Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 203 (2005) 517–538

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Stability of approximate projection methods on cell-centered grids

Robert D. Guy, Aaron L. Fogelson *

Department of Mathematics, University of Utah, 155 South 1400 East, John Widstoe Building, Room JWB 233, Salt Lake City, UT 84112, USA

Received 19 December 2003; received in revised form 24 May 2004; accepted 10 September 2004
Available online 14 October 2004

Abstract

Projection methods are a popular class of methods for solving the incompressible Navier–Stokes equations. If a cell-centered grid is chosen, in order to use high-resolution methods for the advection terms, performing the projection exactly is problematic. An attractive alternative is to use an approximate projection, in which the velocity is required to be only approximately discretely divergence-free. The stability of the cell-centered, approximate projection is highly sensitive to the method used to update the pressure and compute the pressure gradient. This is demonstrated by analyzing a model problem and conducting numerical simulations of the Navier–Stokes equations.
© 2004 Elsevier Inc. All rights reserved.

MSC: 65M06; 65M12; 76D05; 76M20
Keywords: Navier–Stokes equations; Incompressible flow; Projection methods; Approximate projections; Cell-centered discretization; Stability

1. Introduction

This paper considers the stability of approximate projection methods for solving the incompressible Navier–Stokes equations discretized using a cell-centered grid. The non-dimensional form of the equations considered is

$$\boldsymbol{u}_t + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \nu \Delta \boldsymbol{u} + \boldsymbol{f}, \tag{1}$$

* Corresponding author. Tel.: +1 801 581 8150; fax: +1 801 581 4148.
E-mail addresses: guy@math.utah.edu (R.D. Guy), fogelson@math.utah.edu (A.L. Fogelson).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.09.005

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where the non-dimensional viscosity v is equal to Re^{-1}. Of course suitable boundary conditions (for example, $\boldsymbol{u} = \boldsymbol{0}$ on the boundary) must also be prescribed. If one were to naively begin to design a numerical scheme, several challenges would quickly become apparent. Because of the advection terms, the equations are nonlinear. The nonlinearity does complicate the construction of a numerical scheme, but this complication can be handled in a variety of ways, which are discussed later. Another difficulty, which is more specific to these equations, comes from the lack of an evolution equation for the pressure. In incompressible flow the pressure does not evolve, rather, its value is determined to enforce the incompressibility constraint. This idea is the basis of a class of methods called *projection methods*.

The first projection method was developed by Chorin [1,2]. The basic idea behind the method is to use the momentum equation to solve for an intermediate velocity that is not required to be divergence-free. Then the intermediate velocity is projected to yield a discretely divergence-free velocity and a gradient field. The latter can then be used to update the pressure. The nonlinear advection terms can be extrapolated in time, leaving a linear equation to solve for the intermediate velocity. One popular method of extrapolating the nonlinear terms, particularly useful at high Reynolds numbers, uses a high-resolution Godunov method, which require a cell-centered spatial discretization [3]. However, the use of a cell-centered grid complicates the use of mesh refinement and the implementation of a multigrid solver in the projection step of the method [4]. These complications can be avoided if the constraint that the velocity be discretely divergence-free is relaxed so that the discrete velocity is required to be only approximately divergence-free. Such methods are referred to as approximate projection methods [5].

Projection methods and approximate projection methods involve solving for intermediate quantities which are then used to compute the physical quantities, the velocity field and the pressure. Because the intermediate quantities are not physical quantities, there has been much confusion related to the boundary conditions. The proper update for the pressure is another source of ambiguity in projection methods. A recent paper by Brown et al. [6] explored and clarified the role of boundary conditions and different pressure updates in projection methods. This paper also proposed various methods for obtaining a pressure that is globally second-order accurate, a result some have speculated to be unattainable with a projection method [7,8]. The results in [6] followed from the analysis of a system with discrete time but continuous spatial variables. The results of the analysis are supported by numerical tests that verify the second-order accurate pressure. We note that the numerical tests were performed using an approximate projection method on a node-centered grid, rather than the more commonly used cell-centered grid. The behavior of the solution in approximate projection methods has been shown to be sensitive to the grid structure [9].

We extend the work of [6] by taking into account the discretization of space. Our analysis shows that the system of equations becomes very sensitive to the spatial discretization if an approximate projection method is used in combination with the method of obtaining a second-order accurate pressure discussed in [6]. We found instabilities in testing this method with a standard second-order discretization on a cell-centered grid. The nature of the instability is analyzed using a model problem, and numerical results are presented for the model problem and for the Navier–Stokes equations. These computational tests demonstrate the instability and identify several methods that are stable and achieve full second-order accuracy.

There have been many analyses of the accuracy and stability properties of projection methods, beginning with Chorin [2], whose analysis was limited to the case of periodic boundary conditions. There have more recent studies that included the effect of boundaries. Many of these studies considered systems with discrete time and continuous space [7,10–14], and others included the spatial discretization in the analysis [15,16]. The properties of approximate projection methods are not very well understood, since only a few analyses have been performed [9,17]. Numerical results from [9] demonstrate that it is important to consider the spatial discretization in an approximate projection method.

In the rest of this paper, we present the motivation for projection methods followed by a discussion of different approaches to implementing them and the accuracy of these implementations. We then analyze the stability of a model problem. In doing so, we consider several different methods for updating the pressure. This analysis is done for the case of an exact projection on a staggered grid and for an approximate projection on a cell-centered grid. Numerical results are presented for the model problem and for the Navier–Stokes equations in two spatial dimensions.

2. Motivation

The following, well-known decomposition theorem helps clarify the role of the pressure in incompressible flow and motivates projection methods [18].

Theorem 1. (Hodge decomposition) *Let Ω be a smooth, bounded domain, and u^* be smooth vector field on Ω. The vector field u^* can be decomposed in the form*

$$\boldsymbol{u}^* = \boldsymbol{u} + \nabla\phi, \tag{3}$$

where

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \qquad \boldsymbol{u}\cdot\boldsymbol{n} = 0 \quad \text{on } \partial\Omega.$$

We present the elementary proof of the theorem because it is constructive, and thus provides a method for performing this decomposition that can be used to implement a projection method.

Proof. Taking the divergence of (3) gives the Poisson equation

$$\Delta\phi = \nabla\cdot\boldsymbol{u}^*, \tag{4}$$

with the boundary condition given by the normal component of (3) on the boundary

$$\frac{\partial\phi}{\partial n} = \boldsymbol{u}^*\cdot\boldsymbol{n} \quad \text{on } \partial\Omega. \tag{5}$$

Together (4) and (5) define a Neumann problem for ϕ, which has a unique solution, up to an additive constant, provided the solvability condition

$$\int_\Omega \nabla\cdot\boldsymbol{u}^*\,\mathrm{d}V = \int_{\partial\Omega} \boldsymbol{u}^*\cdot\boldsymbol{n}\,\mathrm{d}S \tag{6}$$

is satisfied. Eq. (6) is always satisfied, because this is simply the divergence theorem. This defines the gradient part of the decomposition and the divergence-free field is then defined by $u = u^* - \nabla\phi$. □

The preceding proof describes a procedure for decomposing an arbitrary smooth vector field into a gradient field and a divergence-free field. Let $\mathscr{P}$ denote the operator which takes vector fields to divergence-free vector fields, as in the decomposition theorem. Then $\mathscr{P}$ is a projection operator (it is linear and $\mathscr{P}^2 = \mathscr{P}$), and

$$\mathscr{P}(\boldsymbol{u}^*) = \boldsymbol{u}$$

and

$$(I - \mathscr{P})(\boldsymbol{u}^*) = \nabla\phi.$$

Note that gradient fields and divergence-free fields that have zero normal component on the boundary define orthogonal subspaces of $L_2(\Omega)$. The requirement that $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ on the boundary may be relaxed so that $\boldsymbol{u}\cdot\boldsymbol{n} = g$, provided g satisfies

$$\int_{\partial\Omega} g\,\mathrm{d}S = 0. \tag{7}$$

If $\boldsymbol{u}$ is interpreted as a velocity field, then the condition (7) physically means that the total flux of fluid across the boundary of the domain is zero, which is a consequence of incompressibility. While the decomposition holds in the case $g \neq 0$, the resulting operator $\mathscr{P}$ is no longer a projection. However, numerical methods that employ this decomposition with non-homogeneous boundary conditions are still referred to as projection methods. The decomposition also may be applied to periodic velocity fields, with an obvious modification.

