Computers and Fluids 175 (2018) 230–245

Contents lists available at ScienceDirect

Computers and Fluids

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

The barely implicit correction algorithm for low-Mach-Number flows

Xiao Zhang*, Joseph D. Chung, Carolyn R. Kaplan, Elaine S. Oran

Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742, USA

ARTICLE INFO

Article history:
Received 20 April 2018
Revised 17 August 2018
Accepted 21 August 2018
Available online 27 August 2018

Keywords:
Compressible Navier–Stokes equations
Low-Mach-Number
Pressure correction
Operator splitting
Swirling flows
Vortex breakdown

ABSTRACT

A new Barely Implicit Correction (BIC) algorithm is presented for the simulation of low-Mach-number flows. This new algorithm is based on the original, introduced by G. Patnaik et al. [G. Patnaik, R. H. Guirguis, J. P. Boris and E. S. Oran, A barely implicit correction for flux-corrected transport. In: Journal of Computational Physics 71.1 (1987), pp. 120], which was a solution procedure including an explicit predictor step to solve the convective portion of the Navier–Stokes equations and an implicit corrector step to remove the acoustic limit on the integration time-step. The explicit predictor uses a high-order monotone algorithm while the implicit corrector solves an elliptic equation for a pressure correction to equilibrate acoustic waves. In this paper, we develop and extend BIC for multidimensional viscous flows. We introduce a new filter to further stabilize the algorithm and clarify the solution procedure for the inclusion of the viscous fluxes. The new algorithm is examined in three test problems with successively increased difficulty. First, a two-dimensional lid-driven cavity flow is simulated to demonstrate the ability of BIC on solving steady-state swirling flows. Using time steps at least 100 times larger than the explicit limit, good agreements are obtained for solutions when compared with an incompressible calculation by a prior work. A two-dimensional (2D) doubly periodic shear layer flow is simulated to examine the algorithm on solving a transient flow with strong vorticity gradients. Finally, vortex breakdown in three-dimensional (3D) swirling flows are used to further test the stability and performance of the new BIC algorithm. Comparisons of explicit and implicit BIC calculations of both the 2D doubly periodic shear layer and 3D vortex breakdown are presented side by side. They demonstrate that the new BIC algorithm is able to predict accurate and robust solutions using time steps varying from near the explicit stability limit to tens and hundreds of times larger. Excellent agreement is also obtained when compared with results from other algorithms. We discuss our observations of these computations and features which were found to be critical for robustly simulating low-speed, highly dynamic flows.

1. Introduction

Over the past forty years, many high-order monotone algorithms have been developed and widely applied to the simulation of compressible gas dynamics. These algorithms were originally designed to overcome problems associated with calculating shocks and contact discontinuities accurately. During this time, and especially more recently, these algorithms have been used not only to compute supersonic flow, but also to compute subsonic flows with turbulence and a range of reactive flow from low-speed flames to high-speed detonations. A number of these monotone methods have been reviewed in [1].

High-order monotone algorithms were designed to compute high-speed flows using an explicit time-integration scheme. In this approach, numerical stability is then ensured by restricting the time step, which is based on the sound speed and convective velocity. In supersonic flows, the convective velocities are comparable with the sound speed, so there is no serious limitation on the computation efficiency from using an explicit method. In low-speed flows, however, the convective velocities can be much smaller than the sound speed. This can make simulations of low-speed flow prohibitively expensive when using explicit algorithms.

The motivation of the algorithm described in this paper is the need to calculate low-speed flows accurately and economically. In this flow regime, fluid velocities range from centimeters to tens of meters per second, which could be hundreds of times smaller than the sound speed. If the sound speed restriction in the stability limit could be removed, the computational cost would be

* Corresponding author.
E-mail address: xiaoz@terpmail.umd.edu (X. Zhang).

https://doi.org/10.1016/j.compfluid.2018.08.019

at least one or two order of magnitude lower. This has been the motivation for developing low-Mach number, implicit algorithms. One way to remove the sound speed limit is to make the time integration implicit, such as MacCormack method [2], semi-implicit ICE method [3], implicit nonlinear PPM [4] and TVD methods [5]. These implicit calculations are accurate but very expensive. Improvements on reducing the computational cost are reported in recent works, including the methods developted by Wall, Pierce & Moin [6] and Degond & Tang [7]. Another technique is the perturbation or asymptotic approach. The physical acoustic waves are decoupled from the system of equations by using regular perturbation theory and applying Taylor series expansion on the variables in power terms of the Mach number. Examples of this approach include the methods developed by Jones & Boris [8], Rehm & Baum [9], Paolucci [10], Pember et al. [11], Nicoud [12] and Thornber et al. [13]. The asymptotic approximation still allows compression and rarefaction over time as long as the Mach number is small enough. The spatial variations in pressure, however, are filtered out, which means acoustic wave effects are eliminated.

