Commun. Comput. Phys.
doi: 10.4208/cicp.020312.180712a

Vol. **13**, No. 5, pp. 1408-1431
May 2013

A Jacobian-Free Newton Krylov Implicit-Explicit Time Integration Method for Incompressible Flow Problems

Samet Y. Kadioglu[1,2,*] and Dana A. Knoll[3]

[1] *Department of Mathematical Engineering, Yıldız Technical University, 34210 Davutpaşa-Esenler, İstanbul, Turkey.*
[2] *Fuels Modeling and Simulation Department, Idaho National Laboratory, P.O. Box 1625, MS 3840, Idaho Falls, ID 83415, USA.*
[3] *Theoretical Division, Los Alamos National Laboratory, P.O. Box 1663, MS B-216, Los Alamos, NM, 87545, USA.*

Received 2 March 2012; Accepted (in revised version) 18 July 2012

Available online 8 October 2012

Abstract. We have introduced a fully second order **IM**plicit/**EX**plicit (IMEX) time integration technique for solving the compressible Euler equations plus nonlinear heat conduction problems (also known as the radiation hydrodynamics problems) in *Kadioglu et al.*, J. Comp. Physics [22, 24]. In this paper, we study the implications when this method is applied to the incompressible Navier-Stokes (N-S) equations. The IMEX method is applied to the incompressible flow equations in the following manner. The hyperbolic terms of the flow equations are solved explicitly exploiting the well understood explicit schemes. On the other hand, an implicit strategy is employed for the non-hyperbolic terms. The explicit part is embedded in the implicit step in such a way that it is solved as part of the non-linear function evaluation within the framework of the Jacobian-Free Newton Krylov (JFNK) method [8, 29, 31]. This is done to obtain a self-consistent implementation of the IMEX method that eliminates the potential order reduction in time accuracy due to the specific operator separation. We employ a simple yet quite effective fractional step projection methodology (similar to those in [11,19,21,30]) as our preconditioner inside the JFNK solver. We present results from several test calculations. For each test, we show second order time convergence. Finally, we present a study for the algorithm performance of the JFNK solver with the new projection method based preconditioner.

AMS subject classifications: 01-08, 35Q35, 76Dxx

Key words: Incompressible flow, Navier-Stokes equations, IMEX method, JFNK method, preconditioner.

*Corresponding author. *Email addresses:* samet.kadioglu@inl.gov (S. Y. Kadioglu), nol@lanl.gov (D. A. Knoll)

1 Introduction

In this paper, we present a second order **IM**plicit/**EX**plicit IMEX method for solving the incompressible Navier-Stokes equations. This paper is a continuation of our previous works [22–24] that concern the radiation hydrodynamics problems. [22] considers the low energy density radiation hydrodynamics as in a diffusion approximation limit which can be modelled by the compressible Euler equations plus a nonlinear heat conduction term in the energy equation. [24] considers the more complicated high energy density radiation hydrodynamics as in a diffusion approximation limit which can be modelled by a combination of a hydrodynamical model that resembles to the compressible Euler equations and a radiation energy model that contains separate radiation energy equation with nonlinear diffusion plus coupling terms to the material. In both papers, the hydrodynamics equations (hyperbolic terms) are treated explicitly and an implicit strategy is employed for the diffusion plus source terms. The implicit/explicit (IMEX) technique in [22,24] is implemented in such a way that the explicit part is solved as part of the nonlinear function evaluation within the framework of the Jacobian-Free Newton Krylov (JFNK) method [8,29,31]. In this kind of implementation, there is a continuous interaction between the implicit and explicit blocks. In other words, the improved solutions (in terms of accuracy) at each non-linear iteration (implicit block) are immediately felt by the explicit block, then the improved hydrodynamics solutions (explicit block) are readily available to form the next set of non-linear residuals in the implicit block. We refer the above described IMEX implementation as to "a self-consistent IMEX method" in which all the nonlinearities of the coupled system are converged. These two examples ([22,24]) represent typical multiple time scale flow problems for which having a second order time convergent algorithm is an important advancement. We remark that we also applied this IMEX methodology to a multi-physics problem that tightly couples the neutron diffusion to a linear mechanics model to simulate experimentally observed certain nuclear fuel material behaviors [25]. In [25], the IMEX method is utilized in such a way that the explicit linear mechanics operator is continuously called within the implicit neutron diffusion solver.

We consider this paper as the prototype for the development of a second order IMEX method for multi-phase flow problems that are modelled by two phase incompressible Navier-Stokes equations. We note that in a typical multi-phase flow, there is a strong coupling between the interface and fluid dynamics, thus it is important to introduce an accurate integration technique that converges all the nonlinearities coming from the coupling of these two different dynamics [27]. This paper will provide us important insights about how to separate operators of the multi-phase flow model self-consistently. The operators of the single phase incompressible Navier-Stokes equations are separated in such a way that the momentum advection terms (hyperbolic terms except the pressure) are solved explicitly, and the viscosity plus pressure terms are treated implicitly. The algorithm is implemented in a similarly way as in [22,24] that the explicit part is solved as part of the nonlinear function evaluation within the implicit loop.

In this paper, we introduce a simple and effective projection method based preconditioner for the JFNK solver. The preconditioner relies on derivation of a simple pressure-Poisson problem that can be easily solved very efficiently by an algebraic multi-grid method, and consequently rapid calculation of projected flow velocities. We note that the velocity correction step can be executed by either accounting viscosity effects or neglecting them. In other words, the velocity correction step can be reduced to a set of simple algebraic operations so long as the problem does not exhibit fine viscous time scales. Our algorithm is related to [7] in that it can be viewed as using [7] as a preconditioner to JFNK. In this way JFNK is used to execute the outer nonlinear iteration discussed in [7]. Our algorithm is also related to the recent work in [18]. In contrast to [18], we replace the outer Picard iteration with a Newton-based JFNK solver.

One reasonable question that can be raised about our IMEX method is why not evaluating the explicit part outside of the implicit loop which is computationally cheaper. This approach is previously used in early IMEX methods [3,4,6,30,35,38] for various problems, i.e., one dimensional scalar advection-diffusion, reaction-diffusion prototype equations, or more complicated models (the Navier-Stokes equations, the radiation hydrodynamics equations). We note that on a certain class of problems this traditional IMEX approach will not produce fully second order time convergent results, in other words, it can suffer from order reduction in time accuracy unless special care (e.g, performing external iterations outside of the both explicit and implicit loops [28]) is taken. The main reason for the order reduction is that the explicit and implicit blocks are implemented independent of each other such that there is no or limited influence from one to other. This lack of continuous interactions between the two algorithm blocks may lead to non-converging nonlinearities in the coupled system which are the main source of the time inaccuracies.

The organization of this paper is as follows. In Section 2, the governing equations are presented. In Section 3, the numerical solution procedure is described. In Section 4, the computational results are presented. Section 5 contains some concluding remarks. Finally, appendix provides the interpolation procedures that are used in Section 3.

2 Governing equations

The non-dimensionalized incompressible Navier-Stokes equations in *two* space dimension can be written as

$$\mathbf{u}_t+\mathbf{u}\cdot\nabla\mathbf{u}=-\nabla p+\frac{1}{Re}\Delta\mathbf{u}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{2.2}$$

where $\mathbf{u}=(u,v)$ is the flow velocity, p is the fluid pressure, and Re is the non-dimensional Reynolds number [5, 10, 49]. Eq. (2.2) is referred to as the divergence free constraint. Incorporating with this constraint, (2.1) can be rewritten in the following form

$$\frac{\partial u}{\partial t}+\frac{\partial(u^2)}{\partial x}+\frac{\partial(uv)}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\Big[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\Big], \tag{2.3}$$

$$\frac{\partial v}{\partial t}+\frac{\partial (uv)}{\partial x}+\frac{\partial (v^2)}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\Big[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\Big]. \tag{2.4}$$

3 Numerical algorithm

Finite difference algorithms for the incompressible Navier-Stokes equations are typically based on staggered or non-staggered (co-located) grids. Non-staggered grids store all the unknowns at the same locations (cell centers). On the other hand, staggered grids generally store the pressure at the cell centers whereas the velocity components lies on the cell faces. Fig. 1 represents staggered versus non-staggered computational cells with the flow variables placed at the above-described locations. The non-staggered grid solutions of the Navier-Stokes equations are known to suffer from grid-scale pressure oscillations unless special cares have been introduced [1, 36]. The staggered grid methods do not suffer from pressure oscillations, but it brings extra coding complexities [1, 2, 21, 36]. In this paper, we prefer staggered grid discretizations. The first reason for this choice is to avoid pressure oscillations. The second is that we would like to extend this work to study incompressible multi-phase flows in which staggered velocity representations (face centered velocities) are known to be better suited for solving the interface dynamics [27, 45, 46].

