}{(\Delta y)^2} = 0. \end{aligned} \tag{46}$$

Note that we have assumed the fluxing density in Eq. (27) is equal to the face centered density in Eq. (31). We can utilize the linearized equation of state,

$$\delta P = \left.\frac{\partial P}{\partial \rho}\right|_e \delta\rho + \left.\frac{\partial P}{\partial e}\right|_\rho \delta e, \tag{47}$$

to obtain the "delta" form

$$\left(\frac{\partial P}{\partial \rho}\right)^{-1} \frac{\delta P_{i,j}}{\Delta t} - \frac{\delta P_{i+1,j} - 2\delta P_{i,j} + \delta P_{i-1,j}}{\Delta x^2}\Delta t - \frac{\delta P_{i,j+1} - 2\delta P_{i,j} + \delta P_{i,j-1}}{\Delta y^2}\Delta t = -\widetilde{\mathrm{Res}}_P + \left(\frac{\partial P}{\partial \rho}\right)^{-1}\left(\frac{\partial P}{\partial e}\right)\delta e \tag{48}$$

or

$$\mathbb{A}_{P,P}\delta P = -\widetilde{\mathrm{Res}}_P - \mathbb{A}_{P,e}\delta e. \tag{49}$$

Importantly, the above discretized pressure Poisson equation has a compact stencil and is consistent with the original discretization of the nonlinear functions being solved by JFNK.

(*Step 4*) *Viscosity:*

The last step is preconditioning of the viscosity effects. For this, we approximately invert a simplified viscous operator to avoid the need for system multigrid. Applying system multigrid to the approximate inversion of the coupled momentum equations is an option we do not consider here. Under incompressible, constant viscosity limit, and ignoring advection, we can decouple the momentum equation as follows:

$$\frac{\partial u}{\partial t} - \frac{\mu}{\rho}\nabla^2 u = -\frac{1}{\rho}\frac{\partial P}{\partial x}, \tag{50}$$

$$\frac{\partial v}{\partial t} - \frac{\mu}{\rho}\nabla^2 v = -\frac{1}{\rho}\frac{\partial P}{\partial y}. \tag{51}$$

Because the viscous operator is discretized by second-order central differencing, similar to the heat conduction with the Newton linearization of advection operator, we can readily create a consistent compact discretization of the viscous equation as

$$\frac{u^{n+1}_{i,j} - u^{n}_{i,j}}{\Delta t} - \frac{\mu}{\rho^{n+1}_{i,j}} \left(\frac{u^{n+1}_{i+1,j} - 2u^{n+1}_{i,j} + u^{n+1}_{i-1,j}}{\Delta x} + \frac{u^{n+1}_{i,j-1} - 2u^{n+1}_{i,j} + u^{n+1}_{i,j-1}}{\Delta y} \right) = -\frac{1}{\rho} \frac{\partial P}{\partial x}, \tag{52}$$

$$\frac{v^{n+1}_{i,j} - v^{n}_{i,j}}{\Delta t} - \frac{\mu}{\rho^{n+1}_{i,j}} \left(\frac{v^{n+1}_{i+1,j} - 2v^{n+1}_{i,j} + v^{n+1}_{i-1,j}}{\Delta x} + \frac{v^{n+1}_{i,j+1} - 2v^{n+1}_{i,j} + v^{n+1}_{i,j-1}}{\Delta y} \right) = -\frac{1}{\rho} \frac{\partial P}{\partial y}. \tag{53}$$

Corresponding "delta" forms, accounting for the pressure update, are

$$\frac{\delta u_{i,j}}{\Delta t} - \frac{\mu}{\rho^{k}_{i,j}} \left(\frac{\delta u_{i+1,j} - 2\delta u_{i,j} + \delta u_{i-1,j}}{\Delta x} + \frac{\delta v_{i,j+1} - 2\delta u_{i,j} + \delta u_{i,j-1}}{\Delta y} \right) = -\widetilde{\mathrm{Res}}_u - \frac{1}{\rho} \frac{\partial \delta P}{\partial x}, \tag{54}$$

or

$$\mathbb{A}_{u,u}\delta u = -\widetilde{\mathrm{Res}}_u - \mathbb{A}_{u,p}\delta P, \tag{55}$$

$$\frac{\delta v_{i,j}}{\Delta t} - \frac{\mu}{\rho^{k}_{i,j}} \left(\frac{\delta v_{i+1,j} - 2\delta v_{i,j} + \delta v_{i-1,j}}{\Delta x} + \frac{\delta v_{i,j+1} - 2\delta v_{i,j} + \delta v_{i,j-1}}{\Delta y} \right) = -\widetilde{\mathrm{Res}}_v - \frac{1}{\rho} \frac{\partial \delta P}{\partial y} \tag{56}$$

or

$$\mathbb{A}_{v,v}\delta v = -\widetilde{\mathrm{Res}}_v - \mathbb{A}_{v,p}\delta P. \tag{57}$$

Steps 2–4 require the approximate inversion of four parabolic systems. We would like to emphasize that these standard five-point Laplacian matrices are the only matrices formed. To approximately solve these elliptic/parabolic systems, we use five V-cycles of standard geometric MG with the line smoothing. Of course the extension to 3D and unstructured grids will require five parabolic systems and more sophisticated multilevel methods.

