Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 2012–2043
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics

Samet Y. Kadioglu [a,1], Rupert Klein [b,2], Michael L. Minion [c,*,3]

[a] *Advanced Nuclear Energy Systems Department, Idaho National Laboratory, P.O. Box 1625, MS 3840, Idaho Falls, ID 83415, United States*
[b] *FB Mathematik and Informatik, Freie Universität Berlin, Konrad Zuse Zentrum Für Informationstechnik, Takustr. 7, 14195 Berlin, Germany*
[c] *Department of Mathematics, Phillips Hall, CB 3250, University of North Carolina, Chapel Hill, NC 27599, United States*

Received 18 March 2007; received in revised form 28 September 2007; accepted 4 October 2007
Available online 18 October 2007

Abstract

A fourth-order numerical method for the zero-Mach-number limit of the equations for compressible flow is presented. The method is formed by discretizing a new auxiliary variable formulation of the conservation equations, which is a variable density analog to the impulse or gauge formulation of the incompressible Euler equations. An auxiliary variable projection method is applied to this formulation, and accuracy is achieved by combining a fourth-order finite-volume spatial discretization with a fourth-order temporal scheme based on spectral deferred corrections. Numerical results are included which demonstrate fourth-order spatial and temporal accuracy for non-trivial flows in simple geometries.

Keywords: Impulse methods; Gauge methods; Auxiliary variable methods; Projection methods; Deferred corrections; Gas dynamics

1. Introduction

The numerical simulation of physical systems involving compressible gas flow is one of the canonical problems in computational science and engineering. When the Mach number M (the ratio of the characteristic velocity to the sound speed) is low, the use of standard explicit numerical methods can become very

* Corresponding author. Tel.: +1 919 962 8475; fax: +1 919 962 9345.
E-mail addresses: samet.kadioglu@inl.gov (S.Y. Kadioglu), rupert.klein@zib.de (R. Klein), minion@amath.unc.edu (M.L. Minion).
URL: http://amath.unc.edu/Minion (M.L. Minion).
[1] Author was supported by the Director, DOE Office of Science, Office of Advanced Scientific Computing Research, Office of Mathematics, Information, and Computational Sciences, Applied Mathematical Sciences Program, under Contract DE-AC03-76SF00098.
[2] Author thanks Deutsche Forschungsgemeinschaft for their support through Grants KL 611/6, KL 611/14.
[3] Author was supported by the Alexander von Humboldt Foundation and the Director, DOE Office of Science, Office of Advanced Scientific Computing Research, Office of Mathematics, Information, and Computational Sciences, Applied Mathematical Sciences Program, under Contract DE-AC03-76SF00098.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.10.008

computationally expensive due to the fact that the time step must scale like M. Numerical strategies have been developed to circumvent this problem [77,8,73,58,55], but an alternative is to actually employ a numerical method which approximates the equations which result from passing to the zero-Mach number limit (hereafter referred to more simply as the *zero-Mach equations*). Physically, this means that fast acoustic waves are removed from the system, so this procedure is appropriate when acoustic waves do not contribute to the relevant dynamics being studied. The numerical implication of solving the zero-Mach equations is that the time step used can be hundreds or even thousands of times larger than if an explicit method for the compressible equations were used.

Examples of systems where the use of the zero-Mach equations has proven successful in numerical simulations include, first of all, those systems for which the use of the incompressible, constant density Euler or Navier–Stokes equations, or the Boussinesq approximation represent an appropriate model. Zero-Mach equations allowing for variable density are of relevance, e.g., for technical combustion systems, [5,6,53,36], astrophysical phenomena such as Type Ia Supernovae, [2,3], or atmospheric flows, [56,43,23,9,35]. In these applications, Mach numbers vary within the range $M \approx 0.01, \ldots, 0.1$. For many atmospheric and astrophysical applications, gravity is dominant with the consequence that the leading order density becomes strictly stratified along the direction of gravity, leading to the class of "anelastic" model equations. Here we focus on the zero-Mach number limit equations obtained when the influence of gravity is moderate. In that case, the leading order density is advected with the fluid.

In the zero-Mach limit, the hyperbolic compressible equations undergo a change of type and, depending on which set of primary variables is used, the energy, pressure, or mass conservation equation turns into a divergence constraint on the velocity [56,43,47,34]. The zero-Mach equations are indeed the correct variable-density analog to the incompressible Euler equations. It is no surprise then, that many existing numerical methods for the zero-Mach equations are extensions of methods designed for the incompressible Euler or Navier–Stokes equations.

Projection methods for incompressible flows, (see e.g. [17,72,33,4,29,11,30]) have been an attractive option in numerical flow simulation because of their efficiency. During one time step for a typical projection method, the most computationally expensive step is usually the solution of a Poisson equation related to the pressure field which guarantees compliance with the velocity divergence constraint. Hence, projection methods achieve a low-computational cost by avoiding iterations coupling the pressure and momentum fields as needed in, e.g., SIMPLE-type pressure correction methods (see e.g., [59,32,8,77]). Extensions of projection methods for variable density or zero-Mach flows have also been developed (see e.g. [7,53,62,54,1,68]).

The original projection method proposed by Chorin [17] is first-order accurate in time, and many second-order extensions have appeared since (some of the first being [33,74,4]). One difficulty in extending projection methods to temporal orders higher than two is that they are typically thought of as fractional step schemes. Specifically, one time step to update the numerical approximation to the velocity $\mathbf{v}$ proceeds by first computing a provisional value $\mathbf{v}^*$ by approximating the momentum equation and then applying a numerical projection to extract the divergence-free part of $\mathbf{v}^*$. In [50,51], an auxiliary variable equation is introduced which provides an explicit, unconstrained evolution equation for $\mathbf{v}^*$. Since $\mathbf{v}^*$ does not obey a divergence constraint, a fractional step approach is not necessary, and a higher-order scheme based on the method of lines can be directly applied to the equation for $\mathbf{v}^*$. Auxiliary variable equations are closely related to impulse or gauge variables (see the review in [66]), and were used in [11] to analyze appropriate boundary conditions for semi-implicit projection methods for the Navier–Stokes equations. Higher-order projection methods based on auxiliary variables utilize an impulse type equation for the auxiliary variables within any given time step but avoid the observed difficulties of methods based solely on impulse variables by reinitializing the auxiliary variables to the velocity (or momentum) after each time step. This distinction is addressed in more detail in Section 3.1.1. Until now, neither auxiliary variable equations nor the related gauge or impulse equations have been extended to variable density flows.

In this paper we introduce a new extension to the auxiliary variable equations in [50,51] appropriate for the zero-Mach, variable density case. The new equations are in conservation form and give an explicit evolution equation for an unconstrained auxiliary variable $\mathbf{m}^*$ which is a momentum variable analog to an impulse variable. In Sections 2.1–2.3, we provide an overview of the equations of motion, including projection and impulse formulations for the constant density case. This lays the framework for the derivation of the new auxiliary variable equations in Section 2.4.

We construct a numerical method with fourth-order accuracy in both space and time by coupling a new fourth-order discretization of the finite-volume formulation of the auxiliary variable equations with a temporal integration scheme based on spectral deferred corrections. A detailed description of the numerical method including the rationale for why spectral deferred corrections is used is given in Section 3.

In Section 4 a series of numerical tests are presented which demonstrate the accuracy of the numerical method for problems with non-trivial dynamics. The current implementation is limited to simple geometries in two dimensions. Also included is an accuracy comparison of the current method with the second-order, projection-type scheme described in [68].

Although the current paper describes only a method for zero-Mach number flows, the design of the numerical method is motivated by zero-Mach number asymptotics for the compressible flow equations in the spirit of [34,68,35,52], thereby laying the foundation for future extensions to zero and low Mach number reactive flows, atmospheric flows, etc. A discussion of the challenges involved in these extensions is taken up in Section 5.

2. Governing equations

The numerical method presented here is part of a larger project to develop higher-order numerical methods for the equations governing low-Mach number reacting flow. Although in the current study we consider only the much simpler case of zero-Mach number, non-reacting flows, we present the equations and the numerical method in a manner which most closely corresponds to the more general case.

2.1. The Euler equations and the zero-Mach number limit

Our starting point is given by the mass, momentum, and energy conservation laws for an ideal gas with constant specific heat capacities. Non-dimensionalizing using characteristic values for length, ℓ_{ref}, velocity, u_{ref}, pressure, p_{ref}, and density, ρ_{ref}, and scaling time by a typical advection time scale $t_{\text{ref}} = \ell_{\text{ref}}/u_{\text{ref}}$, we are left with a single dimensionless characteristic number, the Mach number

$$M = \frac{u_{\text{ref}}}{c_{\text{ref}}} \quad \text{where } c_{\text{ref}} = \sqrt{p_{\text{ref}}/\rho_{\text{ref}}}. \tag{1}$$

Then, the conservation laws for the density ρ, the momentum $\mathbf{m}$ and the energy e read

$$\rho_t = -\nabla \cdot (\rho \mathbf{v}), \tag{2}$$

$$\mathbf{m}_t = -\nabla \cdot \left(\rho \mathbf{v} \circ \mathbf{v} + \frac{1}{M^2} I p \right), \tag{3}$$

$$e_t = -\nabla \cdot ((e+p)\mathbf{v}), \tag{4}$$

where

$$e = \frac{p}{\gamma - 1} + M^2 \frac{\rho \mathbf{v} \cdot \mathbf{v}}{2} \tag{5}$$

is the equation of state needed to close the system in (2)–(4).

Following the asymptotic analysis in [34], we pass to the limit of vanishing Mach number while keeping the conservative form of the equations. The resulting limiting conservation laws are

$$\rho_t = -\nabla \cdot (\rho \mathbf{v}), \tag{6}$$

$$\mathbf{m}_t = -\nabla \cdot (\rho \mathbf{v} \circ \mathbf{v} + I p^{(2)}), \tag{7}$$

$$\dot{P}_0 = -\nabla \cdot (\gamma P_0 \mathbf{v}). \tag{8}$$

In these equations, $P_0(t)$ is the prescribed, spatially homogeneous leading order pressure.

Note that it is the energy equation (4) which becomes the velocity divergence constraint (8). Compliance with this constraint is guaranteed by an appropriate adjustment of $p^{(2)}$, which is the second-order contribution in an asymptotic expansion of the pressure p from (2)–(5) in terms of the Mach number. This variable is not

determined through an equation of state, but rather serves as a Lagrangian multiplier for the constrained problem. In a slight abuse of notation, we will refer to $p^{(2)}$ simply as p from here on.

Several prominent numerical methods for the zero-Mach number equations discussed in the following section are based on the non-conservative form of the zero-Mach number equations

$$\rho_t = -\nabla \cdot (\rho \mathbf{v}), \tag{9}$$

$$\mathbf{v}_t + \frac{1}{\rho}\nabla p = -(\mathbf{v} \cdot \nabla)\mathbf{v}, \tag{10}$$

$$\nabla \cdot \mathbf{v} = -\dot{P}_0/\gamma P_0. \tag{11}$$

In the remainder of this paper, we take $P_0(t) \equiv 1$, so that the velocity $\mathbf{v} = \mathbf{m}/\rho$ is divergence-free according to (8) or (11). The above equations can then be further simplified to

$$\rho_t = -(\mathbf{v} \cdot \nabla)\rho, \tag{12}$$

$$\mathbf{v}_t + \frac{1}{\rho}\nabla p = -(\mathbf{v} \cdot \nabla)\mathbf{v}, \tag{13}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{14}$$

2.2. Projection formulations

When the density in Eqs. (6)–(8) is constant in space and, with $P_0 \equiv 1$, also in time, Eqs. 10,11 simplify to the incompressible, constant density Euler equations, for which existing auxiliary variable and projection approaches were first developed. Without loss of generality, we let $\rho \equiv 1$, and therefore

$$\mathbf{v}_t + \nabla p = -(\mathbf{v} \cdot \nabla)\mathbf{v}, \tag{15}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{16}$$

One popular manner to facilitate the construction of numerical methods for Eqs. 15,16 is to cast them in the following *projection* formulation

$$\mathbf{v}_t = \mathbf{P}[-(\mathbf{v} \cdot \nabla)\mathbf{v}], \tag{17}$$

where $\mathbf{P}$ is the operator which projects a velocity field onto the space of divergence-free flows. $\mathbf{P}$ can be defined through the solution of a Poisson problem by $\mathbf{P}(\mathbf{w}) = \mathbf{w} - \nabla\phi$ where

$$\nabla^2\phi = \nabla \cdot \mathbf{w}. \tag{18}$$

In bounded domains, Neumann boundary conditions for Eq. (18) enforce a boundary condition on the normal component of the velocity.

The operator $\mathbf{P}$ can also be thought of as returning the divergence-free part of the Hodge or Helmholtz decomposition of a vector field. Hence, Eq. (17) results directly from applying $\mathbf{P}$ to both sides of Eq. (15). The significant feature of Eq. (17) is that the divergence constraint contained in Eq. (16) is implied by the projection and hence omitted.

Projection methods (see Section 3.1) utilize Eq. (17) to construct a numerical approximation by first computing an intermediate quantity (usually denoted $\mathbf{v}^*$) using Eq. (15), without regard to the divergence constraint (16), and then using a discretized form of the projection operator $\mathbf{P}$ to extract the discretely divergence-free (or approximately so) velocity from $\mathbf{v}^*$.

A common strategy for extending projection methods designed for the constant density equations to variable density is to begin with the form of the Eqs. (9)–(11), see, e.g., [7,53,54]. The left hand side of the equation for $\mathbf{v}$ (either Eq. (10) or (13)) is no longer in the form of the standard Helmholtz decomposition, but this equation can still be transformed into a projection form analogous to Eq. (17). Let the operator $\mathbf{P}_\rho(\mathbf{w})$ be defined by

$$\mathbf{P}_\rho(\mathbf{w}) = \mathbf{w} - \frac{\nabla\phi}{\rho}, \tag{19}$$

where ϕ is the solution of

$$\nabla \cdot \left(\frac{\nabla \phi}{\rho} \right) = \nabla \cdot (\mathbf{w}). \tag{20}$$

By construction, the quantity $\mathbf{v} = \mathbf{P}_\rho(\mathbf{w})$ satisfies the divergence constraint (11), and hence applying $\mathbf{P}_\rho$ to both sides of Eq. (13) yields the variable density analog with Eq. (17)

$$\rho_t = -(\mathbf{v} \cdot \nabla)\rho, \tag{21}$$

$$\mathbf{v}_t = \mathbf{P}_\rho[-(\mathbf{v} \cdot \nabla)\mathbf{v}]. \tag{22}$$

Note that in both the constant density and variable density projection formulations above, the projection operator (either $\mathbf{P}$ or $\mathbf{P}_\rho$) enforces a boundary condition on the normal component of the velocity at domain boundaries.

2.3. Gauge, impulse, and auxiliary variables

In order to develop numerical methods with higher temporal order of accuracy, it is useful to consider an analog to the auxiliary variable formulation of the Euler equations. Auxiliary variables are closely related to a class of equations first introduced by Oseledets [57] which provide an equivalent form of the Euler equations in terms of a variable which does not satisfy the divergence constraint but from which the incompressible velocity can be recovered (by way of the Hodge or Helmholtz decomposition). Different formulations and names for such variables have been proposed by different authors including velocity, magnetization, impetus, impulse, impulse density, and gauge variables [12,13,46,21,22,25,26]. For the remainder of this paper, we adapt the language of [66] and use the terminology *impulse variables* for such variables. Both Lagrangian [12,20,21,63,71] and Eulerian [26,25,50] numerical methods based on impulse variables have been developed for various problems although to date this idea has never been adapted to variable density or low-Mach number flows.