Applying the projection operator to (1) gives the equation

$$\boldsymbol{u}_t = \mathscr{P}(-\boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nu \Delta \boldsymbol{u} + \boldsymbol{f}), \tag{8}$$

in which the pressure has been eliminated. This form is equivalent to the original equations (1) and (2). Note that the operator $\mathscr{P}$ constrains only the normal component of the velocity and additional boundary conditions are required (for example $\boldsymbol{u} \cdot \boldsymbol{\tau} = 0$) to complete the statement of the problem. The equivalence of these two forms of the equations clarifies the role of the pressure in enforcing incompressibility. The proof of the Hodge decomposition is suggestive of a scheme for advancing the velocity and pressure in time: advance the momentum equation (1) to determine an intermediate velocity which is not required to be divergence-free. Then to find the velocity at the new time, project the intermediate velocity field onto the space of divergence-free fields, and use the gradient part of the projection to update the pressure.

3. Implementing accurate projection methods

Chorin's original projection method [1,2] is first-order accurate in space and time. The next generation of projection methods used various means to obtain a velocity which is second-order accurate in space and time, but a pressure that is only first-order accurate [3,19]. In the same time period, Kim and Moin [20] proposed a method in which the pressure is excluded from the computation, but with a modification to the boundary conditions, they obtain a second-order accurate velocity field. Interestingly, they do mention how the pressure can be recovered from their computation, and this pressure turns out to be second-order accurate, provided the result is interpreted at the proper temporal location [6].

Other variations of the original projection method have been utilized. One such modification which we will discuss in detail is the approximate projection method [5,21]. The idea of the approximate projection method is to relax the condition that the velocity field be discretely divergence-free, rather the velocity field that results is approximately divergence-free. The utility of such methods is discussed below when we consider the different spatial discretizations.

3.1. Temporal discretization

A second-order time discretization of the momentum equation (1) is

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + \boldsymbol{g}^{n+1/2} = -\nabla p^{n+1/2} + \frac{\nu}{2}(\Delta \boldsymbol{u}^n + \Delta \boldsymbol{u}^{n+1}), \tag{9}$$

where $\boldsymbol{g}^{n+1/2}$ represents some approximation to the nonlinear, advection terms at the half time level, and we have set the external forces to zero for convenience. We discuss the form of $\boldsymbol{g}^{n+1/2}$ in a later section. Let q be some approximation for the pressure that can be computed from the variables at the current time level. The projection method seeks a solution to (9), by first solving the equation

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + \boldsymbol{g}^{n+1/2} = -\nabla q + \frac{v}{2}(\Delta \boldsymbol{u}^n + \Delta \boldsymbol{u}^*) \tag{10}$$

for an intermediate velocity $\boldsymbol{u}$*. The velocity $\boldsymbol{u}$* is a solution to the discrete momentum equation, but it is not divergence-free. Using the Hodge decomposition theorem, we know that $\boldsymbol{u}$* can be decomposed as

$$\boldsymbol{u}^* = \boldsymbol{u}^{n+1} + \Delta t \nabla \phi, \tag{11}$$

where $\boldsymbol{u}^{n+1}$ is divergence-free. The factor of Δt is not required, but its relevance is clarified in the discussion of the pressure. Proceeding as in the proof of the decomposition, the divergence of this equation and the normal component on the boundary give the Poisson problem

$$\Delta t \Delta \phi = \nabla \cdot \boldsymbol{u}^*, \tag{12}$$

with Neumann boundary conditions

$$\Delta t \frac{\partial \phi}{\partial n} = (\boldsymbol{u}^* - \boldsymbol{u}^{n+1}) \cdot \boldsymbol{n}. \tag{13}$$

Once ϕ is known, the divergence-free velocity is then found by computing

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \Delta t \nabla \phi. \tag{14}$$

Note that there are no prescribed boundary conditions for the intermediate velocity $\boldsymbol{u}$*. Often the boundary conditions for $\boldsymbol{u}^{n+1}$ are used for $\boldsymbol{u}$*. A thorough discussion on boundary conditions can be found in [6].

We now discuss the the computation of the pressure. The variable q in (10) represents some approximation to the pressure at time level $n + 1/2$. Some choices for q that have been used in methods that produce a second-order accurate approximation to the velocity are $q = 0$ [20], $q = p^n$ [19], and $q = p^{n-1/2}$ [3]. We consider the case $q = p^{n-1/2}$; the formulas for the other cases can be derived by straightforward modifications to the following arguments. Eq. (9) can be rearranged as

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + \nabla p^{n+1/2} = -\boldsymbol{g}^{n+1/2} + \frac{v}{2}(\Delta \boldsymbol{u}^n + \Delta \boldsymbol{u}^{n+1}) \tag{15}$$

so that the terms on the left-hand side resemble a Hodge decomposition of the terms on the right-hand side into a divergence-free field and a gradient field. Writing Eqs. (10) and (11) as

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + \nabla p^{n-1/2} = -\boldsymbol{g}^{n+1/2} + \frac{v}{2}(\Delta \boldsymbol{u}^n + \Delta \boldsymbol{u}^*), \tag{16}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} + \nabla \phi = 0 \tag{17}$$

shows that one way of conceptualizing the projection method is a splitting of the time derivative $\boldsymbol{u}_t$ and the pressure into two steps using the intermediate variables $\boldsymbol{u}$* and ϕ. This splitting reveals that ϕ is related to the pressure as the pressure update

$$p^{n+1/2} = p^{n-1/2} + \phi. \tag{18}$$

This is the traditional method used to update the pressure, but it is well known that the resulting pressure is only first-order accurate due to a numerical boundary layer; for analysis see for example [6,7,14]. This error does not degrade the accuracy of the velocity, and so the resulting scheme gives a pressure that is first-order accurate and a second-order accurate velocity.

A method for obtaining a second-order accurate pressure is proposed in [6]. Following their derivation of a pressure update formula, use (11) to eliminate the intermediate velocity in Eq. (10) to obtain

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\boldsymbol{g}^{n+1/2}=-\nabla p^{n-1/2}-\nabla\phi+\frac{v\Delta t}{2}\Delta\nabla\phi+\frac{v}{2}(\Delta\boldsymbol{u}^{n}+\Delta\boldsymbol{u}^{n+1}). \tag{19}$$

Comparison with (9) suggests defining

$$\nabla p^{n+1/2}=\nabla p^{n-1/2}+\nabla\phi-\frac{v\Delta t}{2}\Delta\nabla\phi. \tag{20}$$

Commuting the Laplacian and the gradient operators gives the pressure update formula

$$p^{n+1/2}=p^{n-1/2}+\phi-\frac{v\Delta t}{2}\Delta\phi. \tag{21}$$

Using (12), this can be written as

$$p^{n+1/2}=p^{n-1/2}+\phi-\frac{v}{2}\nabla\cdot\boldsymbol{u}^{*}. \tag{22}$$

In [6], it is shown that this pressure update eliminates the boundary layer and gives a globally second-order accurate pressure. In Section 4 of this paper, we demonstrate through a model problem that when this pressure update is used in an approximate projection method, the stability of the scheme is not robust.

3.2. Spatial discretization

We discuss regular (equally spaced) finite difference spatial discretizations of a rectangular domain. All of these discretizations partition the domain into some number of equal-sized squares. The discretizations differ in where on the domain the discrete data are located. Two standard discretizations are the vertex (or node)-centered grid and the cell-centered grid. On the vertex-centered grid, all discrete variables are located on the vertices of the squares. For the cell-centered grid, all data are located at the centers of the squares. See Fig. 1 for examples of both grids.