(*Step 5*) *Transformation back to the conservative variables:*

The last step is to transform the Newton correction term back to the conservative variables. We use similarity matrix to perform the following transformation

$$\delta \mathbf{U} = A\delta \mathbf{V}. \tag{58}$$

To summarize, we are simply using semi-implicit + operator splitting strategy to precondition JFNK.

5. Numerical examples

In this section, we demonstrate efficiency of the JFNK–PBP framework. We solve several different test problems, varying from a slow incompressible natural circulation ($M \sim 10^{-4}$) to a fast transient ($M \sim 1$) in order to test wide range of applicability of our PBP.

5.1. Thermally driven square cavity problem

As the first example, we solve the thermally driven square cavity problem [7,8,12,24]. The problem consists of the unit square cavity with insulated horizontal walls and heated/cooled vertical walls. The flow in this problem is driven by the buoyancy forces created by the variation in density due to temperature difference. This problem can be categorized by the Prandtl (Pr) and Rayleigh (Ra) numbers and the non-dimensional temperature difference

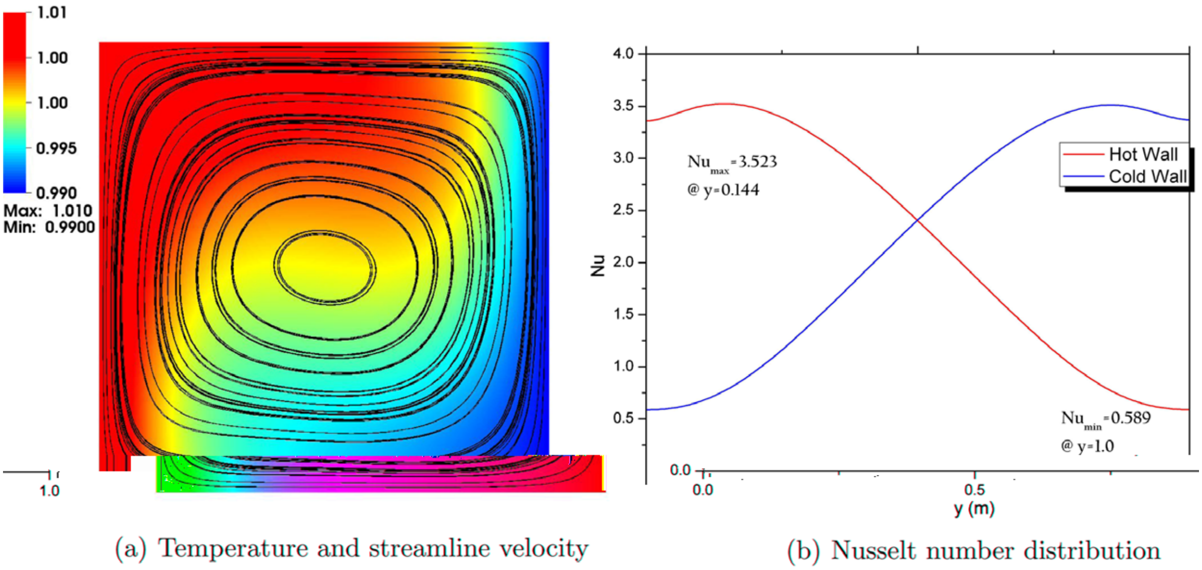
$$\epsilon = \frac{T_h - T_c}{T_h + T_c}. \tag{59}$$

In this section, we solve two distinct cases, where the three parameters (Pr, Ra, ϵ) are $(0.71, 10^4, 0.01)$and $(0.71, 10^6, 0.6)$, respectively. The first problem, with low Ra and ϵ, is one of a group of well-known benchmark problems given in [7,8], and can be modeled appropriately by incompressible + Boussinesq approximations. However, the large disparity between the acoustic mode and dynamical mode introduces stiffness and difficulty solving with compressible flow models. Therefore, it serves as a very challenging test problem for our algorithm.

On the other hand, the high-temperature difference case introduces significant density variations. Therefore, incompressible + Boussinesq approximations fail to reproduce the benchmark solution given in [12,24]. We demonstrate the developed JFNK–PBP framework can solve both cases accurately and efficiently.

5.1.1. Pseudo-transient case

First, we show the result of a steady state problem solved via pseudo-transient continuation. Under relatively low Rayleigh numbers, the problem has a unique steady state solution. The steady state streamline velocity and temperature distributions are depicted in Fig. 1(a). The maximum Mach number of the problem is found to be $\sim 2 \times 10^{-4}$. The local Nusselt number distribution along the vertical walls are shown in Fig. 1(b). Both the maximum and minimum Nusselt numbers along the vertical walls match within 0.2% of the benchmark solution given in [7]. We have used the Backward Euler method for



(a) Temperature and streamline velocity

(b) Nusselt number distribution

Fig. 1. Flow field and Nusselt number distribution.

the temporal integration. The switched evolution relaxation (SER) [23] is employed as the dynamical time step control. The SER adjusts the time step size by the magnitude of the nonlinear residual norm

$$\Delta t^n = \Delta t^0 \frac{\|\mathrm{Res}(\mathbf{U}^0)\|_\infty}{\|\mathrm{Res}(\mathbf{U}^{n-1})\|_\infty}. \tag{60}$$