A general overview is given in [66] in which a form of the equation for impulse variables is presented which contains an arbitrary gauge. Specifically, the starting point is the equivalent form of Eq. (15)

$$\mathbf{v}_t = \mathbf{v} \times (\nabla \times \mathbf{v}) - \nabla \left(p + \frac{1}{2}\mathbf{v} \cdot \mathbf{v} \right). \tag{23}$$

Then, defining a new variable $\mathbf{v}^* = \mathbf{v} + \nabla\phi$, an equivalent equation is

$$\mathbf{v}^*_t = \mathbf{v} \times (\nabla \times \mathbf{v}^*) - \nabla \left(p + \frac{1}{2}\mathbf{v} \cdot \mathbf{v} - \frac{\partial \phi}{\partial t} \right). \tag{24}$$

Since $\nabla\phi$ is arbitrary, this can be rewritten simply as

$$\mathbf{v}^*_t = \mathbf{v} \times (\nabla \times \mathbf{v}^*) + \nabla \Lambda, \tag{25}$$

where Λ is the *gauge*. The pertinent point is that for any choice of gauge Λ, given suitable initial and boundary conditions, $\mathbf{P}(\mathbf{v}^*) = \mathbf{v}$ for all time where $\mathbf{v}$ is the solution to the incompressible Euler Eqs. (15) and (16). Furthermore, the correct pressure can be recovered from the solution of Eq. (25) by

$$\frac{\partial \phi}{\partial t} = p + \frac{1}{2}\mathbf{u} \cdot \mathbf{u} + \Lambda. \tag{26}$$

The appropriate choice of gauge necessary to create many of the governing equations from the references cited above is also presented in [66].

Following the construction for viscous flows in [50], a connection can be made between projection methods and impulse variables. Given initial conditions $\mathbf{v}(\mathbf{x}, 0)$ for the Euler equations (15) and (16), let the solution be denoted $\mathbf{v}(\mathbf{x}, t)$ with pressure $p(\mathbf{x}, t)$. Now let $\mathbf{u}^*(\mathbf{x}, 0) = \mathbf{v}(\mathbf{x}, 0)$, and consider the equation

$$\mathbf{u}^*_t = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla q, \tag{27}$$

where ∇q is an explicitly prescribed approximation to the pressure gradient ∇p, and

$$\mathbf{u} = \mathbf{P}(\mathbf{u}^*).$$

Clearly if $\nabla q = \nabla p$ then $\mathbf{u}^* = \mathbf{u} = \mathbf{v}$; however, ∇p is in general not known *apriori*. Nevertheless, it follows that $\mathbf{u} = \mathbf{P}(\mathbf{u}^*) = \mathbf{v}$ regardless of how well ∇q approximates ∇p since Eq. (27) is an impulse formulation in the language of [66] with gauge

$$\Lambda = -\frac{1}{2}\mathbf{u}\cdot\mathbf{u} - q.$$

Therefore the pressure can also be reconstructed from

$$\nabla p = \nabla q + \nabla \phi_t, \tag{28}$$

where $\mathbf{u}^* = \mathbf{u} + \nabla\phi$.

In [11], it is observed that rather than thinking of a single time step of a projection method as a fractional step scheme with a temporary numerical variable called $\mathbf{v}^*$, one can instead consider the method as a consistent disretization of an *auxiliary variable equation* with $\mathbf{v}^*$ satisfying (27). This point of view then gives a precise mathematical description of the proper boundary conditions for the variable $\mathbf{v}^*$, a subject that has generated a great deal of controversy in the literature. Section 3.1 details how auxiliary variables are combined with the method of lines to construct methods with higher-order temporal accuracy.

2.4. Auxiliary variables for the zero-Mach equations

In the previous section, a connection is made between the projection formulation of the constant density Euler equations, impulse equations, and auxiliary variable equations. To our knowledge, such observations have only to date been made for constant density flows. We present a variable density analog to the auxiliary variable equations now.

Consider forming a set of auxiliary variable equations based on Eqs. (6) and (7). To this end, define an analog to the variable density projection operator $\mathbf{P}_\rho$ which acts on momentum variables rather than velocity variables. Let $\mathbf{Q}_\rho(\mathbf{w}) = \mathbf{w} - \nabla\phi$ where

$$\nabla\cdot\left(\frac{\nabla\phi}{\rho}\right) = \nabla\cdot\left(\frac{\mathbf{w}}{\rho}\right). \tag{29}$$

Now let

$$\rho_t = -\nabla\cdot(\mathbf{v}\rho), \tag{30}$$

$$\mathbf{m}^*_t = -\nabla\cdot(\rho\mathbf{v}\circ\mathbf{v} + Iq), \tag{31}$$

where

$$\mathbf{v} = \frac{\mathbf{Q}_\rho(\mathbf{m}^*)}{\rho}. \tag{32}$$

Again it is clear that choosing $\nabla q = \nabla p$ would imply $\mathbf{m}^* = \mathbf{m}$.

When $\nabla q \neq \nabla p$ however, $\mathbf{m}^* \neq \mathbf{m}$, but as in the constant density case, $\mathbf{m}$ and hence $\mathbf{v}$ can be derived from $\mathbf{m}^*$ through the variable density projection $\mathbf{Q}_\rho$. Let

$$\mathbf{m}^*_t = \mathbf{m}_t + \nabla\chi, \tag{33}$$

where $\nabla\chi = \nabla(p - q)$. Then

$$\mathbf{m} = \mathbf{Q}_\rho(\mathbf{m}^*) = \mathbf{m}^* - \nabla\phi, \tag{34}$$

where

$$\nabla\phi = \int_{t_0}^{t} \nabla\chi\,\mathrm{d}\tau. \tag{35}$$

This is simply an integral form of Eq. (28). Note also, that in the presence of solid wall boundaries, the projection $\mathbf{Q}_\rho$ imposes a normal boundary condition on $\mathbf{m}$, and hence there is an additional freedom in the boundary condition for $\mathbf{m}^*$. In the numerical method discussed below, no boundary condition for $\mathbf{m}^*$ is imposed (as is

also the case with the density ρ). A discussion of the numerical treatment of the boundary conditions appears in Section 3.2.4.

3. The numerical method

In this section we present an overview of the different concepts synthesized to create the numerical method. An overview of auxiliary variable projection methods, spectral deferred corrections, and finite-volume methods are first presented, followed by a concise outline of the numerical method in Section 3.4. Finally, a discussion comparing and contrasting alternative strategies to that employed here is presented.

3.1. Auxiliary variable projection methods

In this section, we first review the connection between projection methods and auxiliary variable projection methods for constant density flows. Then a discussion of auxiliary variable projection methods for the zero-Mach equations is presented. In this section the equations are discretized in time only leaving the spatial derivative operators continuous. The spatial discretization is taken up in Section 3.2.

3.1.1. Constant density auxiliary variable projection methods

Projection methods, which were introduced in the late 1960s [17,16,72], are a popular class of methods for numerically approximating the equations of incompressible flow. This popularity stems largely from the fact that, in projection methods, the numerical approximation to the time evolution of the momentum equation (e.g. Eq. (15)) is decoupled from the numerical procedure to enforce the divergence constraint (e.g. Eq. (16)). The basic strategy by which this is accomplished is as follows. To advance the solution for the velocity $\mathbf{v}$, a suitable discretization of Eq. (15) is used without regard to the divergence constraint Eq. (16). This yields a provisional update, which for reasons that should become clear, we denote $\mathbf{v}^*$. Then a discretization of the projection operator $\mathbf{P}$ is applied to $\mathbf{v}^*$ to enforce Eq. (16) and provide the temporal update to $\mathbf{v}$. For this reason projection methods are often referred to as *fractional step* methods.

To illustrate, we present a first-order temporal approximation to the Eqs. (15) and (16). Let $\mathbf{v}^n$ denote the numerical approximation to the velocity at time t_n, and consider the following method for computing $\mathbf{v}^{n+1}$. Denoting $\Delta t = t_{n+1} - t_n$, let

$$\mathbf{v}^* = \mathbf{v}^n + \Delta t[-(\mathbf{v}^n \cdot \nabla)\mathbf{v}^n - \nabla p^n] \tag{36}$$

and

$$\mathbf{v}^{n+1} = \mathbf{P}(\mathbf{v}^*). \tag{37}$$

The latter equation is equivalent to

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \nabla\phi, \tag{38}$$

where

$$\nabla^2\phi = \nabla \cdot \mathbf{v}^*. \tag{39}$$

Finally, a first-order temporal approximation to ∇p^{n+1} is given by

$$\nabla p^{n+1} = \nabla p^n + \frac{\nabla\phi}{\Delta t}. \tag{40}$$

The above procedure, wherein the nonlinear advective terms are treated explicitly, requires only the solution of a standard Poisson problem to enforce the divergence constraint on the velocity $\mathbf{v}^{n+1}$.

In [11], it is observed that rather than thinking of a single time step of a projection method as a fractional step scheme with auxiliary variable $\mathbf{v}^*$, one can instead consider the projection method as a consistent discretization of an *auxiliary variable* equation where the variable $\mathbf{v}^*$ satisfies an appropriate gauge formulation. Unlike in a direct discretization of an impulse equation, in a projection method $\mathbf{v}^*$ is reset to the value $\mathbf{v}$ (i.e. $\nabla\phi$ is reset to 0) at the beginning of each time step.

To make this connection clear, consider a first-order temporal discretization to Eq. (27). Let $\mathbf{u}^{*,n}$ denote the auxiliary variable at the beginning of the time step with $\mathbf{u}^{*,n} = \mathbf{v}^n$ where $\mathbf{v}^n$ is some divergence-free approximation to the incompressible Euler equations. Since $\mathbf{u}^{*,n}$ is divergence-free, $\mathbf{u}^{*,n} = \mathbf{P}(\mathbf{u}^{*,n}) = \mathbf{u}^n$ and $\nabla\phi^n = 0$. Furthermore, let $\nabla q = \nabla p^n$, be the arbitrary gradient term in Eq. (27) (which is in this case constant in time). Then a first-order auxiliary variable projection method is

$$\mathbf{u}^{*,n+1} = \mathbf{u}^{*,n} + \Delta t[-(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n - \nabla q^n]. \tag{41}$$

By definition

$$\mathbf{u}^{n+1} = \mathbf{P}(\mathbf{u}^{*,n+1}), \tag{42}$$

which is equivalent to

$$\mathbf{u}^{n+1} = \mathbf{u}^{*,n+1} - \nabla\phi^{n+1}, \tag{43}$$

where

$$\nabla^2\phi^{n+1} = \nabla \cdot \mathbf{u}^{*,n+1}. \tag{44}$$

Finally, using Eq. (28), a first-order temporal approximation to ∇p^{n+1} is given by

$$\nabla p^{n+1} = \nabla q^n + \frac{\nabla\phi^{n+1} - \nabla\phi^n}{\Delta t}. \tag{45}$$

Recalling that $\nabla\phi^n = 0$, and $\nabla q^n = \nabla p^n$, we see Eqs. (41)–(45) are identical to Eqs. (36)–(40). However, in a projection or auxiliary variable method, the next time step would again start with $\mathbf{u}^{*,n+1} = \mathbf{v}^{n+1}$ (i.e. the U^* variable is *reset*), whereas in an impulse variable formulation, $\mathbf{u}^*$ is allowed to evolve as an independent variable. As discussed in [66], in many cases, the evolution of impulse variables leads to a concentration of impulse along sheets which causes a lack of resolution for numerical methods.

For the first-order method above, the difference between projection methods and auxiliary variable methods is largely semantic. However, the advantage of using an auxiliary variable approach over a traditional fractional step projection method becomes clear when one considers a higher-order temporal discretization method. In the auxiliary variable approach, higher-order temporal methods can be directly applied to the equation for $\mathbf{u}^*$ by using the impulse variable equations and the methods of lines, since the auxiliary variable is not subject to a divergence constraint. This avoids the complication of finding a temporally higher-order formulation of the fractional-step approach used in projection methods. In practice one could use any higher-order temporal integration scheme in the auxiliary variable approach, however, the choice of using spectral deferred corrections has certain advantages discussed below in Section 3.3. This strategy is exploited in [50,48] to construct auxiliary variable projection methods with up to sixth-order temporal accuracy for constant density flows. The same idea is also used in [45,65].

3.1.2. zero-Mach auxiliary variable projection methods

The numerical method for the zero-Mach equations presented in this paper is based on an extension of the constant density auxiliary variable method presented above. In a given time step, one begins with an approximation to ∇p embodied in ∇q and then solves the auxiliary variable equations 30,31 in conservation form using a spectral deferred correction method which is described in Section 3.3.

A first-order auxiliary variable projection method is defined as follows. Given values of ρ^n, $\mathbf{m}^n$, and q^n, respectively density, momentum and a pressure approximation at time t_n, let $\mathbf{m}^{*,n} = \mathbf{m}^n$. Then

$$\rho^{n+1} = \rho^n + \Delta t[-\nabla \cdot (\rho^n\mathbf{v}^n)], \tag{46}$$

$$\mathbf{m}^{*,n+1} = \mathbf{m}^{*,n} + \Delta t[-\nabla \cdot (\rho^n\mathbf{v}^n \circ \mathbf{v}^n + Iq^n)]. \tag{47}$$

By definition

$$\mathbf{v}^{n+1} = \frac{\mathbf{Q}_{\rho^{n+1}}(\mathbf{m}^{*,n+1})}{\rho^{n+1}}, \tag{48}$$

which is equivalent to

$$\mathbf{m}^{n+1} = \mathbf{m}^{*,n+1} - \nabla\phi^{n+1}, \tag{49}$$

where

$$\nabla \cdot \left(\frac{\nabla\phi^{n+1}}{\rho^{n+1}}\right) = \nabla \cdot \left(\frac{\mathbf{m}^{*,n+1}}{\rho^{n+1}}\right). \tag{50}$$

At solid wall boundaries, Neumann boundary conditions are enforced on $\nabla\phi^{n+1}$ in the solution of Eq. (50) so that the normal component of $\mathbf{m}^{n+1}$ is zero at the boundary. No conditions are enforced on either $\mathbf{m}^{*,n+1}$ or ρ^{n+1}. The spatial discretization of boundary conditions is discussed in Section 3.2.4. Finally, using Eq. (28), a first-order temporal approximation to ∇p^{n+1} is given by

$$\nabla p^{n+1} = \nabla q^n + \frac{\nabla\phi^{n+1}}{\Delta t}. \tag{51}$$

This first-order procedure serves as the predictor step of the deferred correction method described in Section 3.3. A very similar procedure is used in the correction iterations. The SDC method produces accurate values of $\mathbf{m}^*$ and ρ, at intermediate substeps from which values of $\mathbf{v}$ and $\nabla\phi$ are computed using $\mathbf{Q}_\rho$. Once the deferred correction iterations are complete, the pressure is updated using the intermediate values of $\nabla\phi$ and a fourth-order accurate approximation of Eq. (35). Finally, the value of $\mathbf{m}^{*,n+1}$ is reset by subtracting $\nabla\phi^{n+1}$ to produce the starting value for the next time step.

At the beginning of each time step, the value of the explicit function $q(x,t)$ needs to be determined. As is the case for impulse variable methods for the incompressible Euler equations, in the current setting the extent to which q approximates the pressure p affects the accuracy of the numerical solution very little. As explained in [11], the accuracy of q as an approximation to p does play a role when choosing the boundary conditions for the auxiliary variable if viscous terms (and hence a no-slip boundary condition) are included in the equations. Here q is assigned the value of the pressure at the beginning of the time step, namely p^n. As explained in Section 3.3, since q is held constant during the SDC iterations, the explicit form of q only appears in the predictor phase.