In order to perform a discrete projection, we proceed as in the proof of the Hodge decomposition. Let $\boldsymbol{u}_{i,j}=(u_{i,j},v_{i,j})$ be a discrete velocity at the grid point indexed by (i,j), and let $\phi_{i,j}$ be a discrete scalar located at the same grid point. We use the convention that point i, j corresponds to the cell center, and points such as $i-1/2, j$ correspond to the cell edges (left edge in this example). Let D and G denote the discrete divergence

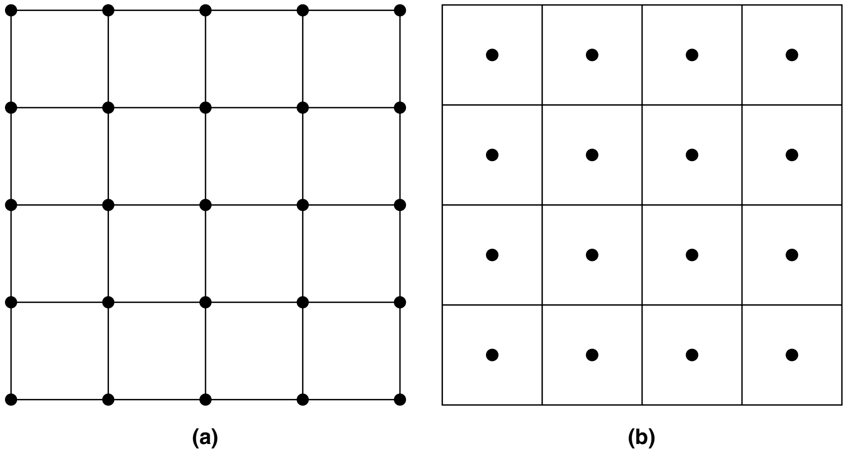

Fig. 1. Examples of (a) vertex-centered grid and (b) cell-centered grid. All data are located at the black dots.

and gradient operators for the cell-centered grid (corresponding operators for the node-centered grid are obtained by an obvious shift of indicies). Second-order accurate operators are

$$(D\boldsymbol{u})_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{23}$$

and

$$(G\phi)_{i,j} = \left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h}\right). \tag{24}$$

To decompose the discrete velocity $\boldsymbol{u}$* into a discretely divergence-free field and a discrete gradient field, we seek $\boldsymbol{u}$ and ϕ such that

$$\boldsymbol{u}^* = \boldsymbol{u} + \Delta t G\phi, \tag{25}$$

$$D\boldsymbol{u} = 0. \tag{26}$$

Applying D to both sides of (25), and using (26), we obtain the equation

$$D\boldsymbol{u}^* = \Delta t DG\phi = \Delta t L^w \phi. \tag{27}$$

This equation can be solved to find ϕ, which then defines $\boldsymbol{u}$ using (25).

The operator L^w represents a discrete approximation to the Laplacian. The superscript w is used to denote the fact that the stencil for the operator is wide. The form of L^w at point i, j is

$$(L^w\phi)_{i,j} = \frac{\phi_{i-2,j} + \phi_{i+2,j} - 4\phi_{i,j} + \phi_{i,j-2} + \phi_{i,j+2}}{4h^2}. \tag{28}$$

The wide stencil for the discrete Laplacian creates some difficulties that are absent when working with the more standard, compact Laplacian

$$(L\phi)_{i,j} = \frac{\phi_{i-1,j} + \phi_{i+1,j} - 4\phi_{i,j} + \phi_{i,j-1} + \phi_{i,j+1}}{h^2}. \tag{29}$$

When solving Eq. (27), the problem decouples into four separate subgrids, on each of which i and j are either always even or always odd, except possibly near the boundaries. This local decoupling must be accounted for when implementing mesh refinement or in designing a multigrid solver [4].

A third grid structure which is commonly used for projection methods is the MAC (or staggered) grid, which was introduced by Harlow and Welch [22]. On the MAC grid, the horizontal velocities are stored at vertical edges, the vertical velocities are stored at horizontal edges, and scalar quantities are stored at cell centers. An example is displayed in Fig. 2. The discrete divergence is

$$(D\boldsymbol{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h}, \tag{30}$$

and the discrete gradient is $G\phi = (G_1\phi, G_2\phi)$, where the first component is defined by

$$(G_1\phi)_{i-1/2,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h} \tag{31}$$

and the second component is defined by

$$(G_2\phi)_{i,j-1/2} = \frac{\phi_{i,j} - \phi_{i,j-1}}{h}. \tag{32}$$

Using this divergence and gradient in Eqs. (25) and (26) gives the equation

$$D\boldsymbol{u}^* = \Delta t DG\phi = \Delta t L\phi. \tag{33}$$

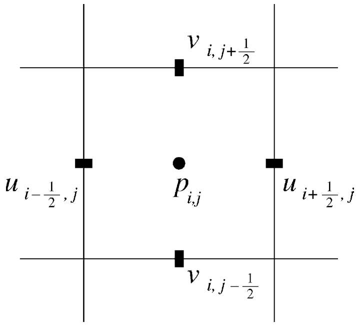

Fig. 2. Example of the MAC grid discretization. Velocity fields are stored at the edges of the cell perpendicular to the component of the velocity, and pressure (and other scalars) are stored at the cell centers.

The operator L is again a discrete approximation to the Laplacian, but it has the compact stencil given in (29). When solving (33), there is no decoupling of grids and handling the boundary conditions is straightforward.

The projection may seem more natural on the MAC grid, but it is sometimes useful to use a vertex- or cell-centered grid. As we discuss later, a popular method for computing the nonlinear terms requires that the velocities be on a cell-centered grid. Either of these other grid structures requires the use of the wide Laplacian to enforce the divergence constraint. If however, the constraint that the velocity be discretely divergence-free is relaxed so that the divergence constraint is only satisfied approximately, the use of the wide Laplacian can be avoided. Such methods are referred to as approximate projections, for obvious reasons. Suppose that a vertex- or cell-centered grid is being used. In the projection step, the compact Laplacian is used in place of the wide Laplacian in solving Eq. (27). The velocity $\boldsymbol{u}$ that results is not discretely divergence-free (with respect to the divergence operator (23)), but its divergence is (h^2) (for sufficiently smooth velocity fields) because this is a second-order discretization of Eq. (4).

3.3. Nonlinear terms

The nonlinear, advective terms can be approximated in a variety of ways. In Chorin's original method [1,2], the momentum equation (10) is solved using an ADI scheme with the nonlinear terms approximated as

$$\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^*. \tag{34}$$

Of course this is only first-order accurate in time and must be modified to get a second-order scheme. Van Kan [19] uses the approximation

$$\tfrac{1}{2}(\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^* + \boldsymbol{u}^* \cdot \nabla \boldsymbol{u}^n), \tag{35}$$

which results in a second-order scheme, and still only requires linear solves.

Another popular method of computing the nonlinear terms is to extrapolate them in time from past velocities. One method of extrapolation used by Kim and Moin [20] is the second-order AB method

$$(\boldsymbol{u} \cdot \nabla \boldsymbol{u})^{n+1/2} = \tfrac{3}{2}(\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n) - \tfrac{1}{2}(\boldsymbol{u}^{n-1} \cdot \nabla \boldsymbol{u}^{n-1}) + \mathcal{O}(\Delta t^2). \tag{36}$$

This approach has been used by others in practice and for analysis [6,8,13,23]. Another very different extrapolation method introduced by Bell et al. [3] and used frequently [4,21,24,25] is particularly useful for high Reynolds number flow. This method is based on high-resolution upwinding methods [26], which require that the cell-centered grid be used. As discussed previously, this grid structure complicates the projection step, and so this situation is ideal for an approximate projection.

A recent variation on the method of using high-resolution upwinding is to employ time splitting to treat the advection separately [27]. In this approach, the cell-centered velocity is advected by a discretely divergence-free edge velocity, $\boldsymbol{u}^{\mathrm{e}}$. At the beginning of the time step, the homogeneous advection equation

$$\boldsymbol{u}_t + \boldsymbol{u}^{\mathrm{e}} \cdot \nabla \boldsymbol{u} = 0, \tag{37}$$

is advanced one time step using the method described in [28]. The resulting cell-centered velocity is then used in place of $\boldsymbol{u}^n$ in the momentum equation to solve for $\boldsymbol{u}^*$. Because the advection is treated separately, this momentum equation is linear. The intermediate velocity is then averaged to the cell edges, and a MAC projection is performed to get the new cell-edge velocity that is discretely divergence-free. The cell-centered velocity is then obtained by averaging the gradient from the MAC projection to the cell centers, and subtracting this average gradient from $\boldsymbol{u}^*$.

3.4. Comparison of methods

A recent paper by Brown et al. [6] removes much of the confusion surrounding the boundary conditions and the pressure in projection methods. In particular, they analyze the accuracy of both the velocity and pressure for various combinations of pressure approximations, pressure update formulas, and boundary conditions on the intermediate velocity. Their analysis is done for a system with discrete time and continuous space. Using an approximate projection method, they generate numerical results that support their analysis. So as long as the spatial operators are discretized to sufficiently high order, the results from [6] predict that full second-order accuracy in the velocity and pressure are attainable.