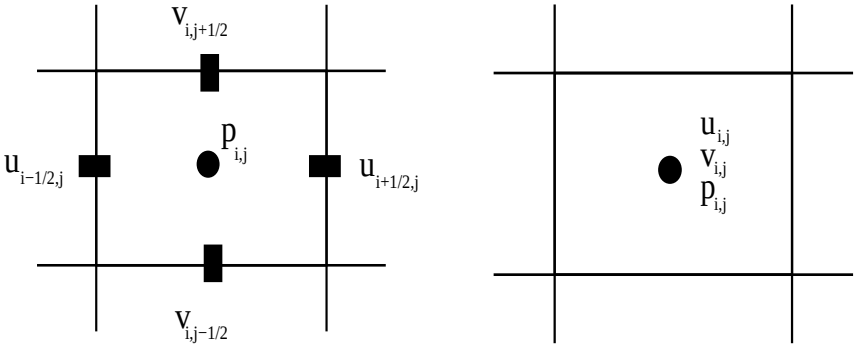

Figure 1: A solution representation on staggered versus non-staggered cells.

Our numerical algorithm consists of an explicit and an implicit blocks. The explicit block solves hyperbolic terms (except the pressure gradients) of (2.3) and (2.4), i.e,

$$\frac{\partial u}{\partial t}+\frac{\partial (u^2)}{\partial x}+\frac{\partial (uv)}{\partial y}=0, \tag{3.1a}$$

$$\frac{\partial v}{\partial t}+\frac{\partial (uv)}{\partial x}+\frac{\partial (v^2)}{\partial y}=0. \tag{3.1b}$$

The implicit block solves

$$\frac{\partial u}{\partial t}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\Big[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\Big], \tag{3.2a}$$

$$\frac{\partial v}{\partial t}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\Big[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\Big], \tag{3.2b}$$

together with the divergence free constraint

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0. \tag{3.3}$$

3.1 Explicit block

The explicit discretization of (3.1) is based on a second order TVD Runge-Kutta method that is known to preserve the strong stability property of the explicit Euler method in order to produce non-oscillatory calculations [16, 17, 33, 42, 43, 48]

$$u^1_{i-1/2,j}=u^n_{i-1/2,j}-\Delta t\Big[\frac{\partial (u^2)^n}{\partial x}+\frac{\partial (uv)^n}{\partial y}\Big]_{i-1/2,j}, \tag{3.4a}$$

$$v^1_{i,j-1/2}=v^n_{i,j-1/2}-\Delta t\Big[\frac{\partial (uv)^n}{\partial x}+\frac{\partial (v^2)^n}{\partial y}\Big]_{i,j-1/2}, \tag{3.4b}$$

and

$$u^*_{i-1/2,j}=(\frac{u^1+u^n}{2})_{i-1/2,j}-\frac{\Delta t}{2}\Big[\frac{\partial (\mathbf{u}^2)^{n+1}}{\partial x}+\frac{\partial (\mathbf{uv})^{n+1}}{\partial y}\Big]_{i-1/2,j}, \tag{3.5a}$$

$$v^*_{i,j-1/2}=(\frac{v^1+v^n}{2})_{i,j-1/2}-\frac{\Delta t}{2}\Big[\frac{\partial (\mathbf{uv})^{n+1}}{\partial x}+\frac{\partial (\mathbf{v}^2)^{n+1}}{\partial y}\Big]_{i,j-1/2}. \tag{3.5b}$$

Here, the explicit block is interacting with the implicit block through the highlighted terms in (3.5). The spatial discretizations involved in (3.4) and (3.5) are as follows

$$\Big[\frac{\partial (u^2)}{\partial x}\Big]_{i-1/2,j}=\frac{U(u^L,u^R)^2_{i,j}-U(u^L,u^R)^2_{i-1,j}}{\Delta x}, \tag{3.6}$$

where we solve a normal Riemann problem for U,

$$U(u^L,u^R)=\begin{cases} u^L, & \text{if } u^L>\epsilon \text{ and } (u^L+u^R)>\epsilon, \\ \frac{u^L+u^R}{2}, & \text{if } u^L<-\epsilon \text{ and } u^R>\epsilon, \\ u^R, & \text{otherwise}, \end{cases} \tag{3.7}$$

where ϵ is used for the preservation of symmetry [12]. Typically ϵ is a small number near the square root of machine varepsilon. For double precision, ϵ can be set from 10^{-6} to

10^{-8} [13]. We note that this Riemann solver works quite well for our test cases. Nonetheless, one can always choose alternative Riemann solvers such as the Local Lax-Friedrichs or Roe solvers to evaluate $U(u^L,u^R)$ [13,33,48].

The *left* and *right* values of u can be calculated either by using a second order interpolation with minmod slopes or by a *third* order WENO-type interpolation [33,34,41,48]. In the appendix, we briefly describe both procedures. We note that the third order WENO-type interpolation gives sharper solution representations. Moreover, [32] suggests that the evaluation of the advective terms with a third order scheme preserves the overall stability property better than a second order one. Therefore we prefer the third order WENO method for our calculations. However, we also make use of the second order interpolation when we perform a convergence study for the analytical solution test (Section 4.1).

Next we define

$$\Big[\frac{\partial(uv)}{\partial y}\Big]_{i-1/2,j}=\frac{(uv)_{i-1/2,j+1/2}-(uv)_{i-1/2,j-1/2}}{\Delta y}, \tag{3.8}$$

where

$$u_{i-1/2,j-1/2}=\frac{u_{i-1/2,j}+u_{i-1/2,j-1}}{2}, \tag{3.9}$$

$$v_{i-1/2,j-1/2}=\frac{v_{i,j-1/2}+v_{i-1,j-1/2}}{2}. \tag{3.10}$$

Notice that we did not use any limiters or a Riemann solver to evaluate (3.9) and (3.10). The reason for this is that we found through our numerical experiments that using limiters and a Riemann solver to evaluate the corner values removes necessary numerical dissipation. In result, one has to use finer grids to resolve complicated flow structures such as sharp vorticity roll ups in Section 4.3.

The spatial terms for the Y-velocity in (3.4) and (3.5) can be handled similarly.

3.2 Implicit block

The explicit block produces the solution set $\{u^*,v^*\}$. These solutions are used in the implicit discretization of (3.2) and (3.3) which is based on the Crank-Nicolson method [44,47], i.e,

$$u^{n+1}_{i-1/2,j}=u^*_{i-1/2,j}+\frac{\Delta t}{2}[f(u^{n+1},p^{n+1})+f(u^n,p^n)]_{i-1/2,j}, \tag{3.11a}$$

$$v^{n+1}_{i,j-1/2}=v^*_{i,j-1/2}+\frac{\Delta t}{2}[g(v^{n+1},p^{n+1})+g(v^n,p^n)]_{i,j-1/2}, \tag{3.11b}$$

$$\frac{u^{n+1}_{i+1/2,j}-u^{n+1}_{i-1/2,j}}{\Delta x}+\frac{v^{n+1}_{i,j+1/2}-v^{n+1}_{i,j-1/2}}{\Delta y}=0, \tag{3.11c}$$

where

$$f(u,p)_{i-1/2,j} = -\left(\frac{\partial p}{\partial x}\right)_{i-1/2,j} + \frac{1}{Re}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right]_{i-1/2,j}, \tag{3.12a}$$

$$g(v,p)_{i,j-1/2} = -\left(\frac{\partial p}{\partial y}\right)_{i,j-1/2} + \frac{1}{Re}\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right]_{i,j-1/2}. \tag{3.12b}$$