Table 1 compares the performance of the JFNK–PBP with different mesh sizes. The 5th and 6th columns displays the average of the maximum acoustic and material CFL at each time steps, defined as

$$\mathrm{CFL_a} = \left\| \frac{(|u| + c)\Delta t}{h} \right\|_\infty, \tag{61}$$

$$\mathrm{CFL_m} = \left\| \frac{|u|\Delta t}{h} \right\|_\infty, \tag{62}$$

where $u,\ c$, and h are the local velocity of fluid, the local sound speed, and mesh length, respectively. $\mathrm{CFL_a}$ is the measure of how much fraction the time step exceeds the explicit stability limit, and $\mathrm{CFL_m}$ is the measure of how closely the dynamical time scale is followed. The number of Newton iterations per time step as well as those of Krylov iterations per Newton step stay relatively constant. The last row of Table 1 shows the performance of our preconditioner with a fixed acoustic CFL. Although the GMRES and Newton iterations converges much faster for the fixed (and small) CFL case, the total number of Krylov iterations are smaller when advective time scale is followed, which reflects in the overall CPU time.

5.1.2. Temporal convergence test

Next, we demonstrate the accuracy of the temporal discretization. In this regard, we consider a transient problem, by setting time-dependent temperature boundary condition at the hot wall as

$$T_h(t) = T_{h0} + \alpha t, \tag{63}$$

Table 1
Efficiency of the operator-split preconditioner in pseudo-transient problem.

Grid size	Total time steps	Average $\frac{\text{GMRES}}{\text{Newton}}$	Average $\frac{\text{Newton}}{\text{time step}}$	Average $\mathrm{CFL_a}$[a]	Average $\mathrm{CFL_m}$[b]	Normalized CPU time
64 × 64	290	20.3	1.68	3093	0.62	1
128 × 128	509	16.9	1.47	4253	0.86	3.62
256 × 256	965	14.8	1.34	4456	0.90	27.3
64 × 64[c]	133,333	2.5	1.0	80	0.016	6.64

[a] Average of maximum acoustic CFL.
[b] Average of maximum material CFL.
[c] With the fixed acoustic CFL.

where $\alpha = \frac{(1+\epsilon)}{10^5}$ in the non-dimensionalized form. The initial condition of the problem was set to be the steady state solution with T_{h0} (Fig. 1(a)). Fig. 2 shows time step convergence for the 1st (BE1) and the 2nd (BDF2) order time integration schemes. The simulation was run for 800 dimensionless time. The reference solution was computed using the BDF2 time integration with $\Delta t = 0.2$. As can be seen from the figure, both BE1 and BDF2 exhibit nearly theoretical convergence rate. However, BDF2 can obtain as much as two orders of magnitude more accurate result with the same time step size (i.e. $\Delta t = 40$ in Fig. 2). Table 2 compares the efficiency between BE1 and BDF2. It is clear that BDF2 is more efficient than BE1 to obtain a similar accuracy. The material CFL number varies between 0.07 and 0.75 for $\Delta t = 10$, while it varies between 0.7 and 7.5 for $\Delta t = 100$. Because the PBP is most efficient at material CFL ~ 1, the number of Krylov iterations is somewhat larger for BDF2 case. This is the reason for the computational time not being scaled with the number of time steps.

5.1.3. Transient case

In order to perform another estimate of the preconditioner performance, we consider a second transient problem, by setting time-dependent temperature boundary conditions at the hot wall as

$$T_h(t) = T_0 + \Delta T \cos[0.4\pi t], \tag{64}$$

where $T_0 = 1$ and $\Delta t = \epsilon$. The simulation was run for 5.0 s (i.e. 2075 dimensionless time). The transient problem was solved by BDF2. We compare the number of iterations (both Newton and GMRES) under variation of two parameters: mesh size (64 × 64, 128 × 128, and 256 × 256) and CFL numbers (250, 1000, and 4000). Dynamics of the transient problem are shown in Fig. 3. Table 3 lists the number of iterations used to solve this transient problem under fixed CFL numbers. From this table, the effectiveness of the developed preconditioner is apparent. It is also seen to be independent of the mesh size and only weakly dependent on the CFL number.

5.1.4. High-temperature difference case

The non-dimensional temperature difference in this example is 0.6, and the Rayleigh number is 10^6. Due to the large temperature difference, the Boussinesq approximation cannot adequately model the buoyancy forces (Fig. 4(b)); therefore, the incompressible models fail to obtain an accurate solution although flow is still relatively slow ($M \sim 10^{-3}$). The purpose of this example is to demonstrate the performance of our algorithm in these high-heat flux conditions.