We briefly summarize their results on the accuracy of projection methods. As discussed previously, let q represent the approximation to the pressure that is used in the momentum equation. The three methods considered are

1. Projection Method I (PM I). The pressure gradient is lagged in the momentum equation, $q = p^{n-1/2}$. The pressure is updated by $p^{n+1/2} = p^{n-1/2} + \phi$.
2. Projection Method II (PM II). The pressure gradient is again lagged in the momentum equation, $q = p^{n-1/2}$, but the pressure is updated by $p^{n+1/2} = p^{n-1/2} + \phi - (v\Delta t/2)\Delta\phi$.
3. Projection Method III (PM III). The pressure gradient is not present in the momentum equation, $q = 0$. The pressure is not updated, rather is is computed by $p^{n+1/2} = \phi - (v\Delta \mathrm{t}/2)\Delta\phi$.

The time accuracy of each method is analyzed using normal mode analysis. Suppose that the intermediate velocity satisfies the boundary conditions that are prescribed for the velocity at the next time step. Then this analysis predicts that PM I gives velocities that are uniformly second-order in time, but the pressure contains a numerical boundary layer making it only globally first-order accurate. PM II is predicted to be globally second-order accurate in both the velocity and the pressure. In order to achieve second-order accuracy for PM III, the tangential component of the intermediate velocity on the boundary must be modified to depend on ϕ from the previous time step.

We have found that projection method II is extremely sensitive to the spatial discretization, becoming unstable with certain discretizations. In this paper, we further explore the stability of the methods analyzed in [6] by considering the spatial discretization. Specifically, we will compare the stability properties of the three projection methods PM I–III using an exact projection on a MAC grid and an approximate projection on a cell-centered grid.

4. Stability analysis

Analyses of projection methods are numerous in the literature. See for example [7,11–16]. On the other hand, approximate projection methods have received much less attention [9,17], and these analyses did not consider the effect of boundary conditions or the form of the pressure update used in PM II. As we show below, when an approximate projection is used in combination with the second-order pressure update, the stability of the numerical scheme is not robust.

In order to explore the stability of the various discretizations, we analyze a one-dimensional model problem: the time-dependent, one-dimensional Stokes equations with homogeneous boundary conditions on the velocity and no external body forces. Although this problem seems trivial, it is not. It describes how particular perturbations to the horizontal velocity and the pressure, those that vary only in the x-direction, propagate in time when a projection method or approximate projection method is used to solve the Stokes equations. The discrete time system for projection methods I and II is

$$\frac{u^* - u^n}{\Delta t} = -\partial_x p^{n-1/2} + \frac{v}{2}\partial_{xx}(u^* + u^n), \tag{38}$$

$$\Delta t \partial_{xx}\phi = \partial_x u^*, \tag{39}$$

$$u^{n+1} = u^* - \Delta t \partial_x \phi, \tag{40}$$

$$p^{n+1/2} = p^{n-1/2} + \phi - \chi\frac{v}{2}\partial_x u^*, \tag{41}$$

where χ is 0 for PM I or 1 for PM II. In order for a discretization to be stable, the solution to this system must decay to zero for any initial data.

Projection method III is treated separately because the pressure from one time step does not affect the velocity or the pressure at the next time step. This simplifies the stability analysis considerably, because we only need to study how errors propagate in the velocity. The discrete time system for PM III is

$$\frac{u^* - u^n}{\Delta t} = \frac{v}{2}\partial_{xx}(u^* + u^n), \tag{42}$$

$$\Delta t \partial_{xx}\phi = \partial_x u^*, \tag{43}$$

$$u^{n+1} = u^* - \Delta t \partial_x \phi, \tag{44}$$

$$p^{n+1/2} = \phi - \frac{v}{2}\partial_x u^*. \tag{45}$$

4.1. Stability on MAC grid

We first discretize space using the MAC discretization, which means that the velocities are stored at cell edges and the pressure is stored at the cell centers. Divide the domain into N cells of equal size, with the width of each cell being $h = 1/N$. There are N unknowns for the pressure (one for each cell) and $N - 1$ unknowns for the velocity ($N + 1$ cell edges with two of the values given). The discrete Laplacian applied to the velocity (and intermediate velocity) is defined in the usual manner

$$(L_{\mathrm{d}}u)_{j+1/2} = \frac{u_{j-1/2} - 2u_{j+1/2} + u_{j+3/2}}{h^2}, \tag{46}$$

where the subscript d signifies that the velocity satisfies homogeneous Dirichlet boundary conditions, so that at the first and last interior cell edges ($j = 1$ and $j = N - 1$), the boundary conditions $u_{1/2} = u_{N+1/2} = 0$ are used to evaluate the operator for both the velocity and the intermediate velocity. The discrete divergence operator is defined as

$$(Du)_j = \frac{u_{j+1/2} - u_{j-1/2}}{h}. \tag{47}$$

As with the Laplacian defined above, homogeneous Dirichlet boundary conditions are used to evaluate the divergence at points adjacent to the boundary.

The gradient operator is defined by

$$(G\phi)_{j+1/2} = \frac{\phi_{j+1} - \phi_j}{h}. \tag{48}$$

Because the gradient maps cell centers to interior cell edges, no boundary conditions are necessary for evaluating the gradient. The gradient and divergence operators are related by

$$G = -D^{\mathrm{T}}, \tag{49}$$

which can be used to show that the discrete projection is orthogonal. The discrete version of Eq. (39) arises from the discrete Hodge decomposition

$$u^* = u^{n+1} + \Delta t G\phi; \quad Du^{n+1} = 0. \tag{50}$$

Applying the discrete divergence operator introduces the discrete Laplacian

$$L_{\mathrm{n}} = DG, \tag{51}$$

where the subscript n signifies that this Laplacian takes into account the homogeneous Neumann boundary conditions used in the projection. With these discrete operators, (38)–(41) for PM I and PM II are discretized to

$$\frac{u^* - u^n}{\Delta t} = -Gp^{n-1/2} + \frac{v}{2} L_{\mathrm{d}}(u^* + u^n), \tag{52}$$

$$\Delta t L_{\mathrm{n}} \phi = Du^*, \tag{53}$$

$$u^{n+1} = u^* - \Delta t G\phi, \tag{54}$$

$$p^{n+1/2} = p^{n-1/2} + \phi - \chi \frac{v}{2} Du^*. \tag{55}$$

In order to analyze the stability of this system, we diagonalize Eqs. (52) and (53), eliminate the intermediate variables to find the linear operator that advances velocity and pressure in time, and then find the eigenvalues of this operator. For $j = 1, \ldots, (N - 1)$ and $m = 1, \ldots, (N - 1)$, the jth component of the mth eigenvector of L_{d} is

$$s_j^{(m)} = \sin(m\pi hj), \tag{56}$$

with corresponding eigenvalue

$$\lambda^{(m)} = -\frac{4}{h^2} \sin^2\left(\frac{m\pi h}{2}\right). \tag{57}$$

The form of the eigenvalues of L_{n} is the same as for L_{d}, but the eigenvectors are

$$c_j^{(m)} = \cos(m\pi h(j - 1/2)), \tag{58}$$

for $m = 0, \ldots, (N-1)$ and $j = 1, \ldots, N$. The discrete divergence and gradient map these sets of eigenvectors to each other in the following manner:

$$D\boldsymbol{s}^{(m)} = \frac{2}{h}\sin\left(\frac{m\pi h}{2}\right)\boldsymbol{c}^{(m)} \tag{59}$$

and

$$G\boldsymbol{c}^{(m)} = -\frac{2}{h}\sin\left(\frac{m\pi h}{2}\right)\boldsymbol{s}^{(m)}. \tag{60}$$

Let S be the orthonormal matrix whose mth column is given by the normalized $\boldsymbol{s}^{(m)}$, and let C be the orthonormal matrix with column $m = 1, \ldots, (N-1)$ given by the normalized $\boldsymbol{c}^{(m)}$, and column N given by the normalized $\boldsymbol{c}^{(0)}$. Make the changes of coordinates

$$u = S\widehat{u}, \quad u^* = S\widehat{u}^*, \quad p = C\widehat{p}, \quad \phi = C\widehat{\phi} \tag{61}$$

in Eqs. (52) and (53). Then multiplying (53) and (55) on the left by C^{T} and (52) and (54) by S^{T}, we diagonalize the equations.