3.2. Finite-volume discretization

Many popular methods in use today for the zero-Mach number equations utilize a discretization strategy which is closely related to methods first developed for hyperbolic conservations laws in general, and the compressible Euler equations in particular. One particularly successful strategy is based on a finite-volume discretization of the equations of motion in conservation form for mass, momentum, and total energy, [67,68,76]. This approach utilizes the cell-average values of physical quantities as the discretization variables which are governed by an equation involving cell edge fluxes. Numerical methods based on a finite-volume discretization update cell averages by computing an approximation to the time–space integral of the flux function at cell edges. A popular class of methods in use today employs a second-order spatial and temporal discretization based on an extension of Godunov's method (see e.g. [75,18,68,76]).

In such second-order methods, the time–space integral of the flux is in essence approximated by the midpoint rule. More specifically, the goal is to compute time and space centered values for the flux function which are used to approximate the integral. In order to achieve fourth-order spatial and temporal accuracy, the method here combines a spectral deferred corrections approach in time (which can be thought of as an approximation to the Gauss–Labatto quadrature rule) with a rather straightforward numerical quadrature of the cell-edge fluxes. This approach is only appropriate for locally Cartesian grids and smooth solutions. Extending this procedure to robustly treat non-smooth solutions using limiters as in the PPM method [19] or ENO-type flux calculations [69,70] is a topic of current research.

The main obstacle in extending the finite-volume approach for hyperbolic problems to the zero-Mach number case is that the advective velocities used to calculate cell edge fluxes must be adjusted so that the divergence constraint is satisfied. A popular strategy is to apply a discrete projection operator to cell edge velocities which

is equivalent to including an approximation to the non-local evolution of the pressure term [7,62,1,67,68]. The numerical projection is discussed in Section 3.2.5.

3.2.1. Notation and formulation

To facilitate the explanation, several notational conventions are first introduced. For ease of presentation, we assume that the physical domain is two-dimensional and is divided into a uniform array of cells of width and height h. Let the cell with center at (x_i, y_j) be denoted by the pair (i,j), and let the half integer subscripts $i+1/2$ and $j+1/2$ denote a shift by distance $h/2$ in the x- and y-direction respectively. The extension to rectangular cells and three dimensions is straightforward.

The finite-volume approach is based on an evolution equation for cell averages. For some quantity such as $\rho(x,y,t)$, define

$$\bar{\rho}(x_i,y_j,t) = \frac{1}{h^2}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} \rho(x,y,t)\mathrm{d}y\,\mathrm{d}x. \tag{52}$$

Similarly, let the average of a quantity over a cell edge be denoted by a tilde. Specifically,

$$\tilde{\rho}(x_{i+1/2},y_j,t) = \frac{1}{h}\int_{y_{j-1/2}}^{y_{j+1/2}} \rho(x_{i+1/2},y,t)\mathrm{d}y \tag{53}$$

and

$$\tilde{\rho}(x_i,y_{j+1/2},t) = \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x,y_{j+1/2},t)\mathrm{d}x. \tag{54}$$

As a further notational convenience, we also use a tilde without index shifting when referring to the cell-edge averages of a vector quantity when the first component of the vector is averaged over the $(i+1/2,j)$ cell edge and the second component is averaged over the $(i,j+1/2)$ edge. This convention will also be followed for gradients, hence for example,

$$\nabla\tilde{\phi}(x_i,y_j,t) = (\tilde{\phi}_x(x_{i+1/2},y_j,t), \tilde{\phi}_y(x_i,y_{j+1/2},t)). \tag{55}$$

To specify the finite-volume formulation of the conservation law

$$Q_t + \nabla\cdot F(Q) = 0, \tag{56}$$

where $Q(x,y,t)$ is the vector of conserved quantities and $F(Q) = (f(Q), g(Q))$ is the flux function, we integrate the equation over the computational cells and use the divergence theorem to attain

$$\overline{Q}_t(x_i,y_j,t) + \frac{1}{h^2}\int_{\square i,j} F(Q(x,y,t)) = 0, \tag{57}$$

where the flux integral above is defined as

$$\begin{aligned}\int_{\square i,j} F(Q) = &\int_{y_{j-1/2}}^{y_{j+1/2}} f(Q(x_{i+1/2},y,t)) - f(Q(x_{i-1/2},y,t))\mathrm{d}y + \int_{x_{i-1/2}}^{x_{i+1/2}} g(Q(x,y_{j+1/2},t))\\ &- g(Q(x,y_{j-1/2},t))\mathrm{d}x.\end{aligned} \tag{58}$$

Using the definitions of the edge average in Eq. (53) and (54)

$$\int_{\square i,j} F(Q(x,y,t)) = h(\tilde{f}(Q(x_{i+1/2},y_j,t)) - \tilde{f}(Q(x_{i-1/2},y_j,t))) + h(\tilde{g}(Q(x_i,y_{j+1/2})) - \tilde{g}(Q(x_i,y_{j-1/2}))). \tag{59}$$

Therefore,

$$\overline{Q}_t(x_i,y_j,t) + \frac{\tilde{f}(Q(x_{i+1/2},y_j,t)) - \tilde{f}(Q(x_{i-1/2},y_j,t))}{h} + \frac{\tilde{g}(Q(x_i,y_{j+1/2})) - \tilde{g}(Q(x_i,y_{j-1/2}))}{h} = 0. \tag{60}$$

Since Eq. (60) resembles a finite-difference approximation to Eq. (56), we introduce an additional notational convention by defining $\tilde{\nabla}\cdot$ as

$$\tilde{\nabla}\cdot\tilde{F}(Q(x_i,y_j,t))=\frac{1}{h^2}\int_{\square i,j}F(Q(x,y,t)). \tag{61}$$

Hence, by suppressing the (i,j) indices we arrive at

$$\overline{Q}_t+\tilde{\nabla}\cdot\tilde{F}(Q)=0. \tag{62}$$

It is important to note that the various forms of the equations above are mathematically exact, i.e. no numerical approximations have been introduced up to this point.

In describing the discretization of Eq. (62), numerical approximations are distinguished by the use of subscripts and superscripts. For example, the approximation to the cell-centered value $\rho(x_i,y_j,t_n)$ will be denoted ρ^n_{ij}. Superscripts or subscripts are suppressed when the meaning is apparent. This convention carries over to cell- or edge-averaged quantities as well, e.g.

$$\tilde{\rho}^n_{i+1/2,j}\approx\tilde{\rho}(x_{i+1/2},y_j,t_n).$$

In order to implement a numerical method based on the above finite-volume formulation, given cell average quantities $\overline{Q}$, it is necessary to construct an accurate approximation of the edge averages of the flux function, namely $\tilde{F}(Q)$. Our approach proceeds in two separate steps:

(1) Computing the edge averages of conserved quantities $\tilde{Q}$ from the cell averages $\overline{Q}$.
(2) Computing the edge averages of the flux function $\tilde{F}(Q)$ from the edge averages $\tilde{Q}$.

3.2.2. Computing edge averages of conserved variables

Given cell average values $\bar{\phi}$, a fourth-order approximation to $\tilde{\phi}$ at the left cell edge is given by

$$\tilde{\phi}_{i+1/2,j}=\frac{-\bar{\phi}_{i-1,j}+7(\bar{\phi}_{i,j}+\bar{\phi}_{i+1,j})-\bar{\phi}_{i+2,j}}{12}. \tag{63}$$

This approximation is derived by simply integrating the standard one-dimensional formula in the y-direction.

The one-dimensional version of this equation forms the basis of the piecewise parabolic method (PPM) [19] as well as the conservative one-dimensional method based on deferred corrections presented in [40]. In both of these papers, limiters are employed in the presence of discontinuities or sharp gradients in the solution which locally reduces the order of accuracy of the numerical method in favor of avoiding oscillations in the numerical solution. Limiters then produce two distinct cell-edge values which are considered the left and right states of a Riemann problem whose solution defines the edge value. We have experimented with the use of limiters and Riemann problems in two dimensions for the current method as well, but since the current work focuses on demonstrating higher-order spatial and temporal accuracy, the results will be presented in the future.

3.2.3. Computing edge averages of the fluxes

In order to compute higher-order accurate values of edge averages of the flux functions, it is necessary to be able to compute the edge average of the products and quotients of variables appearing in Eqs. (30)–(32). The primary difficulty in building higher-order finite-volume methods is that the edge average of a product (or quotient) is not equal to the product (or quotient) of edge averages.

The goal here is to express edge averages of quotients or products as the quotient or product of the edge averages plus a correction term which must be computed by applying a stencil to neighboring edge averages. To derive a stencil for the correction we first form piecewise polynomial interpolants to a quantity given edge averages. For example, the expansion for an arbitrary quantity ϕ on the cell edge is given by the power series

$$\phi(x_{i+1/2},y)\approx\phi_0+\phi_1(y-y_j)+\phi_2(y-y_j)^2\cdots \tag{64}$$

The coefficients ϕ_k, are computed using difference stencils discussed below. Given a similar expansion for the quantity ρ, using the exact integral of the series expansion for the quotient of polynomials yields the following stencil for the edge average of a quotient.

$$\left(\widetilde{\frac{\phi}{\rho}}\right)_{i+1/2,j} = \frac{\tilde{\phi}_{i+1/2,j}}{\tilde{\rho}_{i+1/2,j}} + \frac{h^2}{12}\left(\phi_0\frac{\rho_1^2}{\rho_0^3} - \phi_1\frac{\rho_1}{\rho_0^2}\right) + \frac{h^4}{720}\left(\phi_0\frac{9\rho_0\rho_4 - 5\rho_2^2}{\rho_0^3} + \phi_2\frac{5\rho_2}{\rho_0^2} - \phi_4\frac{9}{\rho_0}\right) + \mathrm{O}(h^6). \tag{65}$$

Suppressing the indices, and using the fact that

$$\tilde{\phi} = \phi_0 + \frac{h^2}{12}\phi_2 + \frac{h^4}{80}\phi_4 + \mathrm{O}(h^6) \tag{66}$$

(and likewise for $\tilde{\rho}$), we can eliminate the ϕ_0 and ρ_0 terms and write

$$\left(\widetilde{\frac{\phi}{\rho}}\right) = \frac{\tilde{\phi}}{\tilde{\rho}} + \frac{h^2}{12}\left(\frac{\tilde{\phi}\rho_1^2}{\tilde{\rho}^3} - \frac{\phi_1\rho_1}{\tilde{\rho}^2}\right) + \mathrm{O}(h^4). \tag{67}$$

Similarly,

$$(\widetilde{\phi\rho}) = \tilde{\phi}\tilde{\rho} + \frac{h^2}{12}\phi_1\rho_1 + \mathrm{O}(h^4). \tag{68}$$

The stencils approximating the polynomial coefficients are as follows:

$$\phi_1 = \frac{-5\tilde{\phi}_{i+1/2,j+2} + 34(\tilde{\phi}_{i+1/2,j+1} - \tilde{\phi}_{i+1/2,j-1}) + 5\tilde{\phi}_{i+1/2,j-2}}{48h},$$
$$\phi_2 = \frac{-\tilde{\phi}_{i+1/2,j+2} + 12\tilde{\phi}_{i+1/2,j+1} - 22\tilde{\phi}_{i+1/2,j} + 12\tilde{\phi}_{i+1/2,j-1} - \tilde{\phi}_{i+1/2,j-2}}{16h^2}.$$

These formulas can also be "limited" in the same sense as the PPM procedure in areas in which the solution is non-smooth.

During the projection step of the algorithm (see Section 3.2.5) it is also necessary to compute edge averages of spatial derivatives. To compute the average over an edge of a derivative normal to the edge using cell averages we use the fourth-order formula

$$(\tilde{\phi}_x)_{i+1/2,j} = \frac{\bar{\phi}_{i-1,j} + 15(-\bar{\phi}_{i,j} + \bar{\phi}_{i+1,j}) - \bar{\phi}_{i+2,j}}{12h}. \tag{69}$$

Note that the cell average of a derivative is analytically equal to the centered difference of edge averages, i.e.

$$(\bar{\phi}_x)_{i,j} = \frac{\tilde{\phi}_{i+1/2,j} - \tilde{\phi}_{i-1/2,j}}{h}. \tag{70}$$

Substituting the edge average formula gives

$$(\bar{\phi}_x)_{i,j} = \frac{\bar{\phi}_{i-2,j} + 8(-\bar{\phi}_{i-1,j} + \bar{\phi}_{i+1,j}) - \bar{\phi}_{i+2,j}}{12h}, \tag{71}$$

which is the same as the fourth-order finite-difference formula for the derivative.

3.2.4. Stencils near the computational boundary

In the implementation, we employ ghost cells around the numerical domain so that the above stencils can be used at all cells and cell edges inside the domain. In all cases, two ghost cells must be assigned values, and we consider here only periodic or solid wall boundary conditions. For periodic boundary conditions, the value in the cells are set using periodicity, but in the case of solid wall boundary conditions, the procedure for setting ghost cell values depends on what is known about the numerical quantity for which the ghost cell is being set. Three distinct cases occur: the value of the quantity is known at the boundary, the normal derivative of the quantity is known at the boundary, or no information about the quantity is known at the boundary, in which case the value at the boundary is extrapolated. In general, the first of the two constraints that define the value of the two ghost cells comes from the known boundary conditions and the second from an extrapolation condition.

For example, consider the setting of ghost cells before the cell-edge averages of the conserved quantities are computed using Eq. (63). Fig. 1, shows the case in the lower y-direction. Assume that the value of the cell average of some quantity ψ is known at the boundary (denoted $\tilde{\psi}_{i,j-1/2}$), either from the prescribed boundary conditions, or from extrapolation from the interior. One constraint on the ghost-cells is that applying Eq. (63) gives the correct value for $\tilde{\psi}_{i,j-1/2}$, i.e.

$$\tilde{\psi}_{i,j-1/2} = \frac{-\bar{\psi}_{i,j-2} + 7\bar{\psi}_{i,j-1} + 7\bar{\psi}_{i,j} - \bar{\psi}_{i,j+1}}{12}. \tag{72}$$

The second condition comes from a fourth-order extrapolation stencil,

$$-\bar{\psi}_{i,j-2} + 4\bar{\psi}_{i,j-1} - 6\bar{\psi}_{i,j} + 4\bar{\psi}_{i,j+1} - \bar{\psi}_{i,j+2} = 0. \tag{73}$$

Solving these two equations for $\bar{\psi}_{i,j-1}$ gives

$$\bar{\psi}_{i,j-1} = \frac{12\tilde{\psi}_{i,j-1/2} - 13\bar{\psi}_{i,j} + 5\bar{\psi}_{i,j+1} - 1\bar{\psi}_{i,j+2}}{3}. \tag{74}$$

The value of $\bar{\psi}_{i,j-2}$ can then be set using Eq. (73).

In the case that the normal derivative of the quantity is prescribed at the cell edge (as for ϕ in the projection), the above procedure is modified to make use of this condition. Let $(\tilde{\psi}_x)_{i,j-1/2}$, be the normal derivative at the boundary. Then Eq. (72) is replaced by enforcing Eq. (69), i.e.

$$(\tilde{\psi}_x)_{i,j-1/2} = \frac{\tilde{\psi}_{i,j-2} - 15\tilde{\psi}_{i,j-1} + 15\tilde{\psi}_{i,j} - \tilde{\psi}_{i,j+1}}{12\Delta y} \tag{75}$$

and a fifth-order extrapolation stencil.

It is also necessary at times to set ghost values of cell-edge average quantities, for example in the computation of the stencils in Eqs. (67) and (68). Fig. 2 shows the layout of the data in this case. The same procedure described above is used to compute cell-edge ghost cells simply by replacing known edge-average values at the boundary ($\tilde{\psi}_{i,j-1/2}$ or $(\tilde{\psi}_x)_{i,j-1/2}$), by point values at cell corner on the boundary ($\psi_{i+1/2,j-1/2}$ or $(\psi_y)_{i+1/2,j-1/2}$).