We solve this problem as a pseudo-transient problem. Fig. 4(a) depicts the steady state temperature distribution and velocity stream lines. The asymmetry in the flow can be observed. The density varies by a factor of four between the hot and cold walls, which leads to $\sim 15\%$ difference in the maximum Nusselt number [12,24]. This asymmetric solution cannot

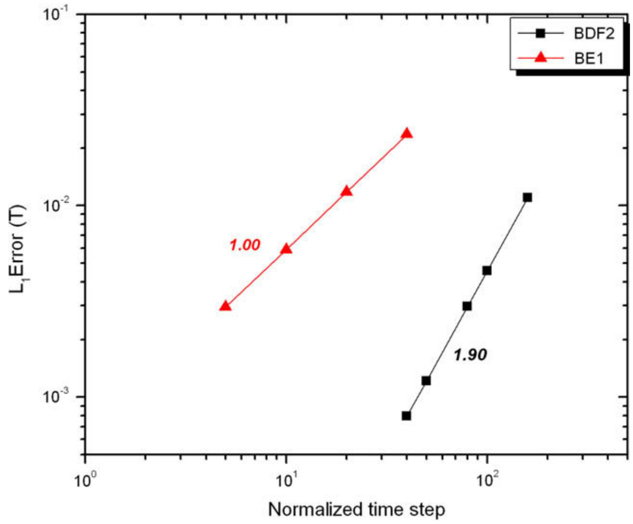
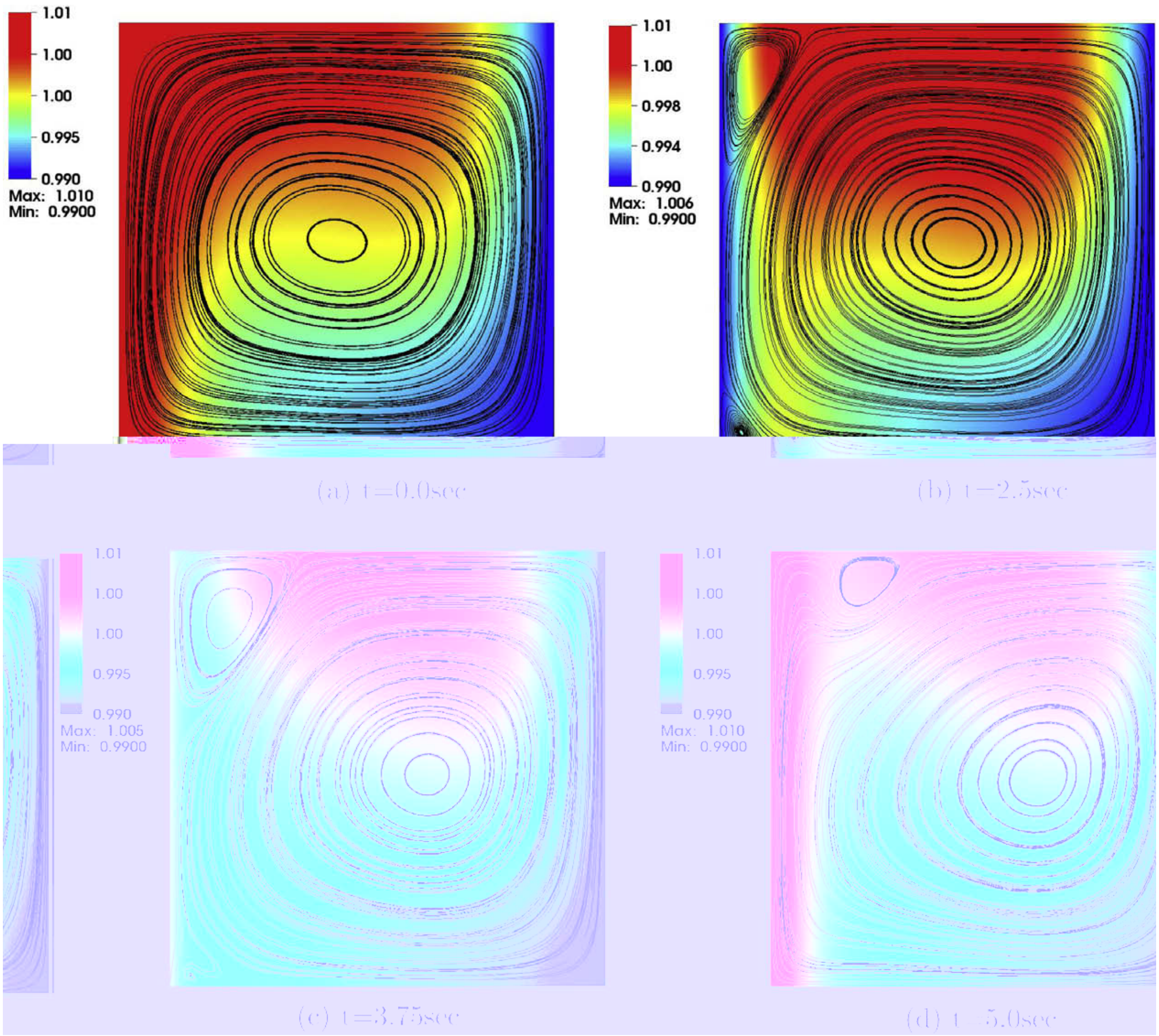


Fig. 2. Error vs. time step size for thermally driven square cavity problem. Reference solution was computed using normalized time step size 1.

Table 2
Efficiency comparison between BE1 and BDF2.