The mth set of equations is

$$\frac{\widehat{u}_m^* - \widehat{u}_m^n}{\Delta t} = \frac{2}{h}\sin\left(\frac{m\pi h}{2}\right)\widehat{p}_m^{n-1/2} - \frac{2v}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)(\widehat{u}_m^* + \widehat{u}_m^n), \tag{62}$$

$$-\frac{4\Delta t}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)\widehat{\phi}_m = \frac{2}{h}\sin\left(\frac{m\pi h}{2}\right)\widehat{u}_m^*, \tag{63}$$

$$\widehat{u}_m^{n+1} = \widehat{u}_m^* + \frac{2\Delta t}{h}\sin\left(\frac{m\pi h}{2}\right)\widehat{\phi}_m, \tag{64}$$

$$\widehat{p}_m^{n+1/2} = \widehat{p}_m^{n-1/2} + \widehat{\phi}_m - \chi\frac{v}{h}\sin\left(\frac{m\pi h}{2}\right)\widehat{u}_m^*. \tag{65}$$

Eq. (62) can be solved for $\widehat{u}^*$, and (63) can be solved for $\widehat{\phi}_m$. After eliminating $\widehat{u}^*$ and $\widehat{\phi}_m$, for each m we obtain a 2×2 matrix X_m which maps $(\widehat{u}_m^n, \widehat{p}_m^{n-1/2})^{\mathrm{T}}$ to $(\widehat{u}_m^{n+1}, \widehat{p}_m^{n+1/2})^{\mathrm{T}}$. The eigenvalues of the matrices X_m determine the stability of the discretization. Solving (63) for $\widehat{\phi}_m$ and substituting this into (64), we see that

$$\widehat{u}_m^{n+1} = 0. \tag{66}$$

This means that errors introduced at some time step in the velocity or pressure, do not propagate to the velocity at the next time step. The first row of X_m is entirely zero for all m, which means that X_m is lower triangular, and so to determine the stability, we need to examine only x_{22}.

After some algebraic manipulations using Eqs. (62)–(65), we find

$$x_{22} = \frac{(1-\chi)\frac{2v\Delta t}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)}{1 + \frac{2v\Delta t}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)}. \tag{67}$$

If PM II is used ($\chi = 1$), then all the eigenvalues of X_m are zero, and $X_m^2 = 0$ for any m. Because $x_{21} \neq 0$, errors introduced in the velocity at a time step appear in the pressure of the following time step, but do not appear in either the velocity or pressure in the time step after that. For PM I ($\chi = 0$), the method is still stable, because all the eigenvalues are less than one in magnitude.

Eqs. (63) and (64) are the same for projection method III. Again $\widehat{u}_m^{n+1} = 0$, and because the pressure does not propagate, it need not be considered for stability. Therefore all three projection methods are stable on the MAC grid.

4.2. Stability on a cell-centered grid – Approximate projection

We now consider the case of an approximate projection on a cell-centered grid. Eqs. (38)–(41) for PM I and PM II are discretized to

$$\frac{u^* - u^n}{\Delta t} = -G_p p^{n-1/2} + \frac{\nu}{2} L_{\mathrm{d}}(u^* + u^n), \tag{68}$$

$$\Delta t L_{\mathrm{n}} \phi = D u^*, \tag{69}$$

$$u^{n+1} = u^* - \Delta t G \phi, \tag{70}$$

$$p^{n+1/2} = p^{n-1/2} + \phi - \chi \frac{\nu}{2} D u^*. \tag{71}$$

This system is similar to (52)–(55), but the definitions of the operators must be modified, because all variables are stored at the cell centers. The definitions of the operators are given below.

The form of the Laplacian is

$$(Lw)_j = \frac{w_{j-1} - 2w_j + w_{j+1}}{h^2}, \tag{72}$$

where the different boundary conditions are accounted for by setting values of ghost cells outside the domain. For example, the ghost cell value for the velocity (and intermediate velocity) at the left boundary is set to be $u_{-1} = -u_1$, which corresponds to linearly extrapolating the velocity from the physical boundary condition and the first interior point. Note that the eigenfunctions of the Laplacian with homogeneous Dirichlet boundary conditions are sine functions, and this method of setting the ghost cells for the velocity reflects this structure. The ghost cell value for ϕ at the left boundary is set by $\phi_{-1} = \phi_1$, which is a discretization of the homogeneous Neumann boundary condition that ϕ satisfies. To avoid confusion, we add the the subscripts d and n to the Laplacian to signify which boundary conditions are used for evaluation.

The divergence and gradient are computed by centered differencing, and as with the Laplacian, the boundary conditions are handled using ghost cells. Note that on the cell-centered grid, boundary conditions are required to evaluate the gradient. This was not the case with the MAC grid. The boundary conditions for ϕ are used to set the its ghost cells as for the Laplacian, but there are no physical boundary conditions for the pressure. In Eq. (68), we have denoted the pressure gradient with the subscript p to indicate that this operator may not be the same gradient operator that is applied to ϕ.

For projection method I, the pressure gradient does not need to be computed from the pressure; it can simply be updated. In this case, the computed pressure is forced to have constant normal derivative in time near the boundary, as a consequence of the boundary conditions used in the projection. For projection method II, there is ambiguity near the boundary whether one chooses to update the pressure gradient or the pressure. Updating the pressure gradient requires computing the Laplacian of the gradient of ϕ. The normal component of the gradient of ϕ is zero on the boundary, and this can be used in computing the Laplacian of the normal component of the gradient of ϕ near the boundary. However, there are no boundary conditions for the tangential component of the gradient of ϕ. On the other hand, if the pressure is to be updated, it must then be differenced, but there are no boundary conditions for the pressure for use in a difference formula. It seems that either a one sided difference must be used near the boundary, or a lower order error must be introduced.

Table 1
Possible ghost cell setting methods and the resulting pressure gradients used at the left boundary

Type	Ghost cell formula	Pressure gradient
Constant	$p_0 = p_1$	$(p_2 - p_1)/2h$
Linear	$p_0 = 2p_1 - p_2$	$(p_2 - p_1)/h$
Quadratic	$p_0 = 3p_1 - 3p_2 + p_3$	$(-p_3 + 4p_2 - 3p_1)/2h$

We analyze the case in which the pressure is updated and then differenced. Because there are no physical boundary conditions for the pressure, the ghost cell values must be extrapolated using data only from the interior cells. Then the standard centered difference operator can be applied at the first interior cell to compute the gradient, but the resulting gradient is effectively a one sided derivative whose form is determined by the manner in which the ghost cell value has been set. Some possible polynomial extrapolations and the resulting pressure gradients are given in Table 1. We refer to these gradients as gradient 0, gradient 1, and gradient 2, to denote their accuracy (or order of polynomial extrapolation used for the ghost cells).

4.2.1. Constant extrapolation

We repeat the procedure used to analyze the stability on the MAC grid, but only for the case when the ghost cell value for the pressure is set by constant extrapolation. The resulting pressure gradient is used for analysis because the system can be diagonalized, while this is not true for the other pressure gradients. Also note that with this choice of extrapolation, the gradient applied to the pressure is the same as the gradient applied to ϕ because the constant extrapolation formula is identical to the discretization of the homogeneous Neumann boundary condition. Even though our analysis is restricted to just one of the many possible gradients, it is still very informative because it gives information about how the stability is affected by using an approximate projection and the high-order pressure correction. All of the gradients are identical at the interior points, and so because they differ only at points near the boundary, they can be viewed as perturbations of one another.

The operator L_n is the same operator used on the MAC grid. The eigenvalues of L_d are given by (57), but the eigenvectors on the cell-centered grid are

$$s_j^{(m)} = \sin(m\pi h(j - 1/2)), \tag{73}$$

for $m = 1, \ldots, N$ and $j = 1, \ldots, N$. The discrete divergence is computed by centered differencing everywhere, with the ghost cell values outside the domain set by $u_0 = -u_1$ to correspond to the homogeneous Dirichlet boundary conditions. This divergence operator applied to the eigenvectors of L_d gives

$$D\boldsymbol{s}^{(m)} = \frac{1}{h}\sin(m\pi h)\boldsymbol{c}^{(m)}. \tag{74}$$

The discrete gradient of ϕ is also computed by centered differencing everywhere, with the ghost cells set by $\phi_0 = \phi_1$, which can be interpreted either as discretizing the homogeneous Neumann boundary condition or as constant extrapolation. This discrete gradient acting on the eigenvectors of L_n gives

$$G\boldsymbol{c}^{(m)} = -\frac{1}{h}\sin(m\pi h)\boldsymbol{s}^{(m)}. \tag{75}$$

Again define the matrices S and C as for the MAC grid, and make the transformations

$$u = S\widehat{u}, \quad u^* = S\widehat{u}^*, \quad p = C\widehat{p}, \quad \phi = C\widehat{\phi}. \tag{76}$$