3.2.5. The numerical projection

In methods for the zero-Mach equations based on schemes originally designed for hyperbolic problems, cell edge fluxes must be modified to account for the divergence constraint. A common strategy is to first calculate time-centered quantities, and then to apply a variable density projection to exactly enforce a local discrete

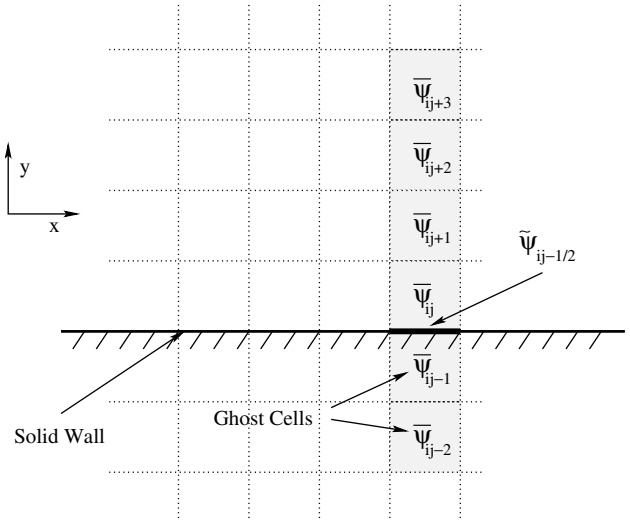

Fig. 1. Ghost cells at solid wall boundary in lower y-direction.

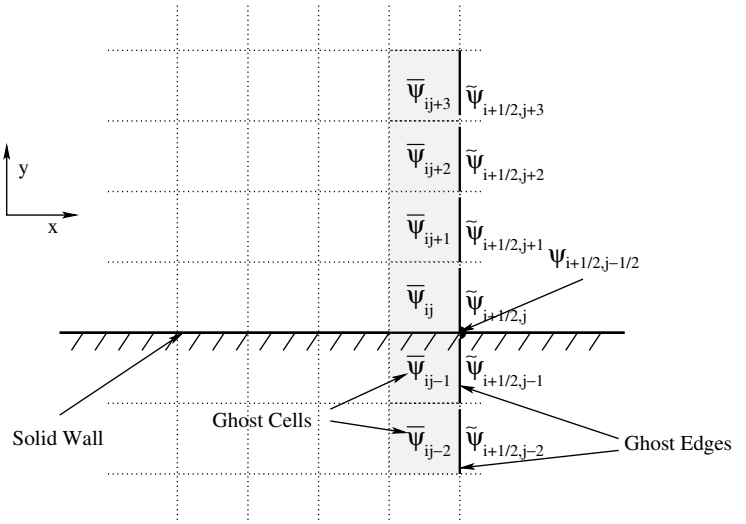

Fig. 2. Ghost cell edge values at solid wall boundary in lower y-direction.

divergence constraint [7,62,1,67,68,76]. This has the attractive features that nonlinear terms can be computed in conservation form and that advected quantities which are constant in some region will remain locally constant (often referred to as *free-stream preservation*). In addition to the projection on time-centered cell edge velocities, the methods cited above also utilize a second projection step applied to the updated cell average values in order to again enforce the divergence constraint as well as provide an update to the pressure term. The schemes in [7,67,68,76] include an exact second projection step, i.e., their cell-centered velocity divergences are controlled up to the error of the elliptic pressure solves associated with the projections. In addition, Vater's scheme, [76], features a compact stencil, and its projection step is inf-sup-stable.

In the current method, we use a discrete version of the variable density projection $\mathbf{Q}_\rho$ defined in Section 2.4. This requires solving a discrete version of Eq. (29).

$$\tilde{\nabla}\cdot\left(\frac{\widetilde{\nabla\phi}}{\rho}\right)=\tilde{\nabla}\cdot\tilde{\mathbf{u}}^*, \tag{76}$$

where

$$\tilde{\mathbf{u}}^*=\frac{\mathbf{m}^*}{\rho} \tag{77}$$

is computed using the division formula (67). Then

$$\tilde{\mathbf{u}}=\tilde{\mathbf{u}}^*-\left(\frac{\widetilde{\nabla\phi}}{\rho}\right) \tag{78}$$

is divergence-free.

In the left hand side of Eq. (76), the divergence operator acts on the cell edge average of $\nabla\phi/\rho$. This is defined by using Eq. (69) and the division formula (67), hence the variable being solved for is the cell average $\bar{\phi}$. This yields a 25 point stencil for the discrete operator. When solid wall boundary conditions are present, the boundary condition

$$(\tilde{\mathbf{m}}^*-\tilde{\nabla}\bar{\phi})\cdot\hat{n}=0 \tag{79}$$

is enforced.

Eq. (76) is solved using a standard multigrid approach. Gauss–Seidel relaxation is used as a relaxation scheme and a reduction in the magnitude of the residual of approximately a factor of ten for each V-cycle

is observed. Note that during the later SDC iterations, the value of $\bar{\phi}$ from the previous SDC iteration is a very good initial guess for the multigrid solver and fewer iterations are required than during the prediction sweep.

Note that in the parlance of projection methods, we are using an "exact" projection, i.e. the velocities satisfy a discrete divergence constraint up to the accuracy of the elliptic solver. However, in the finite-volume discretization, both the normal and tangential velocities are required at each cell edge. The tangential velocities at cell edges are projected by subtracting a centered average of the appropriate derivative of ϕ. The tangential velocities do not satisfy any discrete divergence constraint.

One important consequence of using the deferred correction strategy is that the numerical variables representing cell averages and cell edge fluxes are always computed at the same time level (i.e. there is no time-centering of fluxes as in most second-order Godunov type schemes). In the context of zero-Mach number flows, the implication of this is that the second type of projection applied to cell averages which is discussed above is not needed to enforce a divergence constraint.

3.3. Spectral deferred corrections

In [24], a variant of the traditional deferred and defect correction methods [78,61,60] is presented which allows the construction of stable numerical methods for ordinary differentials equations with, in principle, arbitrarily high order of accuracy. These methods, called spectral deferred corrections (SDC), proceed as traditional deferred/defect correction by first using a standard numerical method to compute a provisional solution at a set of substeps within a given time step. Then, an equation for the correction to this provisional solution is constructed and also approximated on the substeps with a simple numerical scheme.

In theory, one could employ any higher-order method for the temporal integration of the auxiliary variable equation. There are two main reasons why an SDC method is utilized here. First, at the end of each time step, the pressure is updated using the predicted pressure q and the potentials ϕ from the projection steps by an approximation to Eq. (28). Therefore the time gradient of ϕ must be computed. SDC methods provide the values of ϕ (with full order of accuracy) at each of the substeps, hence this time derivative can readily be computed.

Secondly, the explicit SDC approach used here can be easily modified to include an implicit or multiple implicit terms in the equation, and such *semi-* or *multi-implicit* variants of SDC have been shown to compare favorably to approaches based on Runge–Kutta or other methods on ODE test problems (see [42,51,48,41] for specific comparisons). We plan on utilizing the method presented here as the basis for a method for the reacting zero-Mach number equations where a multi-implicit SDC approach [10,40] will be used. Such an approach allows for the diffusion and reaction terms to each be handled separately (and if desired with different time steps), while still achieving higher-order accuracy in time.

We briefly describe the SDC procedure here, for details see [24] or the citations above. Consider the Picard integral form of Eq. (62) for a given time step $[t_n, t_{n+1}]$

$$\overline{Q}_{i,j}^{n+1} = \overline{Q}_{i,j}^{n} - \int_{t_n}^{t_{n+1}} \tilde{\nabla} \cdot \widetilde{F}(Q)_{i,j}\,\mathrm{d}t. \tag{80}$$

For notational ease, define $\mathcal{F}(\overline{Q})$ to be the numerical procedure used to approximate the finite-volume flux, i.e.,

$$\mathcal{F}(\overline{Q})_{i,j} \approx -\tilde{\nabla} \cdot \widetilde{F}(Q)_{i,j}. \tag{81}$$

Let $[t_n,t_{n+1}]$ be subdivided using the points $t_0, t_1, t_2, \ldots, t_4$, where

$$t_n = t_0 < t_1 < t_2 \cdots < t_4 \leqslant t_{n+1} \tag{82}$$

and the points t_m form the Gauss–Lobatto quadrature nodes. Although formal fourth-order accuracy for an SDC method applied to a standard ODE can be achieved with fewer nodes, here we use five Lobatto nodes so that a fourth-order approximation of ϕ_t for Eq. (35) can be computed using the intermediate values.

The explicit SDC method proceeds by first computing a provisional solution $\bar{Q}^{0,m}$ at the intermediate points t_m, $m = 0,\ldots,4$ using the standard forward Euler scheme

$$\overline{Q}_{i,j}^{0,m+1} = \overline{Q}_{i,j}^{0,m} + \Delta t_m \mathcal{F}(\overline{Q}^{0,m})_{i,j}. \tag{83}$$

Next, an approximation to the residual function

$$\epsilon(t_m) = \overline{Q}^{0,n} + \int_{t_n}^{t_m} \mathcal{F}(\tau, \overline{Q}^0)\mathrm{d}\tau - \overline{Q}^0(t_m) \tag{84}$$

is computed by calculating the exact integral of the interpolating polynomial for $\mathcal{F}(\tau, \overline{Q}^0(\tau))$ at the intermediate nodes t_m. Using the residual, a Picard-type integral equation for the error

$$\delta(t) = Q(t) - \overline{Q}^0(t) \tag{85}$$

is

$$\delta(t_m) = \int_{t_n}^{t_m} [\mathcal{F}(\overline{Q}^0 + \delta) - \mathcal{F}(\overline{Q}^0)]\mathrm{d}\tau + \epsilon(t_m). \tag{86}$$

The correction equation is then solved using a procedure similar to forward Euler

$$\delta^{m+1} = \delta^m + \Delta t_m[\mathcal{F}(\overline{Q}^{0,m} + \delta^m) - \mathcal{F}(\overline{Q}^{0,m})] + \epsilon(t_{m+1}) - \epsilon(t_m). \tag{87}$$

Note that unlike the classical deferred or defect correction methods in [78,61,60], the equation for $\delta(t)$ is not written here as an ODE. Also, because of the difference of the flux terms in Eq. (86), terms depending only on time, such as the approximation to the pressure q, do not appear in the correction update except through the term $\epsilon(t_m)$.

Finally, the provisional solution is improved by setting

$$\overline{Q}^{1,m} = \overline{Q}^{0,m} + \delta^m. \tag{88}$$

This procedure can then be iterated to produce a series of more accurate approximations. Each iteration of the correction equation raises the formal temporal order of accuracy of the scheme by one, therefore for a fourth-order method, three iterations of the correction equation are needed.

In the method described here, the deferred correction approach is applied only in time. Other approaches have been suggested which apply deferred corrections in both time and space [31,38]. The goal of such an approach is to avoid the use of higher-order spatial discretizations by using compact second-order stencils and achieving higher-order spatial accuracy through deferred corrections. We refer the reader to the above references for details.

3.4. Summary

Here we provide a concise summary of one time step of the numerical method. To compute the flux terms

$$\widetilde{F}_1 = -\tilde{\nabla}\cdot(\widetilde{\rho\mathbf{v}}), \tag{89}$$

$$\widetilde{F}_2 = -\tilde{\nabla}\cdot(\rho\widetilde{\mathbf{v}\circ\mathbf{v}} + I\tilde{q}) \tag{90}$$

used in the finite-volume method update, the following steps are used:

(1a) Compute cell edge averages $\tilde{\mathbf{m}}^*$, $\tilde{\rho}$, and $\tilde{q}$ from cell averages $\bar{\mathbf{m}}^*$, $\bar{\rho}$, and $\bar{q}$ using Eq. (63).
(2a) Compute $\tilde{\mathbf{v}}^*$ using $\tilde{\mathbf{m}}^*$ and $\tilde{\rho}$ and the division formula (67).
(3a) Solve the variable density Poisson Eq. (76) to yield $\bar{\phi}$.
(4a) Compute divergence-free cell edge velocities $\tilde{\mathbf{v}}$ by Eqs. (67) and (78).
(5a) Use formula (68) to compute the nonlinear flux terms in Eqs. (89) and (90) from $\tilde{\mathbf{v}}$, $\tilde{\mathbf{m}}^*$, $\tilde{\rho}$, and $\tilde{q}$.

The time integration scheme proceeds by first computing a prediction for the cell average pressure in the coming substeps. Here the simplest approximation $\bar{q}^m = \bar{p}^n$ is used. Next a provisional solution $\overline{Q}^0$ is computed at each substep using the forward Euler scheme Eq. (83). Then three iterations of the correction equation are performed by

(1b) Computing the residual $\epsilon(t_m)$ using Eq. (84).
(2b) Computing the approximations δ^m using Eq. (87).
(3b) Setting $\overline{Q}^{k+1,m} = \overline{Q}^{k,m} + \delta^m$.

At the end of each full time step, two additional tasks are completed:

(1c) Compute an update to the $\bar{p}^{n+1}$ using intermediate substep values by computing the derivative of the polynomial interpolant $\bar{\phi}^m$ to approximate $\bar{\phi}_t^{n+1}$.
(2c) Reset $\bar{\mathbf{m}}^{*,n+1}$ by subtracting $\nabla\bar{\phi}^{n+1}$ computed using Eq. (71).

3.5. Why conservation and what is new?

In this subsection we point out why conservation of mass, momentum, and energy are central to the present approach, and we point the reader to several earlier publications on related numerical schemes which also use full conservation form.

The auxiliary variable, spectral deferred correction approach to constructing high-order accurate incompressible flow solvers has been developed originally for constant density flows [50,51]. The starting point for these developments are the non-conservative, co-located-variable or staggered-grid projection methods which use velocity components as the primary unknowns, [15,14,17,72,7,62]. Extension of the auxiliary variable technique to zero Mach number, variable density flows proved difficult, largely because of the nonlinearity of the pressure gradient term, $\nabla p/\rho$, in the momentum equation. Here we adopt the conservative formulation for mass, momentum, and energy as proposed in [27,67,68,76], for which the pressure gradient term, ∇p, is linear instead, and an auxiliary variable formulation is possible.

Fully conservative formulations have been adopted previously in the context of constructing Mach-uniform algorithms. Bijl and Wesseling, [8,73], and Park et al. [58] do use a fully conservative discretization for each of the conserved quantities separately, yet they formulate the balances on a staggered scheme of control volumes. Nerinckx and Dick, [55], formulate an all-Mach number solver which relies on conservation of mass, momentum, and energy for the primary control volumes of a given grid. They emphasize possible efficiency gains associated with coupled equations for pressure and temperature corrections in the presence of molecular diffusion or in the presence of strong gravity. The conservative schemes in [8,73,55] may all be considered as akin to projection methods in the sense that they involve merely one or two elliptic pressure (and temperature) solves per time step. Park and Munz [58] test a number of alternative time integration schemes, including second-order three time level Runge–Kutta and backward differentiation formulae (BDF), which differ conceptually from projection schemes. Yet, the semi-implicit versions of their schemes require only linear elliptic solves for a perturbation pressure.

Reisner et al. [64] introduce a semi-implicit (linearly implicit) scheme for meteorological applications that relies on Rosenbrock-type time integration. An approximate multiplicative decomposition of the matrix multiplying the new time level vector of unknowns in the implicit scheme allows the flexibility to choose the set of variables that is to be treated implicitly. If the scheme is adjusted to discretize only the pressure implicitly, it reduces to a projection-type method formulated in terms of density, momentum, and potential temperature (entropy). Knoth [37] develops a version of this scheme which uses the conservative form of the momentum equation in a similar way as done in the present work. See also Lang [39] for related higher-order adaptive discretization strategies for constant density incompressible flow using Rosenbrock-type time discretizations.