The spatial discretizations in this section are based on the second order central differencing. For instance,

$$\left(\frac{\partial p}{\partial x}\right)_{i-1/2,j} = \frac{p_{i,j} - p_{i-1,j}}{\Delta x}, \tag{3.13}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i-1/2,j} = \frac{u_{i+1/2,j} - 2u_{i-1/2,j} + u_{i-3/2,j}}{\Delta x^2}, \tag{3.14}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i-1/2,j} = \frac{u_{i-1/2,j+1} - 2u_{i-1/2,j} + u_{i-1/2,j-1}}{\Delta y^2}, \tag{3.15}$$

and

$$\left(\frac{\partial p}{\partial y}\right)_{i,j-1/2} = \frac{p_{i,j} - p_{i,j-1}}{\Delta y}, \tag{3.16}$$

$$\left(\frac{\partial^2 v}{\partial x^2}\right)_{i,j-1/2} = \frac{v_{i+1,j-1/2} - 2v_{i,j-1/2} + v_{i-1,j-1/2}}{\Delta x^2}, \tag{3.17}$$

$$\left(\frac{\partial^2 v}{\partial y^2}\right)_{i,j-1/2} = \frac{v_{i,j+1/2} - 2v_{i,j-1/2} + v_{i,j-3/2}}{\Delta y^2}. \tag{3.18}$$

After substitution of the spatial discretizations and the other known values, (3.11) can be viewed as a nonlinear equations system to solve for $\{u^{n+1}, v^{n+1}, p^{n+1}\}$. We use the Jacobian-Free Newton Krylov method to solve this nonlinear system. Upon the convergence of the Newton iteration, we obtain the fully updated solution set $\{u^{n+1}, v^{n+1}, p^{n+1}\}$.

Remark 3.1. We note that the combination of the explicit and implicit blocks can be put in a form that is similar to the classical fully implicit Crank-Nicolson method. The main difference is that one does not have to solve Eqs. (3.4) and (3.5) at every Newton/Krylov step (as this would be the case in a classical fully implicit Crank-Nicolson method) in our method. Instead, (3.4) and (3.5) can be called inside of the implicit loop at every few other Newton/Krylov step. Also, we prefer to describe our algorithm this way, because as we emphasized earlier that this lays a foundation about how to separate the operators of the two-phase incompressible flow model that possesses true non-linear coupling between different dynamics and requires this IMEX kind of execution in order to achieve second order time convergent calculations.

3.3 The Jacobian-free Newton Krylov method and forming the IMEX function

The Jacobian-Free Newton Krylov method is the combination of the Newton method that solves the system of nonlinear equations and a Krylov subspace method that solves the Newton correction equations.

The Newton method solves $\mathbf{F}(W)=0$ (assume Eq. (3.11) is written in this form with $W=(u,v,p)$) iteratively over a sequence of linear system defined by

$$\mathbf{J}(W^k)\delta W^k=-\mathbf{F}(W^k), \tag{3.19a}$$

$$W^{k+1}=W^k+\delta W^k, \qquad k=0,1,\cdots, \tag{3.19b}$$

where $\mathbf{J}(T^k)=\frac{\partial \mathbf{F}}{\partial T}$ is the Jacobian matrix and δW^k is the update vector. The Newton iteration is terminated based on a required drop in the norm of the nonlinear residual, i.e,

$$\|\mathbf{F}(W^k)\|_2<tol_{res}\|\mathbf{F}(W^0)\|_2, \tag{3.20}$$

where tol_{res} is a given tolerance.

The linear system (3.19) (Newton correction equation) is solved by using the Arnoldi based Generalized Minimal RESidual method (GMRES) [39] which belongs to the general class of the Krylov subspace methods [37]. In GMRES, an initial linear residual, $\mathbf{r}_0$, is defined for a given initial guess δW_0,

$$\mathbf{r}_0=-\mathbf{F}(W)-\mathbf{J}\delta W_0. \tag{3.21}$$

Here we dropped the index k convention since the Krylov (GMRES) iteration is performed at a fixed k. Let j be the Krylov iteration index. The j^{th} Krylov iteration minimizes $\|\mathbf{J}\delta W_j+\mathbf{F}(W)\|_2$ within a subspace of small dimension, relative to n (the number of unknowns), in a least-squares sense. δW_j is drawn from the subspace spanned by the Krylov vectors, $\{\mathbf{r}_0,\mathbf{J}\mathbf{r}_0,\mathbf{J}^2\mathbf{r}_0,\cdots,\mathbf{J}^{j-1}\mathbf{r}_0\}$, and can be written as

$$\delta W_j=\delta W_0+\sum_{i=0}^{j-1}\beta_i(\mathbf{J})^i\mathbf{r}_0, \tag{3.22}$$

where the scalar β_i minimizes the residual. The Krylov iteration is terminated based on the following inexact Newton criteria [14]

$$\|J\delta W_j+\mathbf{F}(W)\|_2<\gamma\|\mathbf{F}(W)\|_2, \tag{3.23}$$

where the parameter γ is set in terms of how tight the linear solver should converge at each Newton iterations (we typically use $\gamma=10^{-3}$). One particularly attractive features of the GMRES is that it does not require forming the Jacobian matrix. Instead, only matrix-vector multiplications, $\mathbf{J}v$, are needed, where $v\in\{\mathbf{r}_0,\mathbf{J}\mathbf{r}_0,\mathbf{J}^2\mathbf{r}_0,\cdots\}$. This leads to the so-called *Jacobian-Free* implementations in which the action of the Jacobian matrix times a

vector can be approximated by

$$\mathbf{J}v=\frac{\mathbf{F}(W+\epsilon v)-\mathbf{F}(W)}{\epsilon}, \tag{3.24}$$

where $\epsilon=\frac{1}{n\|v\|_2}\sum_{i=1}^{n}b|w_i|+b$, n is the dimension of the linear system and b is a constant whose magnitude is within a few orders of magnitude of the square root of machine roundoff (typically 10^{-6} for 64-bit double precision).

Here, we describe how to form the IMEX function $F(W)$. We will refer $F(W)$ as the IMEX function, since it uses the both explicit and implicit information. The following pseudo code describes how to form $F(W)$.

Evaluating $F(W^k)$:

Given W^k where k is the current Newton iteration.
Call Explicit block with (u^n,v^n,u^k,v^k) to compute u^*,v^*.
Form $F(W^k)=(F_u,F_v,F_p)$ based on the Crank-Nicolson method as

$$F_u=u^k_{i-1/2,j}-u^*_{i-1/2,j}-\frac{\Delta t}{2}[f(u^k,p^k)+f(u^n,p^n)]_{i-1/2,j}, \tag{3.25a}$$

$$F_v=v^k_{i,j-1/2}-v^*_{i,j-1/2}-\frac{\Delta t}{2}[g(v^k,p^k)+g(v^n,p^n)]_{i,j-1/2}, \tag{3.25b}$$

$$F_p=\frac{u^k_{i+1/2,j}-u^k_{i-1/2,j}}{\Delta x}+\frac{v^k_{i,j+1/2}-v^k_{i,j-1/2}}{\Delta y}. \tag{3.25c}$$

3.4 Preconditioning (pressure projection method)

Our preconditioning step is similar to the original MAC projection step of [21] or the fractional step projection method of [11] in the sense that we both solve an intermediate pressure Poisson problem derived from predicted flow velocities. Below, we summarize our preconditioning options. First we write Eqs. (3.2) and (3.3) incorporating with (3.11) and (3.25) in delta correction-residual form as

$$\frac{\delta u}{\Delta t}+\frac{1}{2}\frac{\partial\delta p}{\partial x}-\frac{1}{2Re}\Big[\frac{\partial^2\delta u}{\partial x^2}+\frac{\partial^2\delta u}{\partial y^2}\Big]=-F_u, \tag{3.26}$$

$$\frac{\delta v}{\Delta t}+\frac{1}{2}\frac{\partial\delta p}{\partial y}-\frac{1}{2Re}\Big[\frac{\partial^2\delta v}{\partial x^2}+\frac{\partial^2\delta v}{\partial y^2}\Big]=-F_v, \tag{3.27}$$

$$\frac{\partial\delta u}{\partial x}+\frac{\partial\delta v}{\partial y}=-F_p, \tag{3.28}$$

where $\delta\mathbf{u}=\mathbf{u}^{n+1}-\mathbf{u}^n$. Notice that we multiply pressure gradients and viscosity terms with $1/2$ in order to be consistent with the Crank-Nicolson discretization. To derive an approximate equation for the pressure correction, we ignore the viscosity effects in (3.26)

and (3.27) and differentiate the rest of the equations in terms of x and y respectively. Then we use (3.28) to get

$$\frac{\partial^2 \delta p}{\partial x^2}+\frac{\partial^2 \delta p}{\partial y^2}=-\frac{2}{\Delta t}\Big[\frac{\partial F_u}{\partial x}+\frac{\partial F_v}{\partial y}-F_p\Big]. \tag{3.29}$$