The *m*th set of equations is

$$\frac{\widehat{u}_m^* - \widehat{u}_m^n}{\Delta t} = \frac{1}{h}\sin(m\pi h)\widehat{p}_m^{n-1/2} - \frac{2v}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)(\widehat{u}_m^* + \widehat{u}_m^n), \tag{77}$$

$$-\frac{4\Delta t}{h^2}\sin^2\left(\frac{m\pi h}{2}\right)\widehat{\phi}_m = \frac{1}{h}\sin(m\pi h)\widehat{u}_m^*, \tag{78}$$

$$\widehat{u}_m^{n+1} = \widehat{u}_m^* + \frac{\Delta t}{h}\sin(m\pi h)\widehat{\phi}_m, \tag{79}$$

$$\widehat{p}_m^{n+1/2} = \widehat{p}_m^{n-1/2} + \widehat{\phi}_m - \chi\frac{v}{2h}\sin(m\pi h)\widehat{u}_m^*. \tag{80}$$

From this system, we eliminate the intermediate variables $\widehat{u}_m^*$ and $\widehat{\phi}_m$, and obtain, after algebraic manipulations,

$$\begin{bmatrix} \widehat{u}_m^{n+1} \\ \widehat{p}_m^{n+1/2} \end{bmatrix} = \begin{bmatrix} \frac{(1-a)}{(1+a)}s^2 & \frac{scba}{(1+a)} \\ -\frac{sc(1-a)(1+\chi a)}{ba(1+a)} & \frac{s^2+a(1-\chi c^2)}{(1+a)} \end{bmatrix} \begin{bmatrix} \widehat{u}_m^{n} \\ \widehat{p}_m^{n-1/2} \end{bmatrix}, \tag{81}$$

where

$$a = \frac{2v\Delta t}{h^2}\sin^2\left(\frac{m\pi h}{2}\right), \quad b = \frac{h}{v}, \quad s = \sin\left(\frac{m\pi h}{2}\right), \quad c = \cos\left(\frac{m\pi h}{2}\right).$$

Let the matrix that maps $(\widehat{u}_m^n, \widehat{p}_m^{n-1/2})^{\mathrm{T}}$ to $(\widehat{u}_m^{n+1}, \widehat{p}_m^{n+1/2})^{\mathrm{T}}$ be denoted by X_m.

We now compute the eigenvalues of each X_m to show that they are less than one in magnitude, making the scheme stable for both the standard pressure update ($\chi = 0$) and the higher order correction ($\chi = 1$). The trace and determinant of X_m are

$$\mathrm{Tr}(X_m) = \frac{2s^2 + a(1-s^2)(1-\chi)}{(1+a)}, \quad \det(X_m) = \frac{(1-a)s^2}{(1+a)}, \tag{82}$$

and the characteristic polynomial is

$$\lambda^2 - \mathrm{Tr}\,\lambda + \det = 0. \tag{83}$$

The roots of this equation are either real, or a complex conjugate pair. If the roots are complex, then their squared magnitude is equal to the determinant, which is less than one. Therefore if the roots are complex, they must have magnitude less than one. Now suppose that the roots are real. The graph of the characteristic polynomial is just a parabola with its vertex at Tr/2, which is positive. This implies that there must be at least one positive root, and if it is less than one, then the smaller root must be greater than negative one. It is sufficient to examine only the larger root. Note that it is also sufficient to consider the case of $\chi = 0$ (PM I), because the trace is larger in that case than for $\chi = 1$ (PM II).

For the case $\chi = 0$, the larger eigenvalue is given by

$$\lambda = \frac{1}{2}\left(\frac{(2-a)s^2 + a}{(1+a)} + \sqrt{\frac{((2-a)s^2+a)^2}{(1+a)^2} - \frac{4(1-a)s^2}{(1+a)}}\right). \tag{84}$$

Simplifying under the square root gives

$$\lambda = \frac{1}{2(1+a)}(p_1(s^2) + \sqrt{p_2(s^2)}), \tag{85}$$

where

$$p_1(r) = (2 - a)r + a$$

and

$$p_2(r) = (2 - a)^2 r^2 + 2(a^2 + 2a - 2)r + a^2.$$

Note that s is in the range [0,1]. If $a \leqslant 2$, then both p_1 and p_2 attain their maximum on [0,1] at 1. And so,

$$\lambda \leqslant \frac{1}{2(1+a)}(p_1(1) + \sqrt{p_2(1)}) = 1 \quad \text{for } a \leqslant 2. \tag{86}$$

Now consider the case of $a > 2$. It turns out that the maximum value of λ is again obtained at $s^2 = 1$, but it is a little more difficult to show this. Denote s^2 by r. A straightforward computation shows that

$$\frac{\partial^2 \lambda}{\partial r^2} = \frac{-2(2a-1)(a-1)}{(p_2(r))^{3/2}} < 0, \quad \text{since } a > 2, \tag{87}$$

which means that the derivative of λ with respect to r is a decreasing function. The minimum value of the derivative of λ is obtained at $r = 1$, which is

$$\left.\frac{\partial \lambda}{\partial r}\right|_{r=1} = \frac{1}{2a} > 0. \tag{88}$$

Together (87) and (88) imply that λ is an increasing function of r on [0,1], and so we have shown that

$$\lambda \leqslant \frac{1}{2(1+a)}(p_1(1) + \sqrt{p_2(1)}) = 1 \quad \text{for } a > 2. \tag{89}$$

Therefore, all the eigenvalues are less than or equal to one in magnitude, and both PM I and PM II are stable.

Showing projection method III is stable is very easy. We can use the results from above by assuming that $\widehat{p}^{n-1/2}$ is zero. Therefore, the element of X_m in the first row and column is all that is needed to determine the stability. This element is always less than 1 in magnitude, and therefore PM III is also stable.

4.2.2. Higher order gradients

As we proved in the previous section, setting the ghost cell values for the pressure using constant extrapolation produced a stable scheme for all projection methods. This analysis did not consider the accuracy of such an approach. The pressure gradient that results is zeroth-order accurate near the boundary, and as we shall see later in a numerical test, the pressure is not second-order accurate for PM II. A more accurate extrapolation, and hence pressure gradient, must be used near the boundary to produce a second-order scheme. For PM III, the pressure gradient never needs to be computed, and so the discussion that follows only applies to PM I and PM II.

When using a different gradient for the pressure, we were unable to diagonalize the resulting system as was done when constant extrapolation was used. All of the gradients are identical to the one analyzed away from the boundary, but differ at grid points adjacent to the boundary. We consider the other pressure gradients as localized perturbations to the one that was analyzed in the previous section. We can gain insight into the stability of these schemes by examining how sensitive the previous scheme is to perturbations.

In the previous section, we showed that the eigenvalues are an increasing function of the parameter $s^2 \in [0,1]$. Recall that

$$s = \sin\left(\frac{m\pi h}{2}\right), \tag{90}$$

and so s near 1 corresponds to m near $1/h = N$. The eigenvalues are thus an increasing function of the frequency, m, of the eigenfunctions. We examine the structure of the matrices X_m for large m. In particular, we examine X_{N-1}, so that $mh = 1 - h$, and we expand X_{N-1} in a Taylor series for small h to obtain

$$X_{N-1} = \begin{bmatrix} -1 + \left(\frac{v\Delta t\pi^2+4}{4v\Delta t}\right)h^2 & \frac{\pi}{2v}h^2 \\ \frac{v\pi}{2}\chi + \left(\frac{3\pi-\chi(v\Delta t\pi^2+6\pi)}{12\Delta t}\right)h^2 & 1 - \frac{\chi\pi^2}{4}h^2 \end{bmatrix} + \mathcal{O}(h^4). \tag{91}$$

This expansion reveals a significant difference between the cases $\chi = 0$ and $\chi = 1$. Suppose first that $\Delta t = \mathcal{O}(h)$ and v is $\mathcal{O}(1)$. With these assumptions, if $\chi = 0$, the matrix is almost diagonal, but if $\chi = 1$ there is an $\mathcal{O}(1)$ off diagonal element. This large off diagonal element makes the matrix significantly more sensitive to perturbations.