4. Numerical tests

In this section, we present three numerical examples to illustrate the accuracy and convergence of our proposed method. The examples chosen here have smoothly varying solutions in space so that the errors due to the temporal integration contribute to the overall error.

4.1. Traveling wave solution to the inviscid Euler equations

Here we test our fourth-order auxiliary variable projection method on a traveling wave solution to the inviscid Euler equations as in e.g. [49,1]. The density field is initialized to $\rho(x,y,0) \equiv 1$, and should stay constant for this problem. The exact solution to the velocity and pressure are

$$\begin{aligned} u(x,y,t) &= 0.75 + 0.25\cos(2\pi(x-0.75t))\sin(2\pi(y-0.75t)),\\ v(x,y,t) &= 0.75 - 0.25\sin(2\pi(x-0.75t))\cos(2\pi(y-0.75t)),\\ p(x,y,t) &= -\frac{1}{64}(\cos(4\pi(x-0.75t)) + \cos(4\pi(y-0.75t))). \end{aligned} \tag{91}$$

The numerical solutions are initialized by setting $t = 0$ in (91) and using Simpson's rule to compute cell averages. The computational domain is $[0,1]\times[0,1]$, and periodic boundary conditions are imposed in both coordinate directions. Fig. 3 is produced by running the code for 32×32, 64×64, and 128×128 grids to a final time of $T = 0.5$ using $\Delta t = \Delta x$. Fig. 3 displays the L_∞ norm of the errors in momentum and ∇p and clearly indicates the fourth-order accuracy of the method.

To further demonstrate that the method is fourth-order in time and space, the same example is run again using both a third and fourth-order method. The solution is computed to $T = 0.1$ on $N\times N$ grids for $N = 64$, 96, 128, 192 and 256. The third-order method is constructed by using the exact same spatial resolution but performing one fewer iteration of the correction step in the SDC algorithm. The data in Fig. 4 demonstrates the different temporal orders of accuracy for the momentum field.

4.2. Traveling vortex problem

Here we test our method on a problem whose exact solution is known; however, in this instance the density is no longer constant in space and time. The problem consists of a smooth traveling vortex with non-constant density. This is in essence a smooth version of Gresho's vortex test [29,44]. The problem consists of a rotating vortex initially placed at $(0.5, 0.5)$ and moving with a prescribed constant speed. The density is initialized as a smooth radially symmetric bump at $(0.5, 0.5)$ and also moves with the vortex.

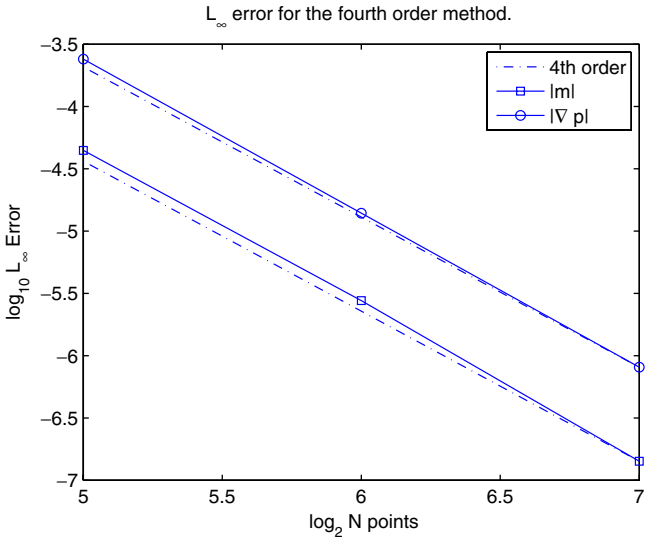

Fig. 3. L_∞ errors in $|\mathbf{m}|$ and $|\nabla p|$ versus number of grid points for the problem 4.1. The dotted and dashed lines superimposed on the graph correspond to fourth-order convergence. $T = 0.5$.

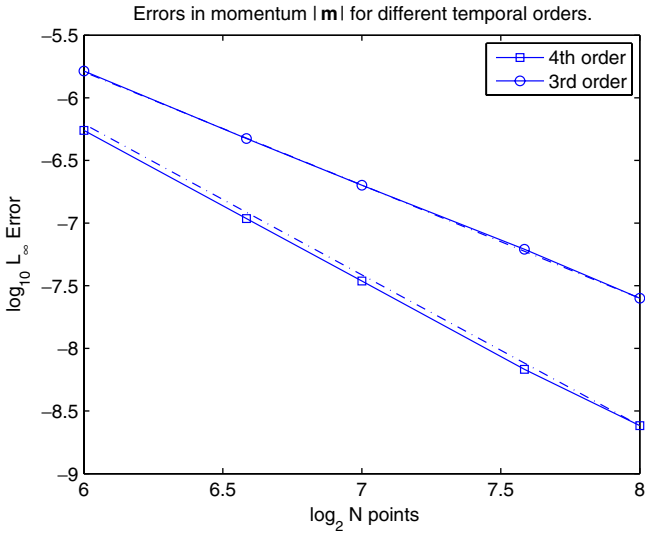

Fig. 4. L_∞ errors in $|\mathbf{m}|$ for different temporal orders versus number of grid points for the problem 4.1. The dotted and dashed lines superimposed on the graph correspond to third and fourth-order convergence. $T = 0.1$.

Specifically,

$$\rho(x,y,t) = \begin{cases} \rho_c + \frac{1}{2}(1-r^2)^6 & \text{if } r < 1, \\ \rho_c & \text{otherwise,} \end{cases} \tag{92}$$

$$u(x,y,t) = \begin{cases} -1024\sin(\theta)(1-r)^6 r^6 + u_c & \text{if } r < 1, \\ u_c & \text{otherwise,} \end{cases} \tag{93}$$

$$v(x,y,t) = \begin{cases} 1024\sin(\theta)(1-r)^6 r^6 + v_c & \text{if} r < 1, \\ v_c & \text{otherwise.} \end{cases} \tag{94}$$

Here u_c and v_c are the prescribed translation speeds of the vortex which will vary for different test cases, so that $x_c(t) = 0.5 + t\ u_c$ and $y_c(t) = 0.5 + t\ v_c$ give the center of the vortex at time t. The radial distance is scaled by the radius of the vortex $R = 0.4$, or

$$r = \sqrt{(x - x_c(t))^2 + (y - y_c(t))^2}/R.$$

Likewise, θ is defined as $\theta = \arctan(\frac{y - y_c(t)}{x - x_c(t)})$. The constant ρ_c is chosen to be 1/2 for all computations.

The pressure field can be derived by considering the centripetal force and is given by

$$p(x,y,t) = \begin{cases} p(r) - p(1) & \text{if} r < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{95}$$

where

$$\begin{aligned} p(r) = 1024^2 \Bigg(& \frac{1}{72}r^{36} - \frac{6}{35}r^{35} + \frac{15}{17}r^{34} - \frac{74}{33}r^{33} + \frac{57}{32}r^{32} + \frac{174}{31}r^{31} - \frac{259}{15}r^{30} + \frac{450}{29}r^{29} + \frac{153}{8}r^{28} \\ & - \frac{1564}{27}r^{27} + \frac{510}{13}r^{26} + \frac{204}{5}r^{25} - \frac{1473}{16}r^{24} + \frac{1014}{23}r^{23} + \frac{1053}{22}r^{22} - \frac{558}{7}r^{21} + \frac{783}{20}r^{20} + \frac{54}{19}r^{19} \\ & - \frac{38}{9}r^{18} - \frac{222}{17}r^{17} + \frac{609}{32}r^{16} - \frac{184}{15}r^{15} + \frac{9}{2}r^{14} - \frac{12}{13}r^{13} + \frac{1}{12}r^{12} \Bigg). \end{aligned} \tag{96}$$

We initialize the conservative variables using Eqs. (92)–(95) and Simpson's rule. Contour plots of the initial conditions are given in Fig. 5 for $u_c = 1$ and $v_c = 1$. For all tests below, $\Delta t = \Delta x$.

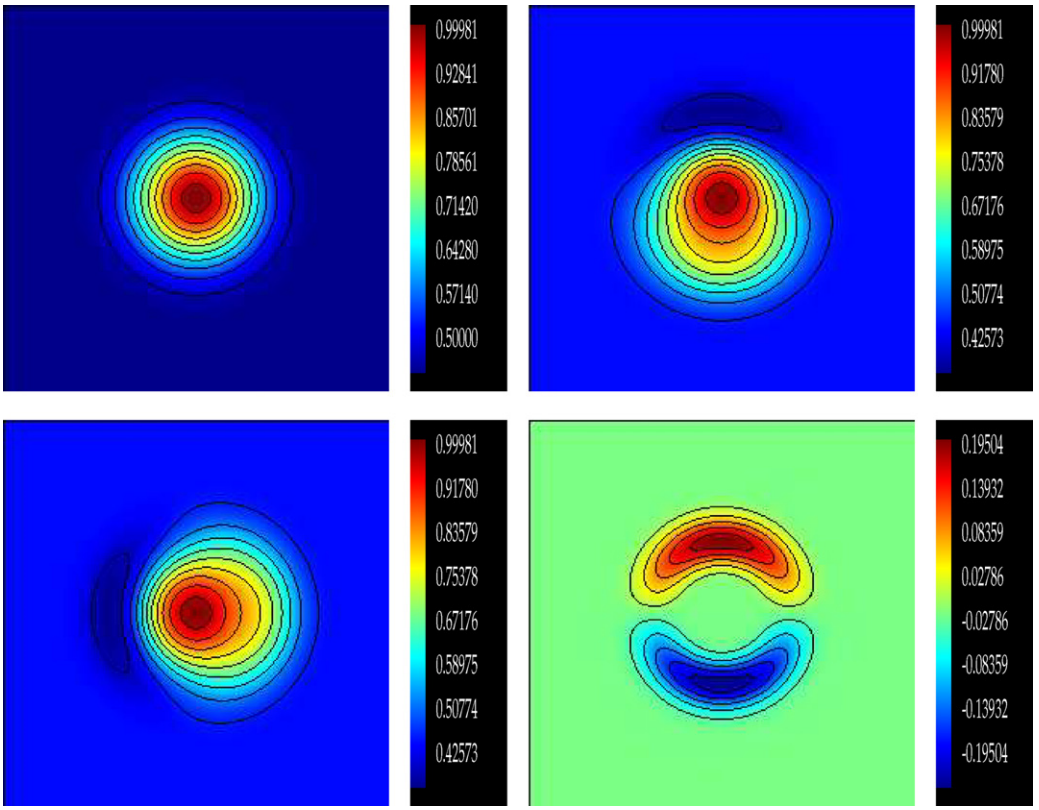

Fig. 5. Initial conditions for the vortex Problem (4.2.1) with $u_c = 1$, $v_c = 1$ on a 192×192 mesh. The upper left figure represents the *density*, the upper right figure represents the *x-momentum*, the lower left figure represents the *y-momentum*, and the lower right figure represents p_y.

4.2.1. Vortex moving diagonally in a doubly periodic domain

For the first test, we run the code in the unit square with periodic boundary conditions in both coordinate directions. The advection velocities are set to $u_c = v_c = 1$ so that the vortex travels in a diagonal direction. We perform the computation until the flow completes one period which corresponds to $T = 1$, hence the exact solution is identical to the initial conditions. Fig. 5 shows the initial conditions produced using a 192×192 grid, and Fig. 6 shows the computed final solution after one period. It is evident from the Fig. 6 that the flow features are captured accurately. In Fig. 7 and Table 1, we demonstrate the accuracy and the convergence rates of our method for different variables. Here $N \times N$ grids for $N = 64, 96, 128, 192$ are used when producing Fig. 7 and Table 1. Fig. 7 and Table 1 clearly indicate the fourth-order of accuracy of our method.

4.2.2. Vortex moving horizontally in a doubly periodic domain

For the next example, the vortex is given by velocities of $u_c = 1$ and $v_c = 0$, hence the vortex moves in x-direction only. The test is included to compare with the following case in which solid wall boundary conditions are imposed. Fig. 8 shows the computed solutions at $T = 1.0$. Again, it is evident from Fig. 8 that our method captures the flow features accurately. Fig. 10 illustrates error in several principle variables at $T = 1.0$ and with a 128×128 mesh. More errors using $N \times N$ grids for $N = 64$, 96, 128 and 192 are calculated to perform an accuracy analysis. Fig. 9 and Table 2, produced from these errors, clearly demonstrate the fourth-order accuracy of our method.

4.2.3. Vortex problem with solid wall boundary conditions

Here we modify the problem in 4.2.2 in that solid wall boundary conditions are enforced in the y-coordinate direction with periodic boundary conditions in the x-direction. The purpose here is to demonstrate that the

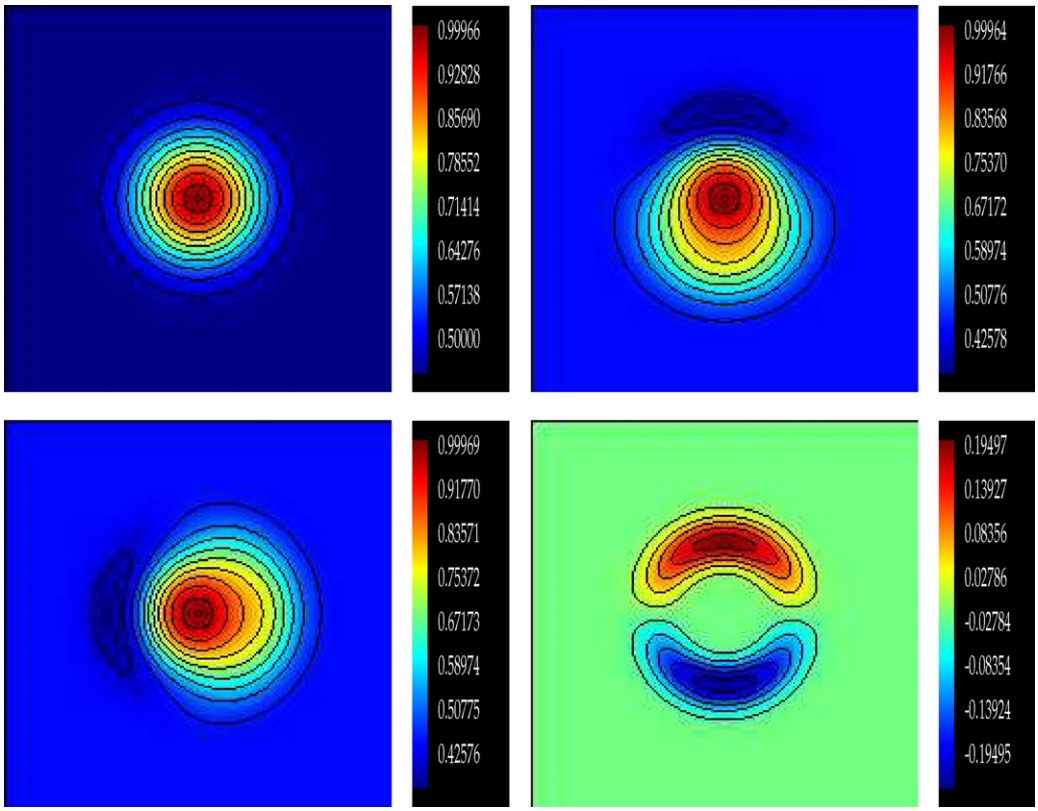

Fig. 6. Computed solution to the vortex problem (4.2.1) with $u_c = 1$ and $v_c = 1$ computed at $T = 1.0$ with 192×192 mesh. The upper left figure represents the *density*, the upper right figure represents the *x-momentum*, the lower left figure represents the *y-momentum*, and the lower right figure represents p_y.