This is the pressure-Poisson problem that we approximately solve as part of our preconditioner. We note that this pressure-Poisson problem can be relaxed by several different ways such as multi-grid, Gauss-Seidel, or a direct solver [39, 44, 47]. We use an algebraic multi-grid method for all of our test calculations. However, we compare multi-grid versus the conventional Gauss-Seidel sweeps when performing nonlinear solver analysis in Section 4.4. After relaxing (3.29) for the pressure correction, we can precondition the components of the velocity corrections. The preconditioning of the components of the velocity corrections without the viscosity effects can be followed from (3.26) and (3.27) as

$$\delta u=-\frac{\Delta t}{2}\frac{\partial \delta p}{\partial x}-F_u, \tag{3.30}$$

$$\delta v=-\frac{\Delta t}{2}\frac{\partial \delta p}{\partial y}-F_v. \tag{3.31}$$

Alternatively, we can relax a Poisson problem for δu and δv derived from (3.26) and (3.27), i.e,

$$\delta u-\frac{\Delta t}{2Re}\Big[\frac{\partial^2 \delta u}{\partial x^2}+\frac{\partial^2 \delta u}{\partial y^2}\Big]=-\frac{\Delta t}{2}\frac{\partial \delta p}{\partial x}-F_u, \tag{3.32}$$

$$\delta v-\frac{\Delta t}{2Re}\Big[\frac{\partial^2 \delta v}{\partial x^2}+\frac{\partial^2 \delta v}{\partial y^2}\Big]=-\frac{\Delta t}{2}\frac{\partial \delta p}{\partial y}-F_v. \tag{3.33}$$

Both velocity preconditioners work well if we solve high Reynolds number flows. However, the second velocity preconditioner results in more effective computations when we deal with low Reynolds number flows (highly viscous flows) or flows containing boundary layers. This is because the governing system poses a fine time scale (viscous time scale) for small Re that the numerical method has to obey for stability and accuracy reasons. Clearly, the explicit formulation, (3.30) and (3.31), has to follow this fine time behavior by taking small time steps. On the other hand, one can step over ($\Delta t > Re\Delta x^2$) the fine time behavior by using the implicit formulation (3.32)-(3.33). Both options are compared/analyzed in Section 4.4.

We note that the order of the implementation of the preconditioning procedures described above is that first we solve a pressure correction equation then update the velocity corrections by either (3.30)-(3.31) or (3.32)-(3.33). However in a classical second order fractional step projection method [7, 18], the first step is to obtain a set of predicted velocity corrections from

$$\Big[1-\frac{\Delta t}{2Re}\nabla^2\Big]\delta u^*=-F_u, \qquad \Big[1-\frac{\Delta t}{2Re}\nabla^2\Big]\delta v^*=-F_v, \tag{3.34}$$

then solve a Poisson problem for a scalar field,

$$\frac{\partial^2 \delta\phi}{\partial x^2}+\frac{\partial^2 \delta\phi}{\partial y^2}=\frac{2}{\Delta t}\Big[\frac{\partial \delta u^*}{\partial x}+\frac{\partial \delta v^*}{\partial y}+F_p\Big]. \tag{3.35}$$

The component of the velocity corrections are updated by

$$\delta u=\delta u^*-\frac{\Delta t}{2}\frac{\partial \phi}{\partial x}, \qquad \delta v=\delta v^*-\frac{\Delta t}{2}\frac{\partial \phi}{\partial y}. \tag{3.36}$$

Then the pressure correction term is calculated with

$$\delta p=\Big[1-\frac{\Delta t}{2Re}\nabla^2\Big]\phi. \tag{3.37}$$

This procedure automatically satisfies the divergence free constraint (3.28). The steps (3.30) and (3.31) together with (3.29) also satisfy the divergence free constraint. However the steps (3.32) and (3.33) do not satisfy the divergence free constraint. We note that the only purpose of the preconditioning is to minimize GMRES iterations. The divergence free constraint is satisfied upon the convergence of the nonlinear Newton iterations. We have compared the second order divergence-free-satisfying fractional step projection method ((3.34)-(3.37)) versus the non-divergence-free-satisfying procedure ((3.32)-(3.33)) and observed small differences in terms of the average number of nonlinear and linear iterations per time steps (refer to Section 4.4).

3.5 Time step control

There have been time step criterion introduced for the Navier-Stokes equations, [15, 20, 36, 49], most of which incorporates with viscosity terms. Our numerical method consists of implicit and explicit blocks. The explicit block of our scheme solves the hyperbolic terms of the Navier-Stokes equations that pose normally more stringent advective time scales (considering that Reynolds number is high in all of our test problems). Therefore it is sufficient to follow explicit time steps to produce accurate and stable solutions. The explicit time steps are computed by

$$\Delta t=C\min\left\{\frac{\Delta x}{\max|u|},\frac{\Delta y}{\max|v|}\right\}, \tag{3.38}$$

where C is a constant number (typically $C=0.5$ [33, 48]).

4 Numerical results

4.1 Travelling wave problem

The following solutions (also referred to as the travelling wave solutions) are known to analytically satisfy (2.1) and (2.2)

$$u(x,y,t) = 0.75 + 0.25\cos(2\pi(x-0.75t))\sin(2\pi(y-0.75t))e^{-8\pi^2 t/Re}, \tag{4.1a}$$

$$v(x,y,t) = 0.75 - 0.25\sin(2\pi(x-0.75t))\cos(2\pi(y-0.75t))e^{-8\pi^2 t/Re}, \tag{4.1b}$$

$$p(x,y,t) = -\frac{1}{64}[\cos(4\pi(x-0.75t)) + \cos(4\pi(y-0.75t))]e^{-16\pi^2 t/Re}. \tag{4.1c}$$

We will compare our numerical solutions against (4.1). We initialize the numerical solutions by setting $t=0$ in (4.1) and carry out our computations in a doubly-periodic unit square. In Tables 1, 2, and 3, we compare *five* different mesh solutions at $time=0.5$. In all calculations the Reynolds number is set to be equal to 10000. We use the following time steps for our convergence study, $\Delta t = \Delta x/4$ assuming $\Delta x = \Delta y$ which can be related to (3.38) by setting $C=1/4$. Tables 1-3 show decrease in errors in L_2 sense and convergence rates due to the consecutive mesh (and so time step) refinements. The convergence rate between the two consecutive mesh refinements is calculated by

$$rate = \log_2(Error_{\Delta x}/Error_{\Delta x/2}), \tag{4.2}$$

where $Error = \|\phi^{num} - \phi^{exact}\|_2$ with ϕ representing the variable of interest. We have checked the convergence rates for both X and Y velocities and the pressure field. Tables 1-3 clearly indicate that we have obtained the desired rate of accuracy (second order) for each variable. We note that we use second order interpolations when constructing the *left* and *right* state variables involved in Section 3.1. This is done to perform a consistent convergence analysis using consistent spatial and time discretizations (both are second order). We also note that if we use the third order WENO-type construction for the *left/right* state variables, then the convergence rates tend to become higher than *two* as one would expect.

Table 1: Convergence rates in X-velocity (u) for problem (4.1).

Mesh Refinement	$\|u^{num} - u^{exact}\|_2$	Convergence rates
$h=1/16$	2.07×10^{-2}	
$h=1/32$	5.48×10^{-3}	1.92
$h=1/64$	1.37×10^{-3}	1.99
$h=1/128$	3.40×10^{-4}	2.01
$h=1/256$	8.43×10^{-5}	2.01

Table 2: Convergence rates in Y-velocity (v) for problem (4.1).