As an example of how the off-diagonal element affects stability, consider the matrix

$$A = \begin{bmatrix} -1 + \delta_1 & 0 \\ d & 1 - \delta_2 \end{bmatrix}. \tag{92}$$

Now suppose that this matrix is perturbed in the upper right element, so that

$$\tilde{A} = \begin{bmatrix} -1 + \delta_1 & \epsilon \\ d & 1 - \delta_2 \end{bmatrix}. \tag{93}$$

If

$$\epsilon > \frac{\delta_2(1 - \delta_1)}{d}, \tag{94}$$

then $\tilde{A}$ has an eigenvalue greater than one. If d is small, then ϵ would have to be large to produce an eigenvalue greater than one. As d gets larger, the perturbations that force an instability become smaller. If d is $\mathcal{O}(1)$, then perturbations of size δ_2 can produce an instability. Applying this to the matrix X_{N-1} with $\chi = 1$, shows that perturbations of order h^2 could produce an instability.

Projection method II with constant extrapolation for the pressure ghost cells produces a stable scheme, but is significantly more sensitive to perturbations than projection method I. This analysis suggests that PM II will become unstable when higher order extrapolations are used to set the ghost cell values of the pressure, unless we choose those extrapolating formulas carefully. This is shown to be the case in the following section.

5. Numerical tests

We perform two numerical tests. First, we test the stability of the model problem for different choices of pressure gradients for both projection methods I and II. We test all of the gradients (gradient 0, gradient 1, and gradient 2) given in Table 1, and one more second-order gradient, which we will refer to as gradient 2a. We derive the formula for gradient 2a by linearly extrapolating the gradient itself near the boundary, rather than extrapolating the pressure. The resulting gradient is second-order accurate at every point, and has the form

$$(Gp)_1 = \frac{-2p_1 + p_2 + 2p_3 - p_4}{2h}. \tag{95}$$

If the ghost cell values are set using the formula

$$p_0 = 2p_1 - 2p_3 + p_4, \tag{96}$$

this gradient can then be computed by using the standard centered difference near the boundary. This method of setting the ghost cells extrapolates the pressure to third order, but it uses a different set of grid points than the standard quadratic polynomial extrapolation. After exploring the stability using the model problem, we then explore both the stability and accuracy when these gradients are used in the full Navier–Stokes equations. Note that the discrete gradient operator applied to ϕ is the same for all tests.

5.1. Model problem

We proved that the model problem is stable if gradient 0 is used to compute the pressure gradient. We tested the stability of the model problem using the pressure gradients 1, 2, and 2a for a large range of time steps and viscosities. For different values of the cell width h, we varied the time step and viscosity between h^3 and h^{-3}. We began the computation with zero pressure, but with a random velocity, normalized to have 2-norm equal to 1. We used a random velocity because it is likely to contain all frequencies. To determine the stability, we ran the computation until it was clear that the perturbation was growing or decaying.

For projection method I, all pressure gradients produced stable schemes. For projection method II, only gradient 2 (quadratic polynomial extrapolation for the ghost cells) became unstable, while all the other gradients were stable for all time steps and viscosities tested. Gradient 2 was stable for very small viscosities and time steps, but was very often unstable. Our analysis predicted that the higher order pressure correction is sensitive to perturbations at high frequencies, and apparently this sensitivity can produce instability depending on the gradient. However, it was not clear why gradient 2 produced an unstable scheme, while the other gradients were stable.

In all of the schemes, high frequency oscillations dominated in the pressure and the velocity after a few time steps. This is not surprising, because we showed that the eigenvalues were an increasing function of the frequency of the eigenvectors. For all the gradients except gradient 2, these oscillations decayed in time. When gradient 2 became unstable, high frequency oscillations in both the velocity and the pressure grew near the boundary. An example is displayed in Fig. 3. This particular computation was run with $N = 32$ cells, time step $h/2$, viscosity 1, and using a parabola as the initial velocity. The velocity and pressure are plotted after 200 time steps.

To see what is different about gradient 2, we examine how each of the gradients act on a particular high frequency mode. Consider the grid function $\psi_j = (-1)^j$, which corresponds to the highest frequency mode that can be represented on the grid. Note that the L_1 norm of this mode is one. For points not adjacent to the boundary, all of the gradients applied to this function evaluate to zero. The values of the discrete pressure gradients at the first grid point are shown in Table 2. The values increase from pressure gradient 0 to 2,

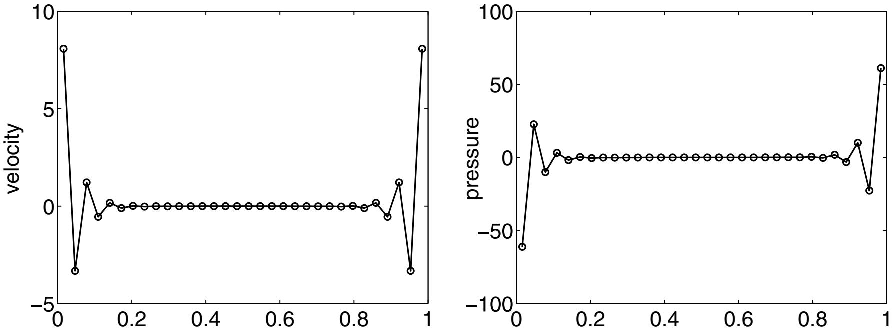

Fig. 3. The velocity and pressure are plotted after 200 time steps for the model problem using pressure gradient 2. The computation was run with $N = 32$ cells, time step $h/2$, viscosity 1, and a parabolic profile as the initial velocity.

Table 2
Value of the discrete pressure gradient applied to $\psi_j = (-1)^j$ at grid point $j = 1$

Name	Pressure gradient	$(G\psi)_1$
0	$(p_2 - p_1)/2h$	$1/h$
1	$(p_2 - p_1)/h$	$2/h$
2	$(-p_3 + 4p_2 - 3p_1)/2h$	$8/h$
2a	$(-2p_1 + p_2 + 2p_3 - p_4)/2h$	0

while pressure gradient 2a gives zero, damping this mode completely in the pressure. In fact, it can be shown that if standard nth-order polynomial extrapolation, that is using the n closest interior grid points, is used to set the ghost cell (as it is in gradients 0–2), the value of the gradient applied to ψ near the boundary is $2^n/h$. Considering the grid point adjacent to the other boundary, the L_1 norm of these pressure gradients of ψ is 2^{n+1}. As the analysis using pressure gradient 0 showed, PM II is sensitive to high frequency modes, and standard polynomial extrapolation of the pressure produced gradients that increase the L_1 norm of the highest frequency mode. This argument provides some insight into why pressure gradient 2 produced an unstable scheme.

5.2. Navier–Stokes test

We now test the significance of the results of the model problem by solving the two-dimensional Navier–Stokes equations. We use a test problem from Brown et al. [6] for forced flow. The solution is

$$u = \cos(2\pi(x-\omega))(3y^2 - 2y), \tag{97}$$

$$v = 2\pi\sin(2\pi(x-\omega))(y^3 - y^2), \tag{98}$$

$$p = -\frac{\omega'}{2\pi}\sin(2\pi(x-\omega))(\sin(2\pi y) - 2\pi y + \pi) = -v\cos(2\pi(x-\omega))(-2\sin(2\pi y) + 2\pi y - \pi), \tag{99}$$

with $\omega = 1 + \sin(2\pi t^2)$. From this solution, the force which drives the flow is calculated. The viscosity is set to one. Periodic boundary conditions are used in the x-direction and Dirichlet boundary conditions are used in the y-direction. The numerical solution is compared with the actual solution at time $t = 0.5$. The time step used for each computation is $\Delta t = h/2$, where h is the grid spacing. The nonlinear terms are handled explicitly using the method of Kim and Moin [20], given in (36).