	Δt	Error	Normalized CPU time
BE1	10	5.9×10^{-2}	2.7
BDF2	100	4.6×10^{-2}	1.0



(a) t=0.0sec

(b) t=2.5sec

(c) t=3.75sec

(d) t=5.0sec

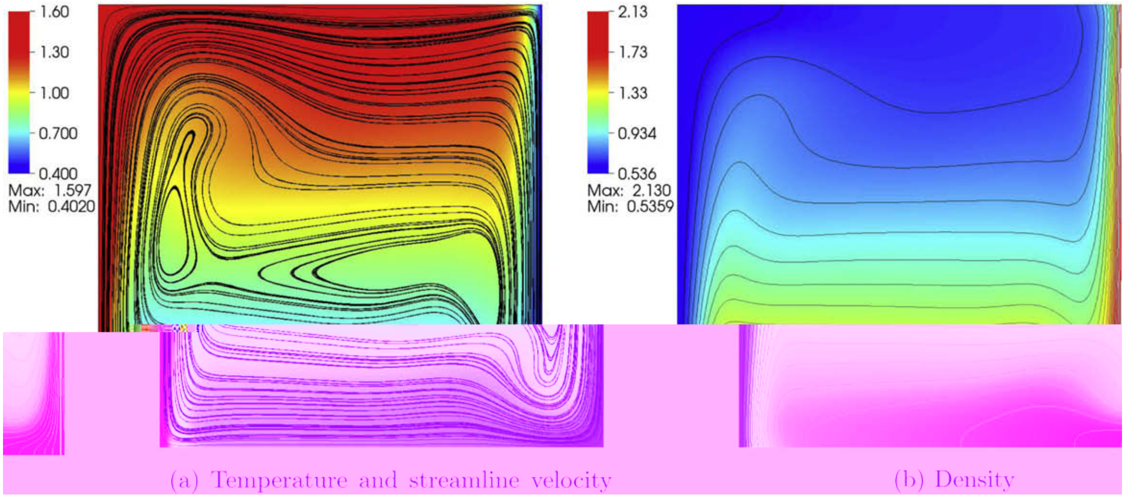
Fig. 3. Temperature distributions of the thermally driven cavity flow transient problem for time.

Table 3
Efficiency of the operator-split preconditioner in transient problem. Dynamics of the temperature field are depicted in Fig. 3.

Grid size	$\text{CFL}_a \approx 250$		$\text{CFL}_a \approx 1000$		$\text{CFL}_a \approx 4000$	
	$\frac{\#\text{Krylov}}{\text{Newton}}$	$\frac{\#\text{Newton}}{\text{time step}}$	$\frac{\#\text{Krylov}}{\text{Newton}}$	$\frac{\#\text{Newton}}{\text{time step}}$	$\frac{\#\text{Krylov}}{\text{Newton}}$	$\frac{\#\text{Newton}}{\text{time step}}$
64 × 64	5.5	1.0	7.3	2.0	14.8	3.2
128 × 128	5.0	1.0	7.1	1.9	12.6	3.3
256 × 256	4.1	1.0	7.2	1.9	11.7	3.4

be obtained by the incompressible Boussinesq approximation flow models. Our result generally matches well with a reference result (<1% error in the maximum Nusselt number with 256 × 256 grid) given in [12].

Table 4 summarizes the algorithm efficiency in this example. By closely following the material CFL (i.e. CFL_m), the average number of GMRES iterations per Newton step and Newton steps per time step are almost independent of mesh sizes. The performance of PBP are similar to the previous case (Table 1). However, the CFL_a number used is smaller. This is due to the time stepping algorithm following the advective time scales. Since we have ignored the advection terms in our preconditioner, it is expected to perform best near the material $\text{CFL} \sim 1$.



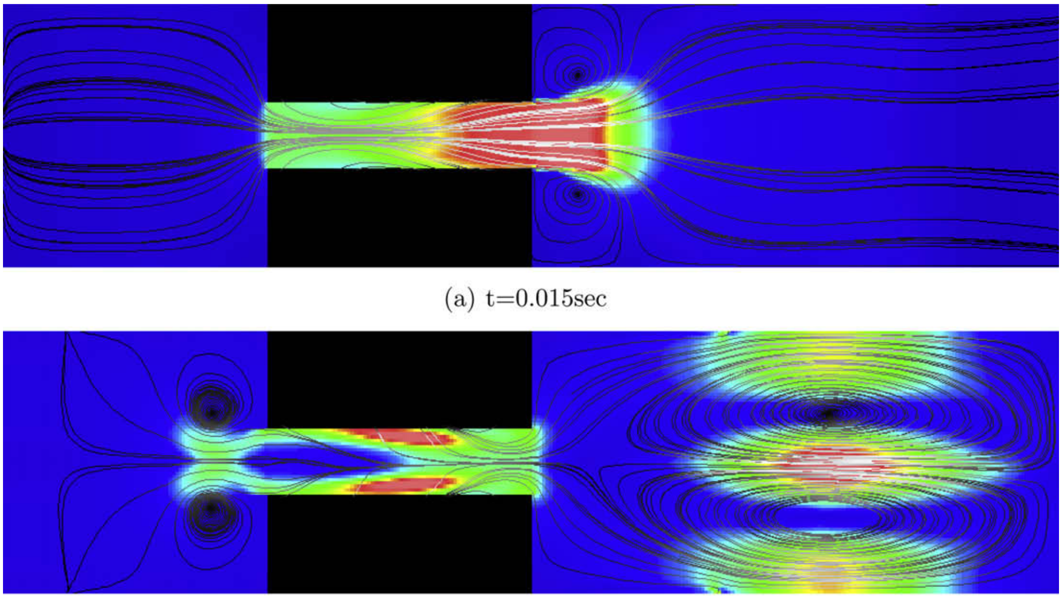
(a) Temperature and streamline velocity

(b) Density

Fig. 4. Temperature, streamlines and density for the thermally driven square cavity problem with $\epsilon = 0.6$ and $Ra = 10^6$.