The barely implicit correction (BIC), originally proposed by Patnaik et al. [14], eliminated the sound speed restriction by solving the governing equations at a large time step determined by the fluid velocity (predictor step), and then applying a pressure correction that effectively equilibrates the acoustic waves. More specifically, at each time step, a pressure correction is applied to the momentum and energy equations (corrector step). The formulation of the pressure correction allows for the corrector step to scale from explicit to fully implicit. Patnaik et al. used the flux-corrected transport (FCT) for the predictor step value, although in principle, any monotone algorithm should work just as well.

The original BIC algorithm [14] was used for one-dimensional and two-dimensional problems, and, in particular, for reactive flows. Examples include computations of premixed cellular flame structure (Patnaik et al. [15]), and simulations of axisymmetric methane-air diffusion flames (Kaplan et al. [16]). These multidimensional reactive-flow computations produced quantitative results when compared to experiments. Nonetheless, there were numerical issues that caused some problems in the execution, such as small, sometimes growing pressure oscillations that had to be damped. Patnaik et al. [15] filtered these oscillations from the solutions by using a high-frequency pressure filter. Kaplan et al. [16] reduced oscillations to an acceptable level by using a control algorithm for the outflow boundary condition.

In this paper, we develop and extend the BIC algorithm so that it is robust and stable for one-, two-, and three-dimensional viscous flow problems with and without inflow and outflow boundaries. The base monotone algorithm is the same used by Patnaik et al. [14], fourth-order FCT. Here the new procedure for developing the pressure correction is described in detail along with an additional filter step. The new formulation retains the flexibility of the original algorithm to scale from explicit to fully implicit. In this work, we focus on fully implicit performance of the new BIC algorithm. This is tested by computing the flow in a two-dimensional (2D) lid-driven cavity and comparing this solution from a previously published incompressible computation [17], a 2D doubly periodic shear layer flow with comparisons against a "Numerical Acoustic Relaxation (NAR)" method [18] and a "pseudospectral" method [19], and 3D vortex breakdown in a rotating flow which is compared with a previous DNS calculation [20]. For the cases of doubly periodic shear layers and the vortex breakdown, implicit and explicit solutions are compared side by side.

The organization of this paper is as follows. Section 2 presents the basic numerical algorithm for the convective portion of the Navier–Stokes equations (i.e., the Euler equations). This section begins with a description of the original procedure and then shows how this must be changed. Details on how to implement the new algorithm for solving the Navier–Stokes equations are presented in Section 3. In Section 4, we demonstrate the performance of this algorithm using the series of test problems described above. The paper ends in Section 5 with a discussion of some of the general performance metrics and special features of the algorithm.

2. The barely implicit correction algorithm

We begin by describing the original BIC procedure, and then show how it can be stabilized, and finally generalize the solution to include viscous and other source terms. The convective fluxes will be solved here using the flux-corrected transport algorithm (FCT). In the original version of FCT, there is a two-stage Runge-Kutta time integration, that is, a half time step (marching from t^o to $t^o + \Delta t/2$) and a full time step (from t^o to $t^o + \Delta t$), which together give second-order in time. In the BIC algorithm given below, the original time integration for FCT is combined with a new BIC step, so that there is only one time step (from t^o to $t^o + \Delta t$) required. This is explained below.

2.1. Original BIC procedure

First consider the original BIC-FCT approach, which solves the Euler equations that can be written as:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{V}) \tag{1}$$

$$\frac{\partial \rho \mathbf{V}}{\partial t} = -\nabla \cdot (\rho \mathbf{V}\mathbf{V}) - \nabla P \tag{2}$$

$$\frac{\partial E}{\partial t} = -\nabla \cdot (E + P)\mathbf{V} \tag{3}$$

in which t is time, ρ is density, P is pressure, E is total energy, $\mathbf{V}$ is the velocity vector. The equation of state relating pressure and the total energy is

$$E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho \mathbf{V}^2 \tag{4}$$

The inclusion of body forces and source terms will be discussed later.

The procedure outlined below is *almost* the same as [14]. Here, we repeat and recast the steps to clarify the path towards the solution and to facilitate the explanation of the need for the new step added to ensure stability. This explanation below is also necessary to address the more substantive modifications needed and explained in a later section.

The term "barely implicit correction" is used to indicate that only selected terms in the equations are treated implicitly [21]. This same basic concept as used here means that only the pressure in Eq. (2) and velocity in Eq. (3) are treated implicitly. There were two main stages in the algorithm. First, there is an explicit predictor step using a large time-step governed by Courant-Friedrichs-Levy (CFL) condition on the fluid velocity (not the acoustic speed) to solve for predicted values. Then there is a second implicit corrector step that solves an elliptic equation for a pressure correction. They also introduced the implicitness parameter, ω, which allows the algorithm to vary from partially implicit (damping of the sound wave is minimized) to fully implicit (damping of the sound wave is maximized).