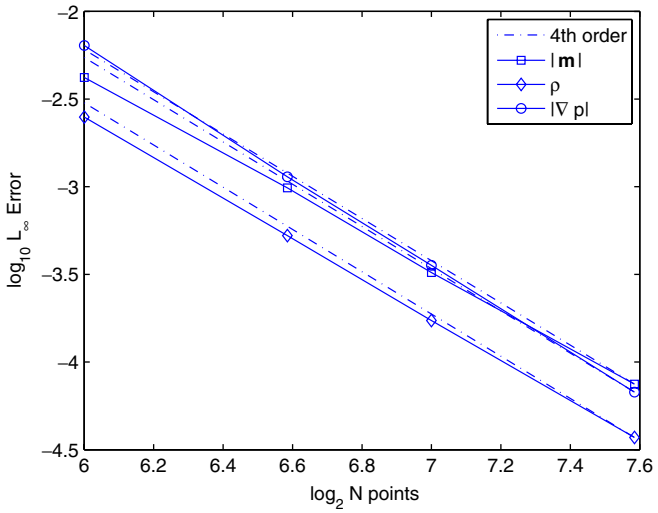

Fig. 7. L_∞ errors versus number of grid points for the Problem 4.2.1 with $u_c = 1$ and $v_c = 1$ at $T = 1.0$.

imposition of solid wall boundary conditions does not affect the accuracy of the numerical simulations compared to those in a doubly periodic domain. The problem is again run until $T = 1.0$, and the errors are plotted in Fig. 11. Comparing these plots to those in Fig. 10, clearly shows that the imposition of solid wall boundaries

Table 1
Convergence rates for the problem 4.2.1 computed by comparing the error when using grid size h with that when using grid size $2h$

	Mesh refinement $2h = 1/64$	Mesh refinement $2h = 1/96$
Momentum	3.70	3.72
Density	3.85	3.82
Grad p	4.16	4.07

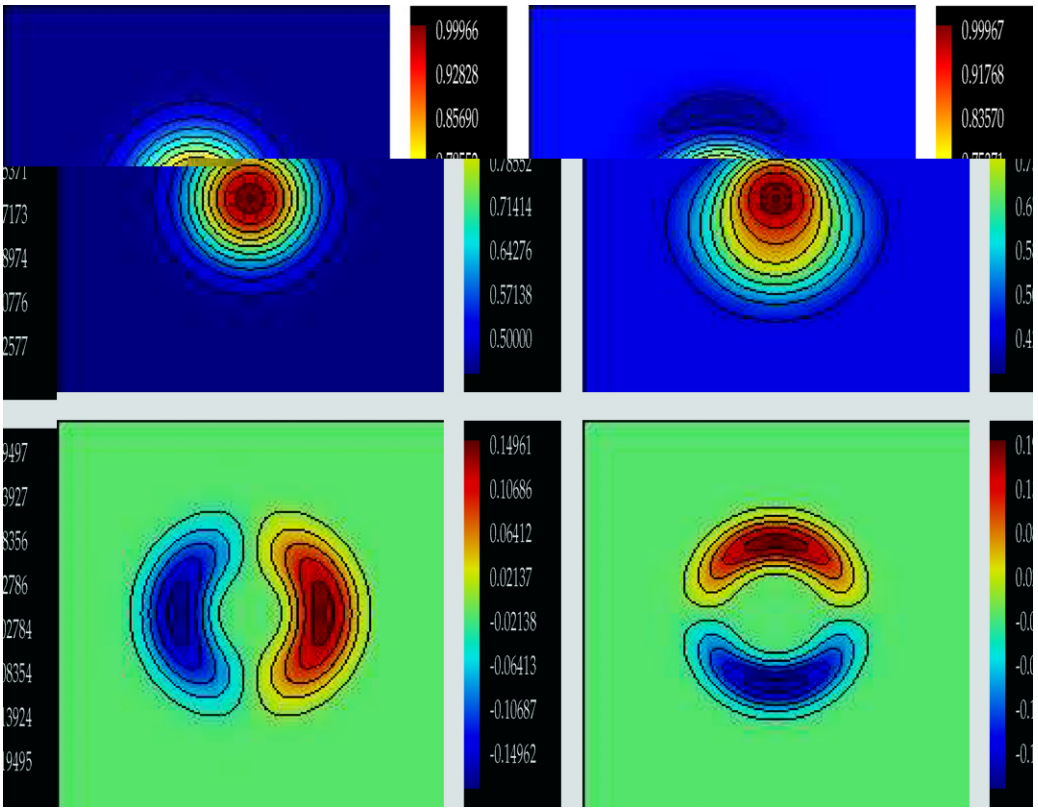

Fig. 8. Computed solution at $T = 1.0$ for the problem 4.2.2 with a 192×192 mesh. Here $u_c = 1$ and $v_c = 0$, thus the vortex moves in the horizontal direction. The upper left figure represents the *density*, the upper right figure represents the *x-momentum*, the lower left figure represents the *y-momentum*, and the lower right figure represents p_y.

does not affect the accuracy of the method for this problem. Fig. 12 shows the convergence rates of our method for this problem and again indicates fourth-order convergence.

4.2.4. Long-time coarse grid evolution

In the spirit of the vortex test case in [28,44], we present results of a longer-time run with poor spatial resolution. Specifically, the example in Section 4.2.3 is repeated using a coarse 32×32 grid to compute the solution after three revolutions (i.e. $T = 3.0$). The results are shown Fig. 13. Even at this coarse resolution, the pertinent features of the vortex are well preserved without the use of limiters of any sort.

4.3. Gravity driven instability

For the final problem we present an example with more complicated dynamics. The initial conditions consist of a shear layer in a stably stratified density field in a domain with periodic boundary conditions in the x-direction and solid wall boundary conditions in the y-directions. A gravity term is included in the momentum

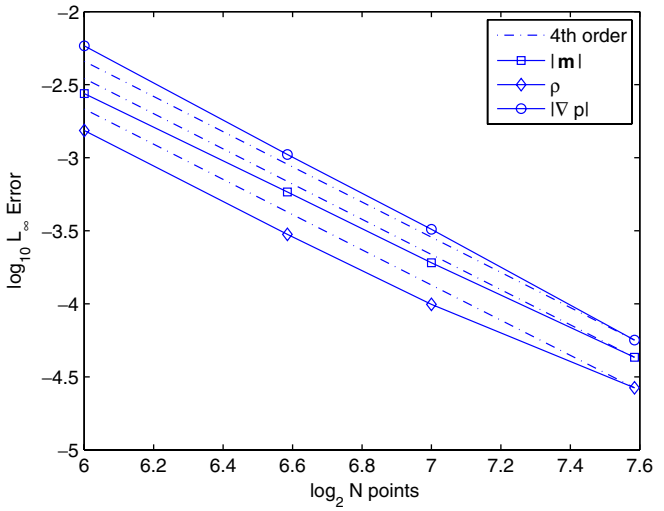

Fig. 9. L_∞ errors versus number of grid points for the Problem 4.2.2 at $T = 1.0$.

Table 2
Convergence rates for the Problem 4.2.2 computed by comparing the error when using grid size h with that when using grid size $2h$

	Mesh refinement $2h = 1/64$	Mesh refinement $2h = 1/96$
Momentum	3.85	3.76
Density	3.95	3.49
Grad p	4.17	4.22

equation, and two perturbations to the density field driven by gravity create an instability in the shear layer which then rolls up into a main vortex and a smaller vortex.

Specifically, the initial velocity is given by the shear layer

$$u(x,y,0) = \frac{1}{2}\tanh\left(\frac{y-0.5}{\epsilon}\right), \tag{97}$$

$$v(x,y,0) = 0, \tag{98}$$

with $\epsilon = 1/50$. The density is given by

$$\rho(x,y,0) = 0.5 - c(y-0.5)^3 + \sum_{N=-10}^{10}\{c_1 e^{-r_1^2} - c_2 e^{-r_2^2}\}, \tag{99}$$

where

$$r_i^2 = \left(\frac{x - x_i - N}{\sigma}\right)^2 + \left(\frac{y - y_i}{\sigma}\right)^2. \tag{100}$$

In these initial conditions, the background density is a stable stratification defined by a cubic polynomial in y. The strength of the stratification is determined by the constant c. The density perturbations are Gaussians centered at $(x_1, y_1) = (0.5, 0.75)$ and $(x_2, y_2) = (0.5, 0.25)$ with strength c_1 and c_2 and width determined by σ.

Note that we make the density periodic in the x-direction in Eq. (99) by summing the periodic images of the density profile originally defined in $[0,1]\times[0,1]$. Since the perturbations decay rapidly, including ten images in each direction is more than sufficient to achieve machine precision.

4.3.1. Convergence test

To test the convergence of the method for this problem, we compute a series of approximations using smooth initial data determined by $\epsilon = 1/10$ in (98) and $\sigma = 0.1$, $c = 0.5$, $c_1 = c_2 = 0.05$ in (99). The final run

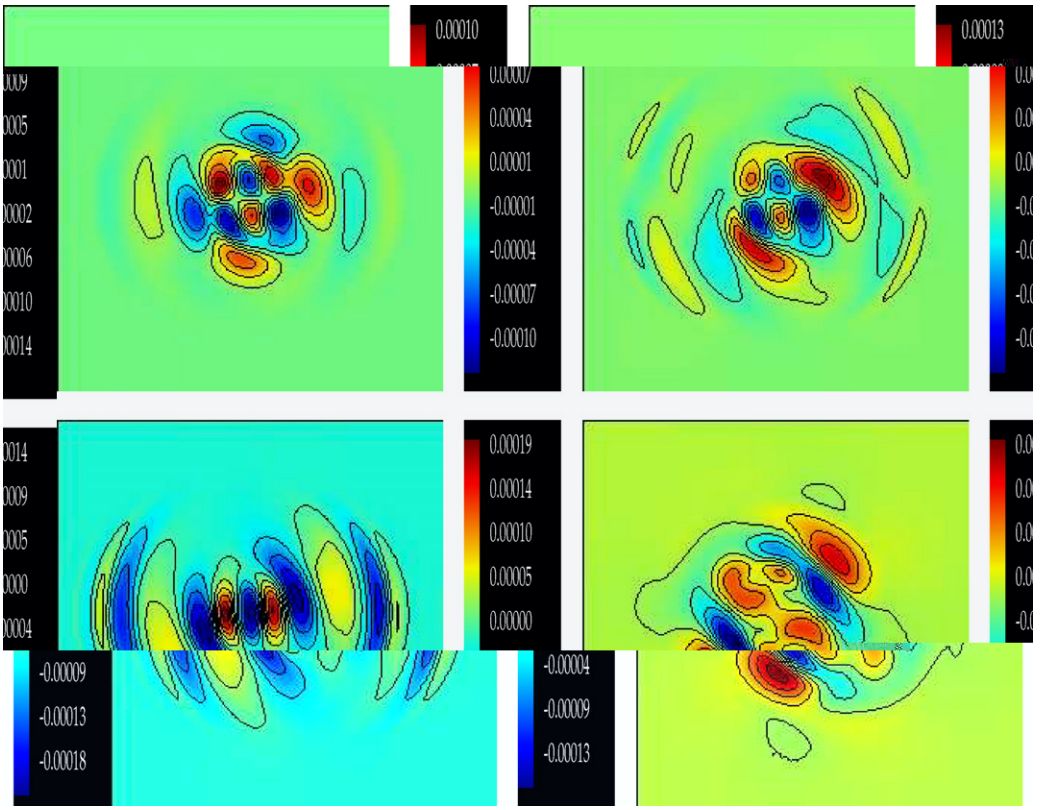

Fig. 10. Error plots in the principle variables for the problem 4.2.2. Periodic boundary conditions are enforced in both x- and y-directions. These plots are produced at $T = 1.0$ with a 128×128 mesh. Here the upper left figure represents the error in *density*, the upper right figure represents the error in *x-momentum*, the lower left figure represents the error in *y-momentum*, and the lower right figure represents the error in p_y.

time for the convergence analysis is set to $T = 0.25$. Since no analytical solution is known, convergence rates are computed by computing the difference between two successive runs in the L_∞ norm. Convergence rates for density, momentum and the pressure gradient using grids of size 32×32 to 256×256 and $\Delta t = \Delta x$ are reported in Tables 3–5. The data indicates that the method is converging with fourth-order accuracy.

4.3.2. Comparison with second-order code

As a final comparison, we compare results computed using the fourth-order method presented here with those from a second-order numerical method described in [68] (with a modified stencil for the projection steps following the presentation in [76]). Here we use initial conditions which are poorly resolved on coarse grids, specifically $\epsilon = 1/50$, $\sigma = 0.25$ and overall magnitude by the constants $c = 0.0156$, and $c_1 = c_2 = 0.00156$. These initial conditions were chosen specifically so that coarse grid runs would have difficulty accurately predicting the time evolution of the vortex formation. Fig. 14 shows the initial conditions and the density and vorticity solutions at $T = 2.5$ computed with the fourth-order code on a 128×128 grid.

Assuming both the fourth-order code and the second-order code are converging to the same solution (which the numerical evidence supports), for a sufficiently fine resolution, the fourth-order code will be more accurate. Mathematically this is saying nothing more than

$$C_4 \Delta t^4 < C_2 \Delta t^2 \tag{101}$$

for sufficiently small Δt, for any choice of the constants C_4 and C_2. The point of this example is to demonstrate that even for relatively coarse grids, the use of the higher-order method can be beneficial. Fig. 15 compares the computed solutions for the two methods on a 32×32 grid. Comparing these figures with Fig. 14, we see that

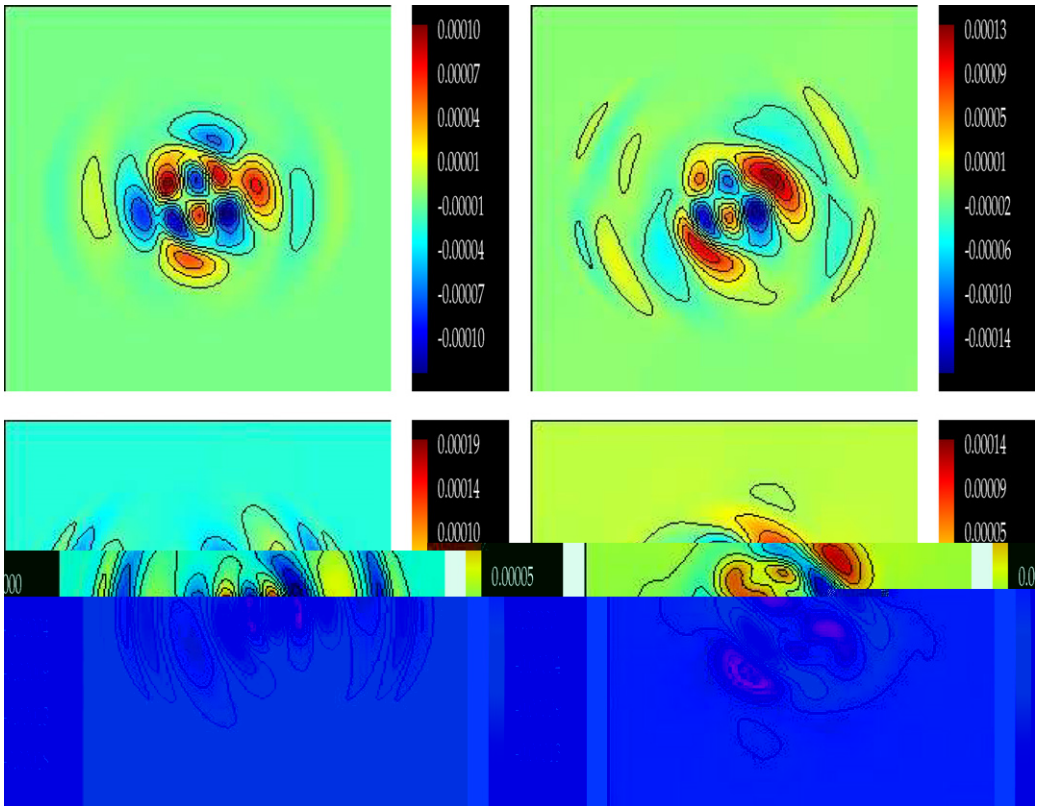

Fig. 11. Error plots for the problem 4.2.3. Periodic boundary conditions are enforced in x-direction and solid wall boundary conditions are enforced in y-direction. Plots are produced at $T = 1.0$ with 128×128 mesh. Here the upper left figure represents the error in *density*, the upper right figure represents the error in *x-momentum*, the lower left figure represents the error in *y-momentum*, and the lower right figure represents the error in p_y.