Table 4
Efficiency of PBP in high-temperature difference case.

Grid size	Total time steps	Average $\frac{\text{GMRES}}{\text{Newton}}$	Average $\frac{\text{Newton}}{\text{time step}}$	Average CFL_a	Average CFL_m	Normalized CPU time
64×64	1561	11.9	2.13	518.4	0.883	1
128×128	2880	13.9	2.26	569.2	0.943	7.98
256×256	5755	10.6	1.90	562.3	0.939	47.73



(a) t=0.015sec

(b) t=0.075sec

Fig. 5. Mach number and streamline plot for the 2D blowdown problem.

5.2. 2D blowdown example

One of the important HTGR accident scenarios is the loss of forced cooling accident with depressurization (DLOFC) due to breakage of pipes [30]. In this accident scenario, the choked flow is likely to be developed due to large pressure difference between the core and atmospheric pressure (e.g. 90:1 pressure ratio in a typical HTGR design). There are two distinct time scales in the DLOFC accident: fast initial blowdown and slow air ingress.

In this example, the initial blowdown of the DLOFC is simulated by a simple and reduced scale two-dimensional depressurization model. Two large tanks (4 m × 4 m left and 12 m × 4 m right) with a pressure ratio of 2:1 are connected by a pipe (4 m × 1.2 m). At $t = 0$, the pipe breaks and starts the transient. Due to the large size of the tank, the flow speed varies from $0 < M < 1$. Fig. 5(a) and(b) depict the Mach number distribution and stream line velocity at time 0.015 and 0.075 s, respectively, while Fig. 6(a)–(d) show the pressure at $y = 2.0$. It can be seen from Fig. 6(c) that the pressure inside the left tank becomes lower than atmospheric pressure. This results in the reverse in flow direction, which is observed by comparing Fig. 5(a) and (b). Because this example is a reduced scale model, the physics of the problem may not reflect that of the true scale problem. To model a real nuclear reactor, a more detailed model needs to be analyzed. To perform the detailed analysis, a proper shock capturing scheme and unstructured grid capability are required. However, this problem requires the capabil-

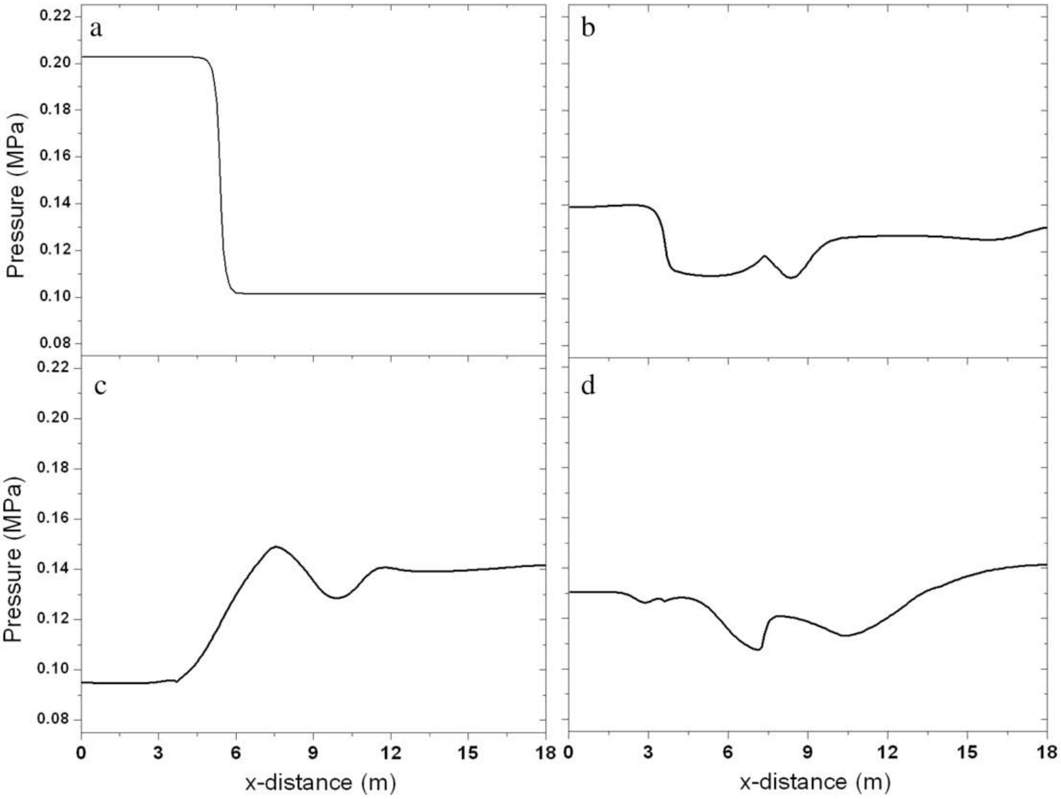


Fig. 6. Pressure at $y = 2.0$; time (a) $t = 0.0$ s, (b) $t = 0.03$ s, (c) $t = 0.06$ s, and (d) $t = 0.09$ s.