Mesh Refinement	$\|u^{num}-u^{exact}\|_2$	Convergence rates
$h=1/16$	2.07×10^{-2}	
$h=1/32$	5.48×10^{-3}	1.92
$h=1/64$	1.37×10^{-3}	1.99
$h=1/128$	3.40×10^{-4}	2.01
$h=1/256$	8.43×10^{-5}	2.01

Table 3: Convergence rates in pressure field for problem (4.1).

Mesh Refinement	$\|u^{num}-u^{exact}\|_2$	Convergence rates
$h=1/16$	9.24×10^{-3}	
$h=1/32$	2.80×10^{-3}	1.72
$h=1/64$	7.55×10^{-4}	1.89
$h=1/128$	1.94×10^{-4}	1.96
$h=1/256$	4.90×10^{-5}	1.98

4.2 Lid driven cavity problem

Steady state solutions of the *two* dimensional lid driven cavity flow (e.g, refer to [9,15,36, 40]) has become a classical test problem in that one can check the accuracy and efficiency of the designed numerical schemes for different Reynolds numbers and different lid velocities that determine the wall boundary conditions. In this paper, we consider a unit square cavity with a top lid moving with the speed of unity (Fig. 2). The initial flow values are set to be *zero* inside the cavity. The Y-velocity is set to be *zero* on the walls at both normal and tangential directions (referred to as the no-slip and no-penetrating velocity

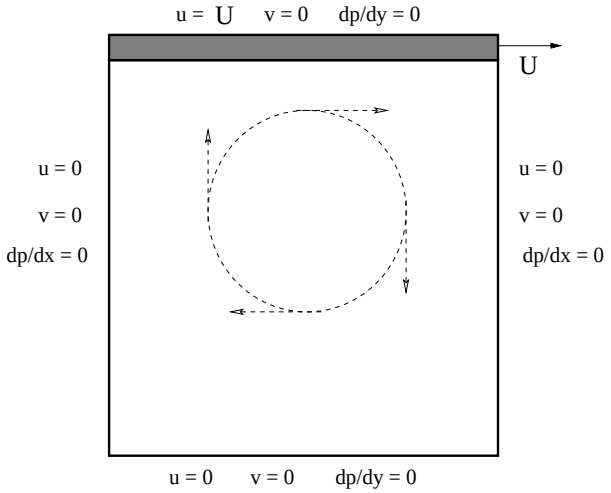

Figure 2: Lid driven cavity initial problem representation (lid is moving with a constant velocity U).

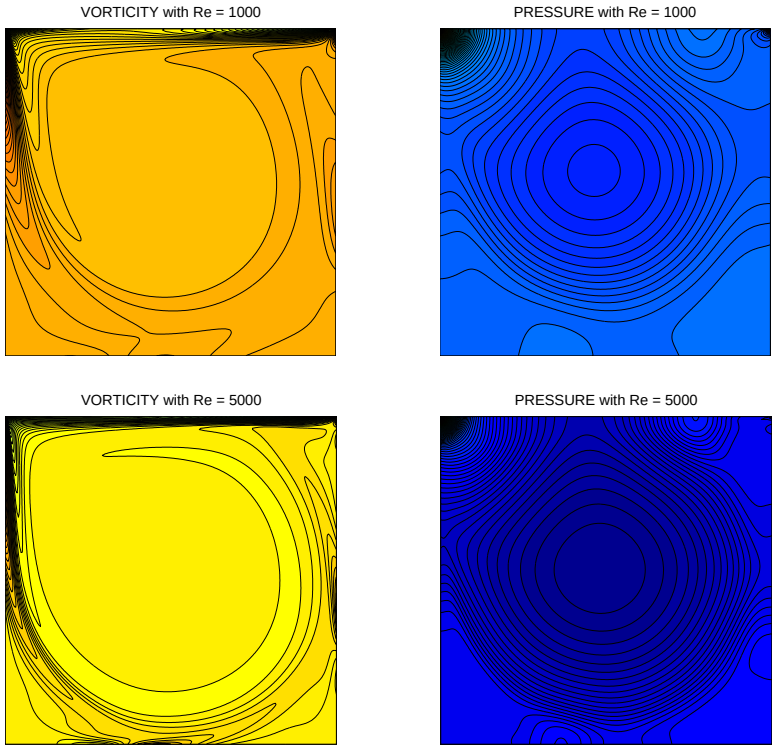

Figure 3: Steady state solutions of the lid driven cavity problem on 256×256 mesh and for two different Reynolds numbers.

boundary conditions). Similarly, the X-velocity is set to be *zero* on all walls except the top wall where it is set to be equal to the lid velocity. We use the homogeneous Neumann boundary conditions for the pressure on all walls. Fig. 2 also illustrates the boundary condition settings. This problem is numerically difficult to tackle, because there is a jump in X-velocity. Therefore, not only we will be testing the accuracy but also we will be testing the stability and the robustness of our numerical method.

Fig. 3 shows the steady state solutions for the vorticity and pressure fields. These solutions are computed on 256×256 mesh. The results at the top half corresponds to the solutions for $Re=1000$ and the bottom half corresponds to the solutions for $Re=5000$. Notice that the vorticity roll ups are more apparent for higher Reynolds numbers. We note that our results are calculated on a relatively coarser mesh compare to [9]. However, the resolutions of the flow structures are comparable. Fig. 4 shows our time convergence study for the X/Y-velocities and pressure. To measure the time convergence rates, we run the code with a fixed mesh (e.g, 256×256) and different time step refinements to a final time (e.g, $t=0.5$ at which the flow is still transient). Then we measure the L_2 norm of errors between *two* consecutive time step refinements and plot the rate of decrease in these errors. The calculations in the convergence study are carried out for Re=1000. Fig. 4

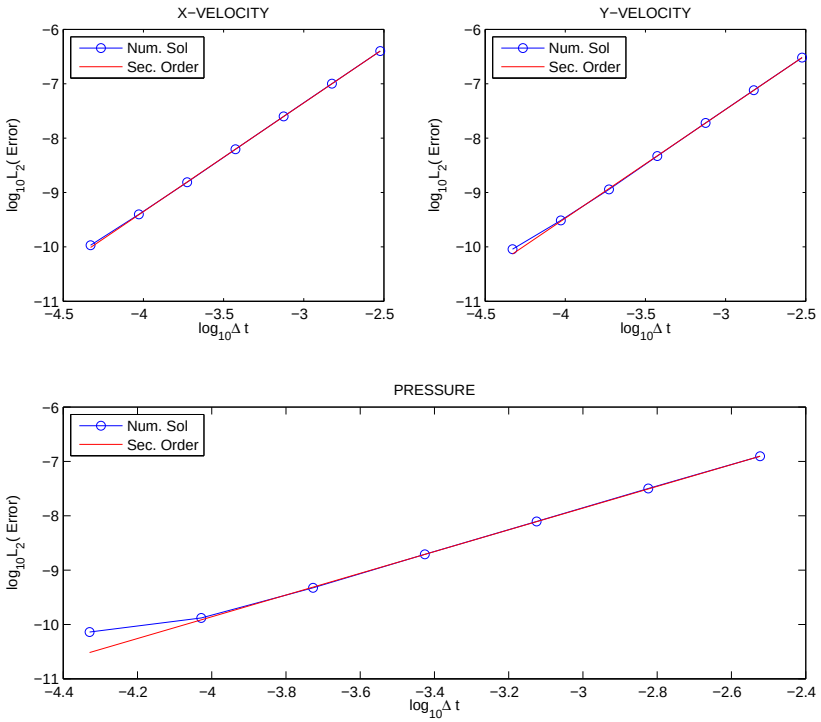

Figure 4: L_2 norm of the time errors of the numerical solutions of the lid driven cavity problem on 256×256 mesh at $time=0.5$ and with $Re=1000$.

clearly shows the second order time convergence for all flow variables. An observation that can be made about the pressure error plot in Fig. 4 is that the error tends not to be second order for the last two time step refinements. This is because for the last two refinements the truncation term is dominated by the spatial errors. And these spatial errors will be maintained at this level of time step refinement unless we refine the mesh.