These two stages are carried out by a four-step procedure. Below, superscripts "o" denotes the old time step and "n" denotes the new time step. The prime represents the predicted values at the end of the predictor step, and these values are then used by the correction step.

1. Solve for predicted density ρ', and momentum $\rho'\mathbf{V}'$:

$$\frac{\rho' - \rho^o}{\Delta t} = -\nabla \cdot \rho^o \mathbf{V}^o \tag{5}$$

$$\frac{\rho'\mathbf{V}' - \rho^o\mathbf{V}^o}{\Delta t} = -\nabla \cdot \rho^o\mathbf{V}^o\mathbf{V}^o - \nabla P^o \tag{6}$$

Use a monotone method (here FCT with one-step time integration) to solve for predicted density ρ' and momentum $\rho'\vec{V}'$. The time step Δt is the large time-step from the CFL condition governed by fluid velocity (not sound speed).

2. Solve for intermediate energy $\overline{E}$:

$$\frac{\overline{E} - E^o}{\Delta t} = -\nabla \cdot (E^o + P^o)\left[\omega\mathbf{V}' + (1-\omega)\mathbf{V}^o\right] \tag{7}$$

Again, use FCT with the one-step time integration with the same Δt as in step 1 to solve for an intermediate energy $\overline{E}$. The velocity $\mathbf{V}'$ can be calculated by $\rho'\mathbf{V}'/\rho'$. The convective velocity used in this step is a weighted sum of the old and predicted velocity. Note the introduction of the implicitness parameter ω.

3. Solve for pressure correction δP. This is a step introduced for the BIC procedure.

$$\frac{\delta P}{(\gamma-1)\omega\Delta t} - \omega\Delta t\nabla\cdot\left(\frac{E^o + P^o}{\rho'}\right)\nabla\delta P = \frac{\overline{E} - E^o}{\Delta t} - \frac{\rho'\mathbf{V}'^2 - \rho^o\mathbf{V}^{o2}}{2\Delta t} \tag{8}$$

Substitute in predicted density ρ', velocity $\mathbf{V}'$, intermediate energy $\overline{E}$ and properties at old time step into this elliptic equation to solve for δP, which is defined as $\delta P \equiv \omega(P^n - P^o)$. Here γ is the ratio of specific heats.

Eq. (8) was originally derived in [14] by rewriting Eq. (2) with the pressure in an implicit form and Eq. (3) with the velocity in an implicit form, and combining them into one equation. To solve Eq. (8) for a one-dimensional (1D) system, a tridiagonal matrix solver is needed. For a two- or three-dimensional system, an elliptic solver is required. For boundary conditions, the Neumann condition can be applied at symmetry, no-slip wall, or outflow (zero-gradient) boundaries. If the internal energy is constant at a physical boundary, then the pressure at this boundary is constant according to the equation of state. Therefore a Dirichlet condition ($\delta P = 0$) can be applied as the pressure does not vary. If the internal energy varies at a physical boundary, then the boundary condition for δP can be derived as a function of the internal energy based on the equation of state.

4. Correct the momentum and energy using δP:

$$\rho^n\mathbf{V}^n = \rho'\mathbf{V}' - \Delta t\nabla\delta P \tag{9}$$

$$e^n = \frac{\delta P}{(\gamma-1)\omega} + e^o \tag{10}$$

Since the density equation was not treated implicitly, the density at a new time step is equal to the predicted density, that is $\rho^n = \rho'$. Eq. (9) and 10 were given in [14]. Here e represents the total internal energy. Exactly how to update the kinetic energy was not described in the original paper.

Using steps 1–4, we were able to reproduce the two original 1D test problems, which were advection of a contact discontinuity and damping of a sound wave [14]. These two tests involved solution of the Euler equations only. Then, we attempted to use the algorithm as given above to solve the full Navier–Stokes equations by including the diffusion terms that were added through straightforward time-step splitting. The result showed there was a growing, unbounded instability in the solution that could be traced to the coupling with the diffusion terms as well as effects from open boundary conditions.

Simulations of a 2D double shear layer with periodic boundaries on all sides were used to isolate the issues with the additional viscous diffusion terms. The simulations showed the evolution of the two shear layers rolling into large vorticies due to an initial sinusoidal velocity perturbation. The implicit calculation ran, but with strong unphysical oscillations in the temperature and density.

Simulations of a 3D columnar vortex were performed using an inflow boundary with open boundaries on all other sides of the domain. The results showed uncontrolled oscillations and was completely unbounded after only a few time-steps. This type of outflow boundary problem was also encountered by Patnaik et al. [15] and Kaplan et al. [16]. The issue of how to stabilize an outflow will be discussed in more detail below.