the second-order method does not predict the instability in the shear layer. The fourth-order method does slightly better although the emergence of the secondary vortex is missed. Fig. 16 compares the computed solutions for the two methods on a 64×64 grid. At this resolution, both methods are able to predict both

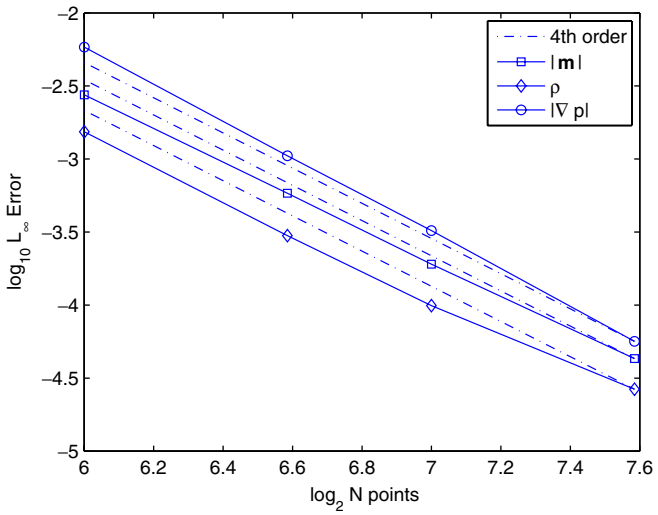

Fig. 12. L_∞ errors versus number of grid points for the Problem 4.2.3. $T = 1.0$.

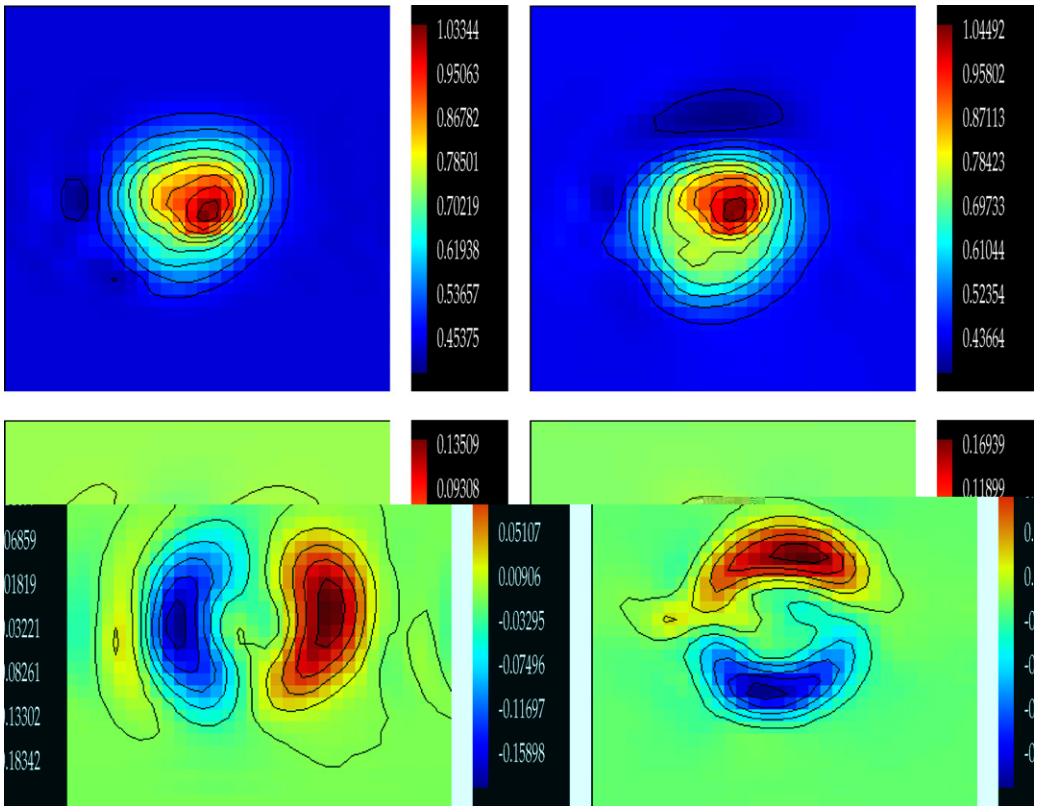

Fig. 13. Computed solution of the problem 4.2.4 at $T = 3.0$ with using a coarse 32×32 mesh. Here the upper left figure represents the *density*, the upper right figure represents the *x-momentum*, the lower left figure represents the *y-momentum*, and the lower right figure represents p_y.

Table 3
Errors and approximate convergence rates for density for the gravity driven instability problem (4.3.1) computed by comparing the difference between a solution computed using grid size h with that using grid size $2h$

Mesh refinement	$\|\rho_h - \rho_{2h}\|_{L_\infty}$	Between mesh	Convergence rates
$2h = 1/32$	4.83×10^{-4}		
$2h = 1/64$	3.76×10^{-5}	1/32–1/64	3.68
$2h = 1/128$	2.45×10^{-6}	1/64–1/128	3.93

Table 4
Errors and approximate convergence rates for the momentum for the gravity driven instability problem (4.3.1) computed by comparing the difference between a solution computed using grid size h with that using grid size $2h$

Mesh refinement	$\|\mathbf{m}_h - \mathbf{m}_{2h}\|_{L_\infty}$	Between mesh	Convergence rates
$2h = 1/32$	1.1×10^{-3}		
$2h = 1/64$	7.49×10^{-5}	1/32–1/64	3.87
$2h = 1/128$	4.96×10^{-6}	1/64–1/128	3.91

the primary and secondary vortices. Numerical diffusion is evident in the second-order results in that the roll-up of the shear layer is less than the resolved solution in Fig. 14. For the fourth-order method, the under-resolution causes some slight oscillations in the vorticity due to the lack of any form of slope limiting in the finite-volume treatment of the nonlinear terms.

Table 5
Errors and approximate convergence rates for the pressure gradient for the gravity driven instability problem (4.3.1)

Mesh refinement	$\|\nabla p_h - \nabla p_{2h}\|_{L_\infty}$	Between mesh	Convergence rates
$2h = 1/32$	2.5×10^{-3}		
$2h = 1/64$	1.69×10^{-4}	1/321/64	3.88
$2h = 1/128$	1.03×10^{-5}	1/64–1/128	4.03

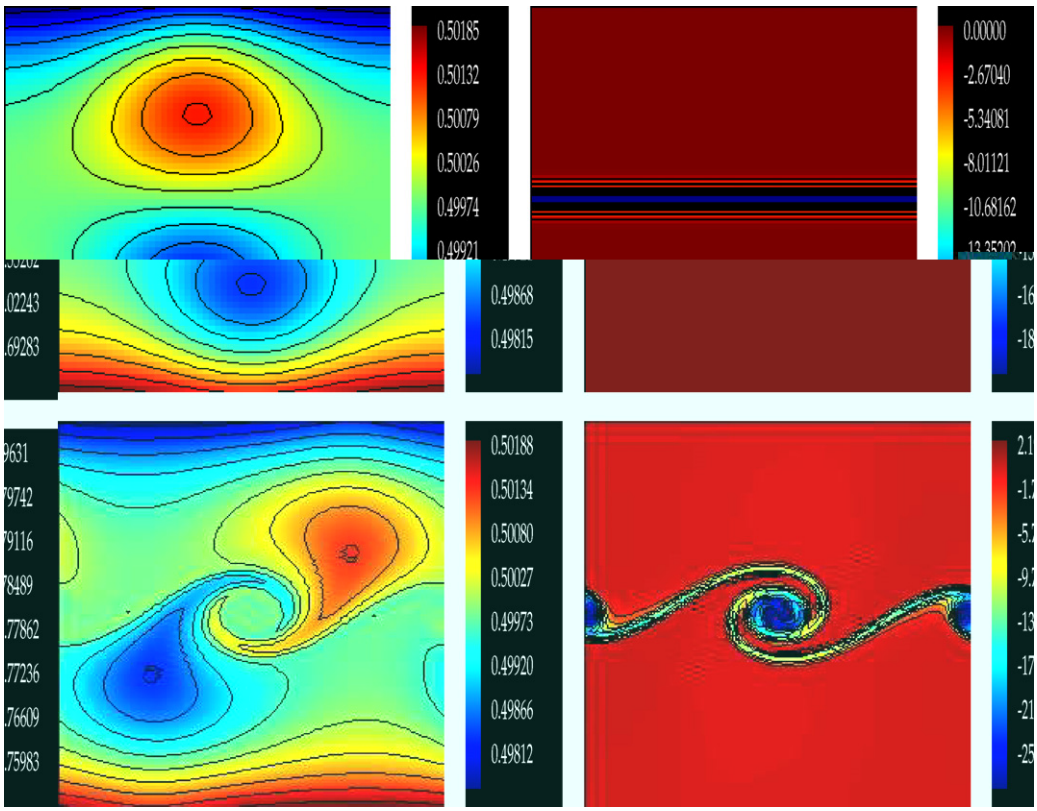

Fig. 14. Initial conditions and solution at $T = 2.5$ computed on a 128×128 mesh for the gravity driven instability test (Problem 4.3.2). Here the upper left and right figures represent the initial density and vorticity respectively. The lower left and right figures are the density and vorticity at $T = 2.5$.

A comparison between two methods should of course also take into account the computational cost per time step of the methods. Ideally, one would like to compare the computational cost for a given level of accuracy, however, performing such a comparison is not straightforward. We present no timing results for the test case above, since a comparison using small grids and non-optimized codes are unlikely to give a fair comparison in general. We can however make some general observations regarding accuracy.

- Given a sufficiently stringent error tolerance and assuming that the computational cost per time step of each method scales linearly with the number of grid points, the fourth-order method will be more efficient than the second-order method. This again is a simple consequence of Eq. (101). Note that the cost per time step may not grow linearly if the method is run in parallel and the number of processors is also increasing.
- For both methods, the most computationally expensive part of each time step is the solution of the Poisson equation associated with the projection. For the fourth-order method, sixteen projections are required for each time step, while only two each time step are needed in the second-order method.

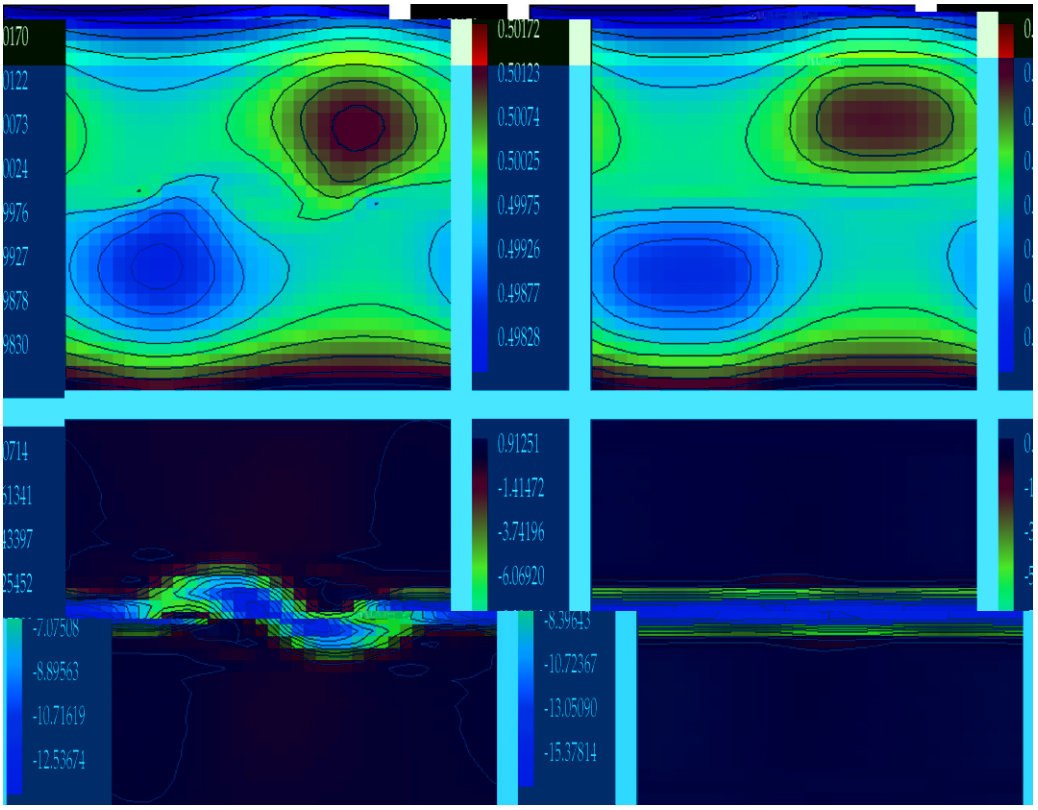

Fig. 15. Comparison of the density and vorticity solutions for the Problem 4.3.2 using a 32×32 grid. The two figures on the right are computed with a second-order method. Those on the left are computed with the fourth-order method described in this paper.

- The maximum stable time-step allowed for the fourth-order method is substantially larger than for the second-order method. In the above test, the fourth-order method was run with $\Delta t = 3\Delta x$, while the second-order method $\Delta t \approx 0.21\Delta x$.
- For both methods the Poisson equations are solved using multi-grid, but the stencil in the fourth-order case is larger than in the second-order method. The increase in stencil size does not seem to have a large impact on the number of V-cycle iterations required for convergence, however, the cost of each multi-grid operation increases with the size of the stencil. Also, as discussed in Section 3.2.5, the initial guesses for the multi-grid solves become better as the deferred correction iterations proceed.
- The fourth-order method has a much higher storage cost due to the necessity of storing the solution values at substeps. Currently, efficiency on single processor computers often depends sensitively on memory access. Any large three-dimensional simulation will likely be done on a parallel computer, where costs associated with increased storage are machine and implementation dependent.
- For three-dimensional simulations, a reduction of the grid spacing by one half in each dimension (including time) leads to a factor of sixteen increase in computational cost, but only a factor of eight in storage. This suggests that if the computational speed and storage of computers continue to increase at roughly the same rate, the increased storage overhead for SDC methods will become less important in the future.

5. Discussion

We have presented a novel numerical method for zero-Mach number limit of the compressible gas dynamics equations which combines a conservative fourth-order finite-volume discretization with a new auxiliary variable formulation of the equations in conservation form. Temporal accuracy is achieved using spectral deferred corrections.