4.3 Thin shear layer problem

This problem is solved in [7] to study the convergence rates of their second order projection method applied to the incompressible Navier-Stokes equations. The problem consists of a pair of horizontal shear layers of finite thickness, perturbed by a small amplitude vertical velocity. The initial problem setting is as follows

$$u(x,y,0)=\begin{cases}\tanh\left(\frac{y-0.25}{\rho}\right), & \text{for } y\leq 0.5,\\ \tanh\left(\frac{0.75-y}{\rho}\right), & \text{for } y>0.5,\end{cases} \tag{4.3}$$

$$v(x,y,0)=\delta\sin(2\pi x), \tag{4.4}$$

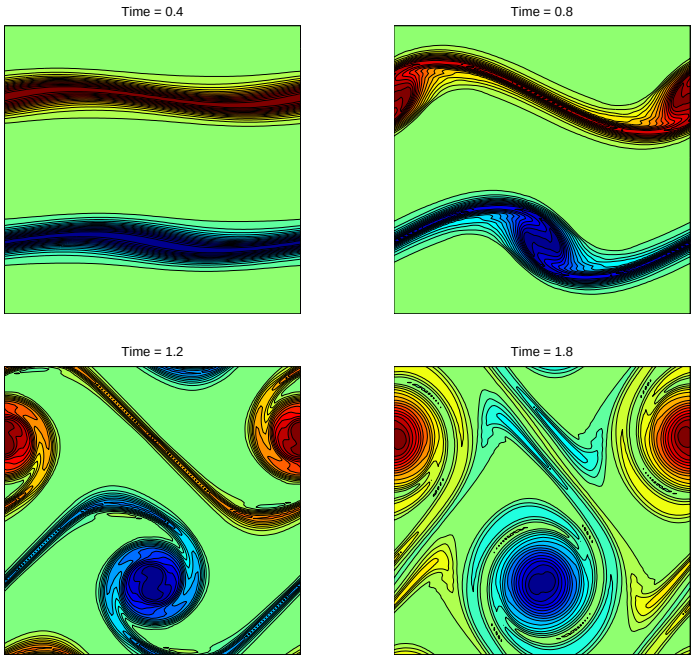

Figure 5: Vorticity solutions from the thin shear layer problem on 128×128 mesh.

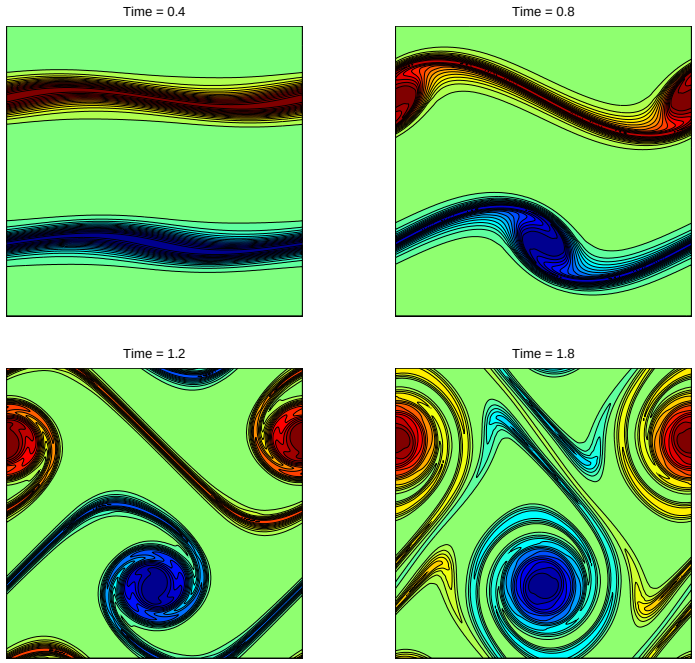

Figure 6: Vorticity solutions from the thin shear layer problem on 256×256 mesh.

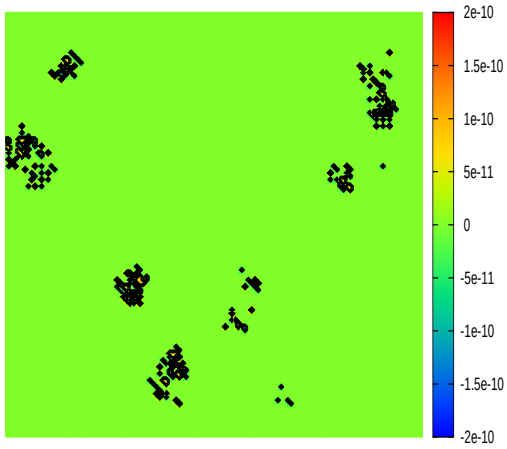

Figure 7: Divergence of the velocity field from the thin shear layer problem on 128×128 mesh.

where ρ determines the width of the shear layer and δ is the strength of the initial perturbation. To be consistent with [7], we set $\rho=1/30$ and $\delta=0.05$ in our computations. Fig. 5 shows the time history of the numerical solutions calculated in a doubly-periodic unit square with 128×128 mesh. In time, each shear layer evolves into a periodic array of large vortices with the shear layer between the rolls being thinned by the large straining field there. Eventually, these thinned layers wrap around the large rolls. The evolution of the top and bottom layers are mirror images of one another. Solutions in Fig. 5 are computed on a rather coarse mesh resulting in small wiggles around the arms of the vortices. However, these wiggles are completely eliminated by refining the mesh (e.g, refer to Fig. 6 that represents the numerical results computed on a 256×256 mesh). Fig. 6 also indicates the spatial convergence of our method. Fig. 7 shows the contours of the divergence of the velocity field on a 128×128 mesh around $t=1.0$. We can see from this figure that the divergence-free constraint is sufficiently satisfied even on a relatively coarse mesh. We note that the flow structures (vorticity roll ups) in [7] seem to be better resolved compared to ours. We attribute this to the dissipativeness of our numerical scheme which is originating from the corner value evaluation (refer to Section 3.1). We remark that it is not our goal to introduce a high resolution scheme in this paper. Rather, our aim is to introduce a self-consistent implementation of an IMEX strategy that leads to demonstrated second order time convergence together with effective flow calculations.

Finally, Fig. 8 shows our time convergence study for the X/Y-velocities and pressure. We carried out our computations on a 256×256 mesh until the final *time* = 1.0 to produce this figure. Notice that the flow is highly transient around this time (refer to Fig. 6). Fig. 8 shows the decrease in the L_2-norm of the time errors of the numerical solutions. The time errors are measured by the same way as in Section 4.2. It is clear from Fig. 8 that we obtain second order time convergence for each flow variable.

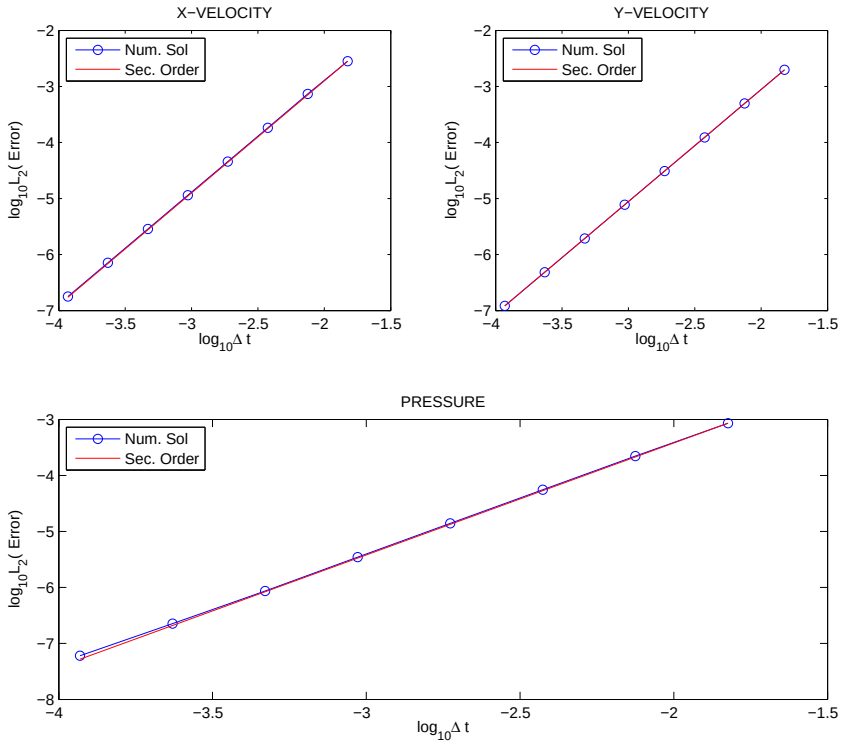

Figure 8: L_2 norm of the time errors of the numerical solutions of the thin shear layer problem on 256×256 mesh at $time=1.0$.