Table 3
Max norm of error in horizontal velocity, u

Method	Pressure gradient	128 × 128	256 × 256	512 × 512	Rate
PM I	0	9.86E − 04	2.40E − 04	5.93E − 05	2.02
	1	9.86E − 04	2.40E − 04	5.93E − 05	2.02
	2	9.86E − 04	2.40E − 04	5.93E − 05	2.02
	2a	9.86E − 04	2.40E − 04	5.93E − 05	2.02
PM II	0	9.30E − 04	2.34E − 04	5.87E − 05	2.00
	1	9.30E − 04	2.33E − 04	5.86E − 05	2.00
	2	2.55E − 03	3.69E − 01	Nan	–
	2a	9.30E − 04	2.33E − 04	5.86E − 05	2.00

Table 4
Max norm of error in pressure

Method	Pressure gradient	128 × 128	256 × 256	512 × 512	Rate
PM I	0	2.33E − 02	7.78E − 03	4.03E − 03	1.55
	1	2.27E − 02	8.61E − 03	4.61E − 03	1.49
	2	2.31E − 02	7.86E − 03	4.37E − 03	1.52
	2a	2.29E − 02	8.16E − 03	4.42E − 03	1.51
PM II	0	2.51E − 02	1.20E − 02	5.91E − 03	1.44
	1	9.90E − 03	2.38E − 03	5.86E − 04	2.03
	2	1.02E − 02	3.22E+00	Nan	–
	2a	9.94E − 03	2.38E − 03	5.88E − 04	2.03

Table 5
L_1 norm of the error in the horizontal velocity, u

Method	Pressure gradient	128 × 128	256 × 256	512 × 512	Rate
PM I	0	1.78E − 04	4.43E − 05	1.10E − 05	2.01
	1	1.77E − 04	4.39E − 05	1.09E − 05	2.01
	2	1.78E − 04	4.40E − 05	1.10E − 05	2.01
	2a	1.77E − 04	4.39E − 05	1.09E − 05	2.01
PM II	0	1.73E − 04	4.35E − 05	1.09E − 05	2.00
	1	1.73E − 04	4.35E − 05	1.09E − 05	2.00
	2	2.00E − 04	4.17E − 03	Nan	–
	2a	1.73E − 04	4.35E − 05	1.09E − 05	2.00

Table 6
L_1 norm of the error in the pressure

Method	Pressure gradient	128 × 128	256 × 256	512 × 512	Rate
PM I	0	5.56E − 03	1.30E − 03	3.19E − 04	2.04
	1	5.58E − 03	1.30E − 03	3.18E − 04	2.05
	2	5.59E − 03	1.31E − 03	3.19E − 04	2.05
	2a	5.59E − 03	1.30E − 03	3.18E − 04	2.05
PM II	0	1.61E − 03	4.44E − 04	1.30E − 04	1.88
	1	1.32E − 03	3.16E − 04	7.77E − 05	2.03
	2	1.18E − 03	2.06E − 01	Nan	–
	2a	1.32E − 03	3.16E − 04	7.77E − 05	2.03

The max norm of the errors are displayed in Table 3 for the horizontal velocity, u, and in Table 4 for the pressure. Tables 5 and 6 give the L_1 norm of the errors for the velocity and pressure respectively. The errors in the vertical velocity, v, are similar to those of u and are not displayed. For projection method I, all the pressure gradients give essentially equivalent results. The velocity is second-order accurate in both the L_1 and maximum norm, and the pressure is second-order accurate in the L_1 norm, but appears to be order 3/2 accurate in the maximum norm.

For projection method II, pressure gradient 2 was sometimes unstable, and all the other gradients were stable. This agrees with the numerical results for the model problem. Gradients 1 and 2a showed full second-order accuracy in the pressure, but pressure gradient 0 showed order 3/2 accuracy. The method using gradient 2 became unstable as time and space were refined. The instability showed up in these refined cases, not because of the smaller time step or space step, but because more time steps were taken. To test this we reran the computation with $h = 1/128$ for more time steps, and the instability became apparent by time step 418 ($t \approx 1.63$).

6. Discussion

We have discussed the ideas behind projection methods for solving the incompressible Navier–Stokes equations. A recent paper by Brown et al. [6] clarified issues surrounding the boundary conditions, the pressure, and the accuracy of projection methods. They may be the first to show full second-order accuracy in the pressure. However, their analysis did not consider the stability of these schemes. We have extended their results by analyzing stability. We showed that on the MAC grid, all schemes are stable. However, when an approximate projection is used, PM II is susceptible to instabilities, depending on how the gradient is computed near the boundary.

This work was inspired by our observation of instabilities while experimenting with an approximate projection method. In implementing PM II by computing the pressure, it is required to choose a difference formula for the pressure at points adjacent to the boundary. For a scheme that is supposed to be second-order accurate point-wise, a natural choice for differencing the pressure is the second-order, one-sided difference using the three points closest to the boundary, which in this paper we refer to as gradient 2. As we demonstrated in this paper, this gradient produced an unstable scheme. It appears that using a first-order pressure gradient (gradient 1) near the boundary is sufficient to produce a stable, second-order accurate scheme. However, the only gradient for which we proved the scheme is stable is gradient 0, which did not show second-order accuracy. We prefer gradient 2a, because, unlike gradient 1, it damps high-frequency errors of the form that appeared when the scheme was unstable, as displayed in Fig. 3. We did not prove that this scheme is stable, but we have been using a code with gradient 2a for some time, and have not noted any instabilities related to the projection method.

We have focused on approximate projection methods on cell-centered grids. This is because cell-centered grids are particularly useful if high-resolution upwinding is used for the nonlinear terms, and are more commonly used than vertex-centered grids. An approximate projection on a vertex-centered grid was used for the numerical tests in [6], and no instabilities arose. As we showed, the instabilities arose from the manner in which the pressure was differenced near the boundary. On a node-centered grid, the pressure is computed at the boundary, and so no extrapolation is required to difference the pressure.

Acknowledgments

The authors hank Elijah Newren and Jingyi Zhu for helpful discussions while writing this paper. R.D.G. was supported in part by a VIGRE Grant and Grant DMS-0139926 to A.L.F., and A.L.F. was supported in part by Grants DMS-9805518 and DMS-0139926 from the National Science Foundation.

References

[1] A.J. Chorin, Numerical solutions of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[2] A.J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, Math. Comput. 23 (1969) 341–353.

[3] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comp. Phys. 85 (1989) 257–283.

[4] L.H. Howell, J.B. Bell, An adaptive mesh projection method for viscous incompressible flow, SIAM J. Sci. Comput. 18 (1997) 996–1013.

[5] A.S. Almgren, J.B. Bell, W.G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (1996) 358–369.

[6] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comp. Phys. 168 (2001) 464–499.

[7] J.C. Strikwerda, Y.S. Lee, The accuracy of the fractional step method, SIAM J. Numer. Anal. 37 (1999) 37–47.

[8] J.B. Perot, An analysis of the fractional step method, J. Comp. Phys. 108 (1993) 51–58.

[9] A.S. Almgren, J.B. Bell, W.Y. Crutchfield, Approximate projection methods: Part I. Invicid analysis, SIAM J. Sci. Comput. 22 (2000) 1139–1159.

[10] S.A. Orszag, M. Israeli, M.O. Deville, Boundary conditions for incompressible flows, J. Sci. Comput. 1 (1986) 75–111.

[11] J. Shen, On error estimates of projection methods for Navier–Stokes equations: First-order schemes, SIAM J. Numer. Anal. 29 (1992) 57–77.

[12] J. Shen, On error estimates of the projection methods for the Navier–Stokes equations: Second-order schemes, Math. Comput. 65 (1996) 1039–1065.

[13] Wienan E., J.-G. Liu, Projection method I: Convergence and numerical boundary layers, SIAM J. Numer. Anal. 32 (1995) 1017–1057.

[14] Wienan E., J.-G. Liu, Projection method II: Godunov–Ryabenki analysis, SIAM J. Numer. Anal. 33 (1996) 1597–1621.

[15] B.R. Wetton, Analysis of the spatial error for a class of finite difference methods for viscous incompressible flow, SIAM J. Numer. Anal. 34 (1997) 723–755.

[16] B.R. Wetton, Error analysis for Chorin's original fully discrete projection method and regularizations in space and time, SIAM J. Numer. Anal. 34 (1997) 1683–1697.

[17] W.J. Rider, The robust formulation of approximate projection methods for incompressible flows, Techical Report. LA-UR-94-2000, LANL, 1994.

[18] A.J. Chorin, J.E. Marsden, A Mathematical Introduction to Fluid Mechanics, third ed., Springer, New York, 1998.

[19] J. Van Kan, A second order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.

[20] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comp. Phys. 59 (1985) 308–323.

[21] M.L. Minion, A projection method for locally refined grids, J. Comp. Phys. 127 (1996) 158–178.

[22] F.H. Harlow, J.E. Welch, Numerical calculation of time dependent viscous incompressible flow of fluids with a free surface, Phys. Fluids 8 (1965) 2182.

[23] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equation with singular forces, J. Comp. Phys. 171 (2001) 822–842.

[24] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comp. Phys. 142 (1998) 1–46.

[25] J. Zhu, The second-order projection method for the backward-facing step flow, J. Comp. Phys. 117 (1995) 318–331.

[26] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comp. Phys. 87 (1990) 171–200.

[27] L. Lee, R.J. LeVeque, An immersed interface method for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.

[28] R.J. LeVeque, Wave propagation algorithms for multidimensional hyperbolic systems, J. Comp. Phys. 131 (1997) 327–353.