Table 5
Efficiency of PBP in 2D blowdown example with various mesh sizes (pressure ratio 2:1).

Grid size	Average $\frac{\text{GMRES}}{\text{Newton}}$	Average $\frac{\text{Newton}}{\text{time step}}$	Average CFL_a	Average CFL_m	Normalized CPU time
100 × 20	6.7	9.7	2.4	0.69	1
200 × 40	6.9	4.0	2.5	0.79	3.0
400 × 80	7.6	4.8	2.6	0.89	30.0

Table 6
Efficiency of PBP in 2D blowdown example with various pressure ratio (grid size 200 × 40).

Pressure ratio	Average $\frac{\text{GMRES}}{\text{Newton}}$	Average $\frac{\text{Newton}}{\text{time step}}$	$M_{\max}$	Average CFL_a	Average CFL_m	Normalized CPU time
2:1	6.9	4.0	0.59	2.5	0.79	1
5:1	7.2	4.7	1.48	1.7	0.84	2.4
10:1	9.0	5.4	2.13	1.9	1.05	3.3

ity of modeling fast transient which is in the order of acoustic modes. Thus, it serves as a challenging problem for low-Mach flow algorithms.

Table 5 compares the efficiency of our preconditioner with various mesh sizes. The pressure ratio 2:1 produces the maximum Mach number of approximately 0.6. The average number of GMRES iterations is again almost independent of mesh sizes while we closely follow the advective time scale of the problem. Table 6 compares the preconditioner performance with various pressure ratio. The choked flow occurs in the cases with pressure ratios of 5:1 and 10:1. Even with these cases, the number of GMRES iterations stays relatively small when the advective time scale is followed. The ability to solve a wide range of flow speeds without any modification in the algorithm is desirable to analyze the transient phenomena which involves stages with both fast dynamics and slow transients.

6. Conclusion

In this paper, we have developed an efficient implicit solution algorithm for low to intermediate Mach number flows. Although the computational effort per time step can be relatively large, an implicit time integration of the problem allows one to step over the fast time scales without stability consideration, which ultimately leads to an overall high level of efficiency. The expensive Jacobian formation required by the Newton method was avoided by the Jacobian-free formulation, and the numerical stiffness due to the disparity of the fast and slow time scales are remedied by employing a physics-based preconditioner of the Krylov method. Our physics-based preconditioner resembles traditional semi-implicit or physics-based splitting algorithms in order to obtain an approximate solution of the distinct physical phenomena efficiently. The preconditioner first performs a consistent transformation of the original conservative-variable form to the primitive-variable form, and then targets the distinct physics (i.e. heat conduction, stiff acoustic waves and viscosity) separately.

The efficiency of the algorithm is further amplified by employment of the Rhie–Chow based interpolation for the advection operators. As a result of the interpolation, a pressure Poisson equation with a compact stencil is formed, while maintaining a consistent discretization with the original equations. The resulting scalar parabolic systems are approximately inverted by a fixed number of the multigrid V-cycles in an efficient manner.

Numerical examples have demonstrated the efficiency and accuracy of the developed algorithm for a wide range of flow conditions. The behavior of the algorithm is almost independent of the mesh sizes and weakly depend on acoustic CFL numbers when the time step size does not exceed advection time scale. An extension of this work to three-dimensional unstructured mesh problems within parallel computational framework is an ongoing effort.

Acknowledgments

The submitted manuscript has been authored by a contractor of the US Government under Contract Nos. DE-AC05-00OR22725 and DE-AC07-05ID14517 (INL/JOU-08-15190). Accordingly, the US Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for US Government purposes. The authors wish to thank Dr. Rick Rauenzahn (LANL) for useful discussion about Rhie–Chow based interpolation.

References

[1] R.A. Berry, Notes on the PCICE method: simplification, generalization and compressibility properties, Journal of Computational Physics 215 (1) (2006) 6.

[2] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM Journal of Science and Statistical Computing 11 (1990) 450–480.

[3] Y.-H. Choi, C.L. Merkle, The application of preconditioning in viscous flows, Journal of Computational Physics 105 (1993) 207.

[4] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, Journal of Computational Physics 2 (1967) 12–26.

[5] P. Colella, K. Pao, A projection method for low speed flows, Journal of Computational Physics 149 (1999).

[6] M. Darbandi, S.F. Hosseinizadeh, Numerical study of natural convection in vertical enclosures using a novel non-Boussinesq algorithm, Numerical heat Transfer Part A: Applications 52 (2007) 849–873.