4.4 Solver/preconditioner performance

In this section, we present the nonlinear solver performance analysis of our method. This analysis will gives us important insights about our preconditioning procedure and solver technology. For this analysis, we consider the thin shear layer problem.

First we consider the Table 4. Table 4 is created by calculations on a 64×64 mesh until the final $time=0.5$. Table 4 compares the solver performance when we use the velocity preconditioner with or without viscosity effects ((3.30)-(3.31) versus (3.32)-(3.33)). The pressure-Poisson problem is relaxed by an algebraic multi-grid method for both options. The multi-grid solver uses only *one* V-cycle each time it is called. Table 4 compares the average number of GMRES steps per Newton iteration and the average number of Newton iterations per time step for the two velocity preconditioner options. The Newton iteration is terminated when (3.20) is satisfied for $tol_{res}=10^{-8}$. The GMRES iteration is terminated when (3.23) is satisfied for $\gamma=10^{-3}$. Clearly, for low Reynolds numbers (viscous flow), the velocity preconditioner with the viscosity effects produces more effective calculations (about *twice* more effective) than the other option. This comparison is in terms of GMRES iterations not the CPU time. The reduced GMRES iterations come at the cost of a more expansive preconditioner. We note that we used a fixed time step ($\Delta t=10^{-3}$) to

Table 4: Comparing the velocity preconditioner with or without viscosity effects for different Reynolds numbers.

Reynolds Number	Preconditioning with viscosity		Preconditioning without viscosity	
	GMRES steps	Newton steps	GMRES steps	Newton steps
$Re=0.5$	32	4	80	6
$Re=1.0$	28	3	57	4
$Re=10.0$	10	3	12	3
$Re=100.0$	4	3	5	3

Table 5: Comparing the velocity preconditioner with viscosity effects to the classic projection method for different Reynolds numbers.

Reynolds Number	Preconditioning with viscosity		Classic projection method	
	GMRES steps	Newton steps	GMRES steps	Newton steps
$Re=0.5$	32	4	24	4
$Re=1.0$	28	3	19	3
$Re=10.0$	10	3	7	3
$Re=100.0$	4	3	3	3

Table 6: Comparing multi-grid pressure solve versus Gauss-Seidel sweeps. Velocity preconditioner doesn't use viscosity effects.

Mesh Refinement	Multi-grid pressure solve		Gauss-Seidel pressure solve	
	GMRES steps	Newton steps	GMRES steps	Newton steps
64×64	6	3	17	3
128×128	9	3	44	3
256×256	13	3	95	4

produce this table. For small $Re=0.5$, the viscous time steps scale like $\Delta t/(Re\Delta x^2)\simeq 10$. Considering the average number of nonlinear and linear iterations, we can state that we comfortably step over the viscous time steps by bringing the viscosity effects in the velocity preconditioning. However, when we increase the Reynolds number, the difference for the performance of both options is almost negligible. This is because the fixed time step we took falls within the range of viscous time scale (e.g, $\Delta t=10^{-3}<Re\Delta x^2\simeq 2\times10^{-2}$ when $Re=100$).

In Table 5, we compare the solver performance when using the preconditioning procedure with viscosity effects ((3.32)-(3.33) together with (3.29)) versus the classical second order fractional step projection method. Again we run the code until the final $time=0.5$ on a 64×64 mesh with a fixed time step $\Delta t=10^{-3}$. The Newton and GMRES tolerances are same as above. The pressure Poisson problem is solved by an algebraic multi-grid with *one* V-cycle for both options. On the other hand, (3.34) is solved by *ten* forward-backward conventional Gauss-Seidel sweeps. Comparing the average number of Newton and GMRES iterations from this table, we can conclude that there is not much advantage of one method to other especially with the increasing Reynolds numbers.

Next, Table 6 compares the solver performance for two different ways, multi-grid and

Gauss-Seidel sweeps, to solve the pressure-Poisson problem (e.g, solving (3.29)). We run the code until the final *time* $=0.5$ for *three* different mesh options. The Reynolds number is set to be 10000, thus we don't use viscosity effects for the velocity preconditioning (we use (3.30) and (3.31)). The first pressure-Poisson solver uses an algebraic multi-grid with *one* V-cycle and the second option uses *ten* forward-backward conventional Gauss-Seidel sweeps. Table 6 compares the average number of GMRES steps per Newton iteration and the average number of Newton iterations per time step for the two options. The Newton and GMRES tolerances are same as in the second paragraph of this section. Table 6 indicates that the multi-grid option as a pressure-Poisson solver is more effective (especially for finer grids) than the conventional Gauss-Seidel sweeps. We note that we also compared multi-grid *one* V-cycle versus *two* and *three* V-cycles for the finer mesh (256×256), and we have not observed significant difference in terms of the average number of GMRES iterations.

5 Conclusion

We have presented a second order self-consistent IMEX method for solving the incompressible Navier-Stokes equations. The key for successful implementation of the self-consistent IMEX strategy is to carry out the explicit integrations as part of the non-linear function evaluation within the implicit block. This way, the improved time accuracy of the non-linear iterations (implicit block) is readily felt by the explicit block and vice versa. This continuous interaction between the two algorithm blocks implicitly balances the nonlinear coupling terms so that it does not require extra efforts (outer iterations) to converge nonlinearities. This will be important especially when we solve multi-phase flow problems. We have solved several test problems to check/verify the numerical accuracy and the performance of the new preconditioner. For each test, we have established second order time convergence and obtained reasonable nonlinear solver performance with the new preconditioner. Thus far, we have applied this IMEX methodology to the single phase incompressible flow equations, radiation hydrodynamical systems ranging from low energy density to high energy density limits [22, 24], and multi-physics problems [25]. Currently, we are applying this methodology to multi-phase flow systems in a way that the fluid interfaces are explicitly tracked as part of the non-linear function evaluation within the implicit flow solver [26].

Acknowledgments

The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DEAC07-05ID14517. Accordingly, the U.S. 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 U.S. Government purposes.

Appendix

A.1 Second order interpolations

In this section, we describe how to calculate the *left/right* values of the velocity components needed by the Riemann solver in Section 3.1. The procedure is based on a Taylor series expansion in that we keep the first order terms, i.e,

$$u^L_{i,j} = u_{i-1/2,j} + \frac{\Delta x}{2} u_{x,i-1/2,j}\,, \tag{A.1}$$

$$u^R_{i,j} = u_{i+1/2,j} - \frac{\Delta x}{2} u_{x,i+1/2,j}\,, \tag{A.2}$$

$$u_{x,i-1/2,j} = minmod(a,b) = \begin{cases} a, & \text{if } |a|<|b| \text{ and } ab>0, \\ b, & \text{if } |b|<|a| \text{ and } ab>0, \\ 0, & \text{if } ab\leq 0, \end{cases} \tag{A.3}$$

where

$$a = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x}, \tag{A.4}$$

$$b = \frac{u_{i-1/2,j} - u_{i-3/2,j}}{\Delta x}, \tag{A.5}$$

The terms related to the other velocity component, $v^L_{i,j}$ and $v^R_{i,j}$, can be computed similarly. For more information regarding different slope options etc. we refer to [33, 48].