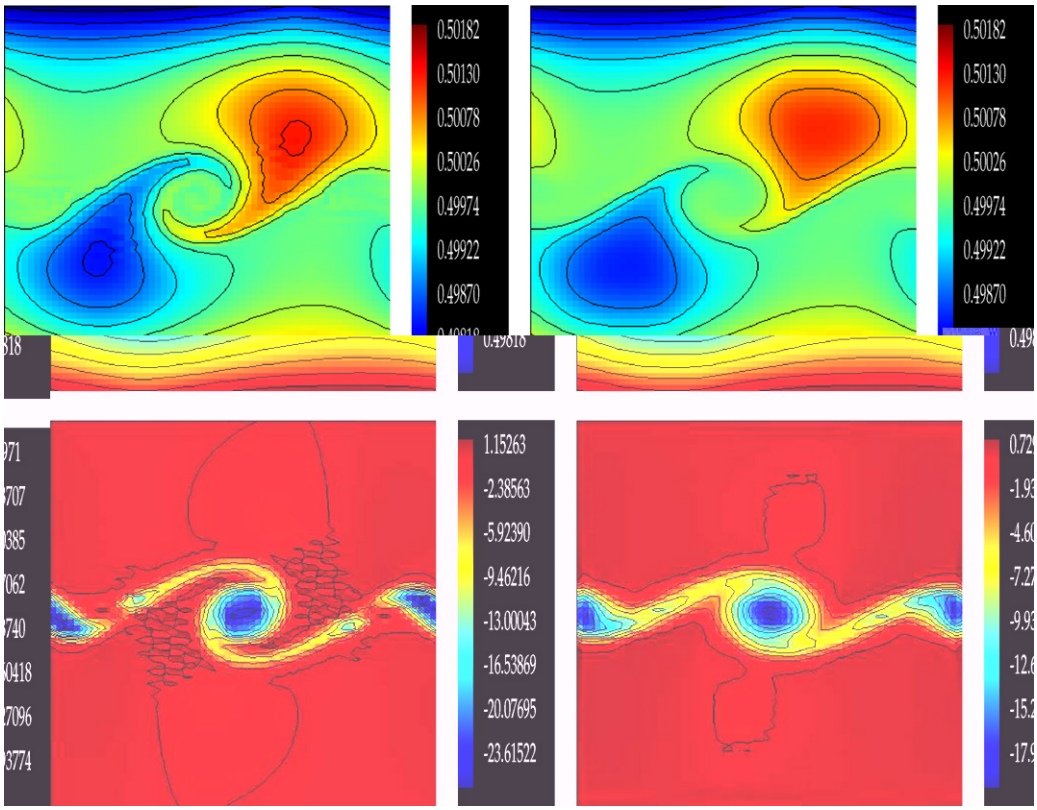

Fig. 16. Comparison of the density and vorticity solutions for the problem 4.3.2 using a 64×64 grid. The two figures on the right are computed with a second-order method. Those on the left are computed with the fourth-order method described in this paper.

As mentioned in the introduction, this is the first step in creating a higher-order numerical method for the reacting zero-Mach number equations. We are currently working on adding species equations and reaction and diffusion terms to the numerical method pursuing the multi-implicit temporal approach from [10,40].

Another direction that we are pursuing involves extending the current ideas to the physically important case where the Mach number is small, but not zero. In this case, there is no specific divergence constraint, rather the energy equation and the equation of state return. Following the strategy presented in this paper, one could posit an approximation to the pressure $q(x,t)$ and solve the resulting equations without regard to the equation of state. However, the natural analog of the auxiliary momentum variable in this approach will not differ from the true momentum by a pure gradient, and therefore the pressure $p(t)$ cannot be recovered given an approximate value $q(t)$ and the solution of the auxiliary equation. However, we are pursuing a strategy in which a sequence of auxiliary variable equations are formed which provide an iterative improvement of the pressure. These iterations can then be integrated into the deferred corrections iterations to achieve an efficient method.

Acknowledgements

M.M. acknowledges the support of the *Alexander-von-Humboldt Stiftung* through a research stipend, and funding of part of the present work by the US Department of Energy. S.K. was funded by the US Department of Energy through the Scientific Discovery Through Advanced Computing program. R.K. appreciates partial funding of the present work by Deutsche Forschungsgemeinschaft, grants KL 611/6, KL 611/14. The authors thank Matthias Münch for helping out with the second-order calculations in Section 4.3.2.

References

[1] 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. Comput. Phys. 142 (1998) 1–46.
[2] A.S. Almgren, J.B. Bell, C.A. Rendleman, M. Zingale, Low mach number modeling of type Ia supernovae. I. Hydrodynamics, Astrophys. J. 637 (2006) 922–936.
[3] A.S. Almgren, J.B. Bell, C.A. Rendleman, M. Zingale, Low mach number modeling of type Ia supernovae. II. Energy evolution, Astrophys. J. 649 (2006) 927–938.
[4] J.B. Bell, P. Colella, H.M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.
[5] J.B. Bell, M.S. Day, A.S. Almgren, M.J. Lijewski, C.A. Rendleman, Adaptive numerical simulation of turbulent premixed combustion, in: Proceedings of the First MIT Conference on Computational Fluid and Solid Mechanics, June 2001.
[6] J.B. Bell, M.S. Day, I.G. Shepherd, M. Johnson, R.K. Cheng, J.F. Grcar, V.E. Beckner, M.J. Lijewski, Numerical simulation of a laboratory-scale turbulent V-flame, Proc. Natl. Acad. Sci. USA 102 (2005) 10006–10011.
[7] J.B. Bell, D.L. Marcus, A second order projection method for variable density flows, J. Comput. Phys. 101 (1) (1992) 1–24.
[8] H. Bijl, P. Wesseling, A unified method for computing incompressible and compressible flows in boundary-fitted coordinates, J. Comput. Phys. 141 (1998) 153–173.
[9] N. Botta, R. Klein, A.S. Almgren, Asymptotic analysis of a dry atmosphere, in: P. Neittaanmäki, T. Tiihonen, P. Tarvainen (Eds.), ENUMATH 99 – Proceedings of the 3rd European Conference on Numerical Mathematics, Jyväskylä, Finland, 1999, World Scientific, Singapore, 2000.
[10] A. Bourlioux, A.T. Layton, M.L. Minion, Higher-order multi-implicit spectral deferred correction methods for problems of reacting flow, J. Comput. Phys. 189 (2003) 351–376.
[11] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[12] T. Buttke, Velicity methods: Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow, NATO ASI Series C, vol. 395, Kluwer Academic Publisher, 1993, pp. 39–57.
[13] T. Buttke, A.J. Chorin, Turbulence calculations in magnetization variables, Appl. Numer. Math. 12 (1993) 47–54.
[14] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[15] A.J. Chorin, Numerical solution of the Navier–Stokes equations for an incompressible fluid, Bull. Am. Math. Soc. 73 (1967) 928–931.
[16] A.J. Chorin, Numerical solution of incompressible flow problems, Stud. Numer. Anal. 2 (1968) 64–71.
[17] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 742–762.
[18] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.
[19] P. Colella, P.R. Woodward, The piecewise parabolic method (ppm) for gas-dynamics simulations, J. Comput. Phys. 54 (1984) 174–201.
[20] Ricardo Cortez, Impulse-based Particle Methods for Fluid Flow, Ph.D. Thesis, University of California, Berkeley, May 1995.
[21] Ricardo Cortez, An impulse-based approximation of fluid motion due to boundary forces, J. Comput. Phys. 123 (1996) 341–353.
[22] Ricardo Cortez, On the accuracy of impulse methods for fluid flow, SIAM J. Sci. Comput. 19 (4) (1998) 1290–1302.
[23] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer Verlag, New York, Berlin, Heidelberg, 1999.
[24] Alok Dutt, Leslie Greengard, Vladimir Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[25] E. Wienan, Jian-Guo Liu, Finite difference schemes for incompressible flows in the velocity-impulse density formulation, J. Comput. Phys. 130 (1997) 67–76.
[26] W. E, Jian-Guo Liu, Gauge method for viscous incompressible flows, Commun. Math. Sci. 1 (2003) 317–332.
[27] K.J. Geratz, Erweiterung eines Godunov Typ Verfahrens für zweidimensionale kompressible Strömungen auf die Fälle kleiner und verschwindender Machzahl, Ph.D. Thesis, Rheinisch Westfälische Technische Hochschule, Aachen, 1997.
[28] P.M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly-consistent mass matrix, Int. J. Numer. Methods Fluids 11 (1990) 587.
[29] P.M. Gresho, S.T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly-consistent mass matrix: part 2: Implementation, Int. J. Numer. Methods Fluids 11 (1990) 621–659.
[30] J.L. Guermond, P. Minev, Jie Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006).
[31] Bertil Gustafsson, Wendy Kress, Deferred correction methods for initial value problems, BIT 41 (2001) 986–995.
[32] K.C. Karki, S.V. Patankar, Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations, AIAA J. 27 (1989) 1167–1174.
[33] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[34] R. Klein, Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: One-dimensional flow, J. Comput. Phys. 121 (1995) 213–237.
[35] R. Klein, Asymptotic analyses for atmospheric flows and the construction of asymptotically adaptive numerical methods, ZAMM 80 (2000) 765–777.

[36] O.M. Knio, H.N. Najm, P.S. Wyckoff, A semi-implicit numerical scheme for reacting flow. ii stiff, operator-split formulation, JCP 154 (1999) 428–467.
[37] O. Knoth, Compressible atmospheric modelling at all scales, in: Mathematical Theory and Modelling in Atmosphere-Ocean Science, 2006.
[38] Wendy Kress, Bertil Gustafsson, Deferred correction methods for initial boundary value problems, J. Sci. Comput. 17 (1–4) (2002) 241–251.
[39] J. Lang, Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems. Theory, Algorithm, and Applications, Springer-Verlag, 2000.
[40] Anita T. Layton, Michael L. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics, J. Comput. Phys. 194 (2) (2004) 697–714.
[41] Anita T. Layton, Michael L. Minion, Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations, BIT 45 (2005) 341–373.
[42] Anita T. Layton, Michael L. Minion, Implications of the choice of predictors for semi-implicit Picard integral deferred corrections methods, Commun. Appl. Math. Comput. Sci. 2 (1) (2007) 1–34.
[43] F. Lipps, R. Hemler, A scale analysis of deep moist convection and some related numerical calculations, JAS 29 (1982) 2192–2210.
[44] R. Liska, B. Wendroff, Comparison of several finite difference schemes on 1D and 2D test problems for the Euler equations, SIAM J. Sci. Comput. 25 (3) (2003) 995–1017.
[45] M. Liu, Y. Ren, H. Zhang, A class of fully second order accurate projection methods for solving the incompressible Navier–Stokes equations, J. Comput. Phys. 200 (1) (2004) 325–346.
[46] J.H. Maddocks, R.L. Pego, An unconstrained hamiltonian formulation for incompressible fluid flow, Commun. Math. Phys. 170 (1995) 207–217.
[47] A. Majda, J. Sethian, The derivation and numerical solution of the equations for zero Mach number combustion, Combust. Sci. Technol. 42 (1985) 185–205.
[48] M.L. Minion, Semi-implicit projection methods for incompressible flow based on spectral deferred corrections, Appl. Numer. Math. 48 (3–4) (2004) 369–387.
[49] Michael L. Minion, A note on the stability of Godunov-projection methods, J. Comput. Phys. 123 (1996).
[50] Michael L. Minion, Higher-order semi-implicit projection methods, in: M. Hafez, (Ed.), Numerical Simulations of Incompressible Flows. Papers from the workshop held in Half Moon Bay, CA, June 19–21, 2001,World Scientific Publishing, River Edge, NJ, January 2003, pp. 126–140.
[51] Michael L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Commun. Math. Sci. 1 (2003) 471–500.
[52] C.D. Munz, S. Roller, R. Klein, K.J. Geratz, The extension of incompressible flow solvers to the weakly compressible regime, CandF 32 (2003) 173–196.
[53] H.N. Najm, A conservative low Mach number projection method for reacting flow modeling, in: S.H. Chan (Ed.), Transport Phenomena in Combustion, vol. 2, Taylor and Francis, Washington, DC, 1996, pp. 921–932.
[54] H.N. Najm, P.S. Wyckoff, O.M. Knio, A semi-implicit numerical scheme for reacting flow: I stiff chemistry, JCP 143 (1998) 381–402.
[55] K. Nerinckx, J. Vierendeels, E. Dick, Mach-uniformity through the coupled pressure and temperature correction algorithm, J. Comput. Phys. 206 (2005) 597–623.
[56] Y. Ogura, N. Phillips, Scale analysis for deep and shallow convection in the atmosphere, JAS 19 (1962) 173–179.
[57] V.I. Oseledets, On a new way of writing the Navier–Stokes equation: The Hamiltonian formalism, Commun. Moscow Math. Soc. 44 (1) (1989) 210–211.
[58] J.H. Park, C.D. Munz, Multiple pressure variables methods for fluid flow at all Mach numbers, Int. J. Numer. Methods Fluids 49 (2005) 905–931.
[59] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flow, Int. J. Heat Mass Transfer 15 (1972) 1787–1806.
[60] V. Pereyra, Iterated deferred corrections for nonlinear operator equations, Numer. Math. 10 (1966) 316–323.
[61] V. Pereyra, On improving an approximate solution of a functional equation by deferred corrections, Numer. Math. 8 (1966) 376–391.
[62] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, William J. Rider, A second-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[63] M.C. Recchioni, G. Russo, Hamiltonian-based numerical methods for a fluid–membrane interaction in two and three dimensions, SIAM J. Sci. Comput. 19 (1998) 861–892.
[64] J.M. Reisner, D.A. Knoll, A.A. Wyszogradzki, An implicitly balanced hurricane model with physics based preconditioning, Monthly Weather Rev. 133 (2005) 1003–1022.
[65] Y. Ren, Y. Jiang, M. Liu, H. Zhang, A class of fully third-order accurate projection methods for solving the incompressible Navier–Stokes equations, Acta Mech. Sin. 21 (6) (2006) 542–549.
[66] G. Russo, P. Smereka, Impulse formulation of the Euler equations: general properties and numerical methods, J. Fluid Mech. 391 (1999) 189–209.
[67] Th. Schneider, N. Botta, K.J. Geratz, R. Klein, Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flow. Technical Report SC 98-31, ZIB, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1998.
[68] Th. Schneider, N. Botta, K.J. Geratz, R. Klein, Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flow, J. Comput. Phys. 155 (1999) 248–286.
[69] Guang Shan Jiang, Chi-Wang Shu, Efficient implementation of weighted eno schemes, J. Comput. Phys. 126 (1996) 202–228.

[70] C. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws.
[71] D.M. Summers, A.J. Chorin, Numerical vorticity creation based on impulse conservation, Proc. Natl. Acad. Sci. USA 93 (1996) 1881–1885.
[72] R. Temam, Sur lapproximation de la solution des equations de Navier–Stokes par le methode des fractionaire II, Arch. Rational Mech. Anal. 33 (1969) 377–385.
[73] D.R. van der Heul, C. Vuik, P. Wesseling, Conservative pressure-correction method for flow at all speeds, Comput. Fluids 32 (2003) 1113–1132.
[74] J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Comput. 7 (3) (1986) 870–891.
[75] B. Van Leer, Towards the ultimate conservative difference scheme IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[76] S. Vater, R. Klein, Stability of a cartesian grid projection method for zero froude number shallow water flows, Numerische Mathematik (2007), submitted for publication.
[77] P. Wesseling, Principles of Computational Fluid Dynamics, Springer-Verlag, Berlin, Heidelberg, New York, 2001.
[78] P.E. Zadunaisky, A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations, in: G. Contopoulus (Ed.), The Theory of Orbits in the Solar System and in Stellar Systems, Proceedings of International Astronomical Union, Symposium, vol. 25, 1964.