[7] G. de Vahl Davis, Natural convection of air in a square cavity: a benchmark numerical solution, International Journal for Numerical Methods in Fluids 3 (1983) 249–264.

[8] G. de Vahl Davis, I.P. Jones, Natural convection in a square cavity: a comparison exercise, International Journal for Numerical Methods in Fluids 3 (1983) 227–248.

[9] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM Journal of Numerical Analysis 19 (1982) 400–408.

[10] K. J Evans, D. A Knoll, M.A. Pernice, Development of a 2-D algorithm to simulate convection and phase transition efficiently, Journal of Computational Physics (2006).

[11] K.J. Evans, D.A. Knoll, M.A. Pernice, Enhanced algorithm efficiency using a multigrid preconditioner and simple smoother, Journal of Computational Physics (2007).

[12] H. Paillère, P. Le Quéré, C. Weisman, J. Vierendeels, E. Dick, M. Braack, F. Dabbene, A. Beccantini, E. Studer, T. Kloczko, C. Corre, V. Heuveline, M. Darbandi, S.F. Hosseinizadeh, Modelling of natural convection flows with large temperature differences: a benchmark problem for low Mach number solvers. Part 2. Contributions to the june 2004 conference, Mathematical Modelling and Numerical Analysis 39 (3) (2005) 617–621.

[13] F.H. Harlow, A.A. Amsden, A numerical fluid dynamics calculation method for all flow speeds, Journal of Computational Physics 8 (1971) 197–213.

[14] H. Hirsch, Numerical computation of internal and external flows, Computational Methods for Inviscid and Viscous Flows, vol. 2, John Wiley and Sons, 1988.

[15] R. Klein, Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics. 1: One-dimensional flow, Journal of Computational Physics 121 (1995) 213–217.

[16] D.A. Knoll, D. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, Journal of Computational Physics 193 (2004) 357–397.

[17] D.A. Knoll, V.A. Mousseau, L. Chacon, J.M. Reisner, Jacobian-free Newton–Krylov methods for the accurate time integration of stiff wave systems, Journal of Scientific Computing 25 (2005) 213–230.

[18] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Computer Methods in Applied Mechanics and Engineering 19 (1978) 59–98.

[19] M. Liou, A sequel to AUSM. Part II: AUSM$^+$-up for all speeds, Journal of Computational Physics 214 (2006) 137–170.

[20] W. Liu, G. Makhviladze, An implicit finite element solution of thermal flows at low Mach number, Journal of Computational Physics 227 (2008) 2743–2757.

[21] R.C. Martineau, R.A. Berry, The pressure-corrected ICE finite element method (PCICE-FEM) for compressible flows on unstructured meshes, Journal of Computational Physics 198 (2004) 659.

[22] C.L. Merkle, Y.-H. Choi, Computation of low-speed compressible flows with time marching procedures, International Journal for Numerical Methods in Engineering 25 (1988) 831.

[23] W.A. Mulder, B van Leer, Experiments with implicit upwind methods for the Euler equations, Journal of Computational Physics 59 (2) (1985) 232–246.

[24] P. Le Quéré, C. Weisman, H. Paillère, J. Vierendeels, E. Dick, R. Becker, M. Braack, J. Locke, Modelling of natural convection flows with large temperature differences: a benchmark problem for low Mach number solvers. Part 1. Reference solutions, Mathematical Modelling and Numerical Analysis 39 (3) (2005) 609–616.

[25] S. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, 1980.

[26] M. Pernice, H.F. Walker, NITSOL: a Newton iterative solver for nonlinear system, SIAM Journal on Scientific and Statistical Computing (1998).

[27] M. Preric, R. Kessler, G. Scheuerer, Comparison of finite-volume numerical methods with staggered and colocated grids, Computers and Fluids 16 (4) (1988) 389–403.

[28] J.M. Reisner, V.A. Mousseau, A.A. Wyszogrodzki, D.A. Knoll, A fully implicit hurricane model with physics-based preconditioning, Monthly Weather Review 133 (2005) 1003–1022.

[29] J.M. Reisner, A.A. Wyszogrodzki, V.A. Mousseau, D.A. Knoll, An efficient physics-based preconditioner for the fully implicit solution of small-scale thermally driven atmospheric flows, Journal of Computational Physics 189 (1) (2003) 30–44.

[30] F. Reitsma, G. Strydom, J.B.M. de Haas, K. Ivanov, B. Tyobeka, R. Mphahlele, T.J. Downar, V. Seker, H.D. Gougar, D.F. Da Cruz, U.E. Sikik, The PBMR steady-state and coupled kinetics core thermal-hydraulics benchmark test problems, Nuclear Engineering and Design 236 (5–6) (2006) 657–668.

[31] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA Journal 21 (11) (1982) 1525–1532.

[32] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856.

[33] J.-S. Shuen, K.-H. Chen, Y.-H. Choi, A coupled implicit method for chemical non-equilibrium flows at all speeds, Journal of Computational Physics 106 (1993) 306.

[34] E. Turkel, Review of preconditioning techniques for fluid dynamics, Applied Numerical Mathematics 12 (1993) 257.

[35] van Leer B, Towards the ultimate conservation difference scheme. IV: A new approach to numerical convection, Journal of Computational Physics 23 (1977) 276–299.