A.2 Third order WENO interpolations

In this section, we present the third order WENO-type interpolation to built $u^L_{i,j}$. The procedure to construct $u^R_{i,j}$ is the mirror image of $u^L_{i,j}$. The main idea is to construct a third order polynomial by the weighted average of *two* second order polynomials. In particular,

$$u^L_{i,j} = w_1 u^{(1)}_{i,j} + w_2 u^{(2)}_{i,j}\,, \tag{A.6}$$

where

$$u^{(1)}_{i,j} = \frac{3}{2} u_{i-1/2,j} - \frac{1}{2} u_{i-3/2,j}\,, \tag{A.7}$$

$$u^{(2)}_{i,j} = \frac{1}{2} u_{i+1/2,j} + \frac{1}{2} u_{i-1/2,j}\,, \tag{A.8}$$

and the nonlinear weights, w's, are defined by

$$w_1 = \frac{\widetilde{w}_1}{\widetilde{w}_1 + \widetilde{w}_2}, \qquad \widetilde{w}_1 = \frac{\gamma_1}{(\epsilon+\beta_1)^2}, \tag{A.9a}$$

$$w_2 = \frac{\widetilde{w}_2}{\widetilde{w}_1 + \widetilde{w}_2}, \qquad \widetilde{w}_2 = \frac{\gamma_2}{(\epsilon+\beta_2)^2}, \tag{A.9b}$$

where the constants, $\gamma_1 = 1/4$ and $\gamma_2 = 3/4$, are referred to as the linear weights, ϵ is a small positive number used to avoid the denominator becoming *zero* and is typically set to be $\epsilon = 10^{-6}$, and finally β's are the smoothness indicators given by

$$\beta_1 = u^2_{i-3/2,j} - 2u_{i-3/2,j}u_{i-1/2,j} + u^2_{i-1/2,j}, \tag{A.10}$$

$$\beta_2 = u^2_{i-1/2,j} - 2u_{i-1/2,j}u_{i+1/2,j} + u^2_{i+1/2,j}. \tag{A.11}$$

Details regarding the intermediate steps and the construction of higher order interpolations can be found in [41].

References

[1] S. W. Armfield. Finite difference solutions of the Navier-Stokes equations on staggered and non-staggered grids. Computers and Fluids, 20:1–17, 1991.

[2] S. W. Armfield and R. Street. Comparison of staggered and non-staggered grid Navier-Stokes solutions for the 8:1 cavity natural convection flow. ANZIAM, 46:918–934, 2004.

[3] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit runge-kutta methods for time dependent partial differential equations. Applied Numerical Mathematics, 25:151–167, 1997.

[4] U. M. Ascher, S. J. Ruuth, and B. Wetton. Implicit-explicit methods for time dependent pde's. SIAM J. Numer. Anal., 32:797–823, 1995.

[5] G.K. Batchelor. An Introduction to Fluid Dynamics. Cambridge University Press, 1967.

[6] J. W. Bates, D. A. Knoll, W. J. Rider, R. B. Lowrie, and V. A. Mousseau. On consistent time-integration methods for radiation hydrodynamics in the equilibrium diffusion limit: Low energy-density regime. J. Comput. Phys., 167:99–130, 2001.

[7] J.B. Bell, P. Colella, and H.M. Glaz. A second order projection method for the incompressible navier-stokes equations. J. Comput. Phys., 85:257–283, 1989.

[8] P.N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. SIAM J. Sci. Stat. Comput., 11:450–481, 1990.

[9] C.H. Bruneau and M. Saad. The 2-D lid-driven cavity problem revisited. Computers and Fluids, 35:326–348, 2006.

[10] A. Chorin and J.E. Marsden. A Mathematical Introduction to Fluid Mechanics. Springer-Verlag, 1979.

[11] A. J. Chorin. Numerical solution of the Navier-Stokes equations. Math. Comp., 22:742–762, 1968.

[12] D. Drikakis. Study of bifurcation flow phenomena in incompressible sudden expansion flows. Physics of Fluids, 9:76–87, 1997.

[13] D. Drikakis and W. Rider. High-Resolution Methods for Incompressible and Low-Speed Flows. Springer., 2005.

[14] R. Dembo et al. Inexact newton methods. SIAM J. Num. Anal., 19:400–408, 1982.

[15] C.A.J. Fletcher. Computational Techniques for Fluid Dynamics, Volume 2. Springer.

[16] S. Gottlieb and C. W. Shu. Total variation diminishing runge-kutta schemes. Mathematics of Computation, 221:73–85, 1998.

[17] S. Gottlieb, C. W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. Siam Review, 43-1:89–112, 2001.

[18] B.E. Griffith. An accurate and efficient method for the incompressible navier-stokes equations using the projection method as a preconditioner. J. Comput. Phys., 228:7565–7595, 2009.

[19] J. L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. Comput. Methods Appl. Mech. Engrg., 195:6011–6045, 2006.
[20] B. Gustafson, H. O. Kreiss, and J. Oliger. Time dependent problems and difference methods. Wiley, New York, 1995.
[21] F.H. Harlow and J.E. Welch. Numerical calculation of time dependent viscous incompressible flow of fluid with a free surface. Phys. Fluids, 8:2182, 1965.
[22] S. Y. Kadioglu and D. A. Knoll. A fully second order implicit/explicit time integration technique for hydrodynamics plus nonlinear heat conduction problems. J. Comput. Phys., 229-9:3237–3249, 2010.
[23] S. Y. Kadioglu and D. A. Knoll. Hydrodynamics (Advanced Topics). INTECH, 2011.
[24] S. Y. Kadioglu, D. A. Knoll, Robert B. Lowrie, and Rick M. Rauenzahn. A second order self-consistent IMEX method for radiation hydrodynamics. J. Comput. Phys., 229-22:8313–8332, 2010.
[25] S. Y. Kadioglu, D. A. Knoll, and C. Oliveria. Multi-physics analysis of spherical fast burst reactors. Nuclear Science and Engineering, 163:1–12, 2009.
[26] S. Y. Kadioglu, D. A. Knoll, and Mark Sussman. An IMEX method for multi-phase flows. In progress, 2011.
[27] S.Y. Kadioglu, D.A. Knoll, M. Sussman, and R. Martineau. A second order JFNK-based IMEX method for single and multi-phase flows. Computational Fluid Dynamics, Springer-Verlag, DOI 10.1007/978−3−642−17884−9_69, 2010.
[28] S.Y. Kadioglu, M. Sussman, S. Osher, J.P. Wright, and M. Kang. A Second Order Primitive Preconditioner For Solving All Speed Multi-Phase Flows. J. Comput. Phys., 209-2:477–503, 2005.
[29] C. T. Kelley. Solving Nonlinear Equations with Newton's Method. Siam, 2003.
[30] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. J. Comput. Phys., 59:308–323, 1985.
[31] D. A. Knoll and D. E. Keyes. Jacobian-free Newton Krylov methods: a survey of approaches and applications. J. Comput. Phys., 193:357–397, 2004.
[32] B. P. Leonard. The ultimate conservative difference scheme applied to unsteady one dimensional advection. Comp. Methods Appl. Mech. Eng, 88, 1991.
[33] R. J. Leveque. Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Texts in Applied Mathematics, 1998.
[34] X.D Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. J. Comput. Phys., 115:200–212, 1994.
[35] R. B. Lowrie, J. E. Morel, and J. A. Hittinger. The coupling of radiation and hydrodynamics. Astrophys. J., 521:432, 1999.
[36] S.V. Patankar. Numerical heat transfer and fluid flow. Hemisphere, New York, 1980.
[37] J. K. Reid. On the methods of conjugate gradients for the solution of large sparse systems of linear equations. Large Sparse Sets of Linear Equations, Academic Press, New York, 1971.
[38] S. J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. J. Math. Biol., 34:148–176, 1995.
[39] Y. Saad. Iterative Methods for Sparse Linear Systems. Siam, 2003.
[40] R. Schreiber and H.B. Keller. Driven cavity flows by efficient numerical techniques. J. Comput. Phys., 49, 1983.
[41] C.W. Shu. High order weighted essentially non-oscillatory schemes for convection dominated problems. SIAM Review, 51:82–126, 2009.
[42] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock captur-

ing schemes. J. Comput. Phys., 77:439, 1988.
[43] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II. J. Comput. Phys., 83:32, 1989.
[44] J. C. Strikwerda. Finite Difference Schemes Partial Differential Equations. Wadsworth & Brooks/Cole, Advance Books & Software, Pacific Grove, California, 1989.
[45] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. J. Comput. Phys., 187:110–136, 2003.
[46] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and Axisymmetric incompressible two-phase flows. J. Comput. Phys., 162:301–337, 2000.
[47] J.W. Thomas. Numerical Partial Differential Equations I (Finite Difference Methods). Springer-Verlag New York, Texts in Applied Mathematics, 1998.
[48] J.W. Thomas. Numerical Partial Differential Equations II (Conservation Laws and Elliptic Equations). Springer-Verlag New York, Texts in Applied Mathematics, 1999.
[49] P. Wesseling. Principles of Computational Fluid Dynamics. Springer Series in Computational Mathematics, 2000.