2.2. New solution procedure

In order to stabilize the algorithm, we first need to isolate the cause of the instability. This leads us to change our focus from pure convection (Euler equations) to additional physical processes that occur in a flow. When numerically solving the full Navier–Stokes equations using explicit time schemes, timestep-splitting is often used. This means each physical process, that is convection, diffusion, and chemical reactions, is calculated serially in one time step. Then the solutions of all the individual processes are summed, effectively coupling all the physical processes to advance in time. This method, in theory, is correct for small time steps and it works well when using explicit time integration for convection, or when the changes in variables are not significant in one time step. When using implicit algorithms such as BIC, however, the large time step may cause large changes in momentum and energy due to diffusion or reaction (if there is chemical heat release, which will be addressed in a later work). Since the original BIC algorithm was only applied to the convection stage, the changes in momentum and energy from the other physical processes were not explicitly accounted for in the pressure correction. This mismatch is what was causing the instability issue. The pressure-correction procedure has to be modified, in some way to include the changes from other physical processes.

One approach was given in the most recent document of the BIC algorithm [22]. An extra term S was included in the energy equation Eq. (7) as a source term, and this formed the new equation for the intermediate energy:

$$\frac{\overline{E} - E^o}{\Delta t} = -\nabla \cdot (E^o + P^o)\left[\omega\mathbf{V}' + (1-\omega)\mathbf{V}^o\right] + S \tag{11}$$

This term S serves as a "storage" variable which accumulates the change of total internal energy due to diffusion and other processes. Then using this new expression for $\left(\overline{E} - E^o\right)/\Delta t$, the effects of S are then fed into the elliptic equation, Eq. (8). Now δP includes the effects of energy change in all of the physical processes considered.

Then, step 4, which is the correction of momentum, energy and pressure, was given by [22] as:

$$\rho^n\mathbf{V}^n = \rho'\mathbf{V}' - \Delta t\nabla\delta P \tag{12}$$

$$E^n = \frac{\delta P}{(\gamma-1)\omega} + E^o \tag{13}$$

$$P^n = P^o + \delta P \tag{14}$$

This update of variables, however, introduces two inconsistencies. The first is in the kinetic energy computed from the updated

momentum $\rho^n\mathbf{V}^n$ and updated total energy of Eq. (13). This inconsistency manifests itself because the new pressure correction is applied to the old total energy, which does not account for the relatively large new kinetic energy. We show this mismatch by first expanding the old total energy E^o,

$$E^o = \frac{P^o}{\gamma - 1} + \frac{1}{2}\rho^o\mathbf{V}^{o2}. \tag{15}$$

The old total energy is expressed as a sum of the internal and the kinetic energies, with the internal energy written using the equation of state. We then incorporate the pressure correction by substituting Eq. (15) into Eq. (13). Rearranging then gives:

$$E^n = \frac{\omega P^o + \delta P}{(\gamma - 1)\omega} + \frac{1}{2}\rho^o\mathbf{V}^{o2} \tag{16}$$

which means only the internal energy is updated with the pressure correction term, and the kinetic energy does not change from its value at the old time step. The momentum, however, is updated according to Eq. (12). This mismatch violated conservation and we observed that this asynchronism generated nonphysical pressure waves within large momentum gradient regions.

We now introduce one modification to the kinetic energy correction to address the mismatch of energies. We replace Eq. (13) at step 4 with:

$$E^n = \frac{\omega P^o + \delta P}{(\gamma - 1)\omega} + \frac{1}{2}\rho^n\mathbf{V}^{n2} \tag{17}$$

This corrects the kinetic energy from the updated momentum $\rho^n\mathbf{V}^n$. The velocity at the new time step $\mathbf{V}^n$ can be calculated by $\rho^n\mathbf{V}^n/\rho^n$.

The second inconsistency is in Eq. (14), which does not account for the implicitness ω. The corrected form is written as:

$$P^n = P^o + \delta P/\omega. \tag{18}$$

Although adding the extra term S helps account for all the physical processes in the pressure correction, oscillations in pressure still arises when the time step is very large. Large implicit time steps result in regions with strong energy source, and when such a region is under-resolved, numerical instabilities occur and eventually grow unbounded if left unchecked. These oscillations decrease as the time step decreases, and they are effectively gone when the time step is close to a usual explicit time step. Two approaches are often used to suppress such numerical instabilities. These are artificial dissipation by including additional damping terms in the equations [23,24], and more efficiently, spatial filtering which does not affect the main physical gradients in the flow [25–28]. A high-frequency pressure filter is suggested for BIC in [22] to avoid the oscillations:

$$P^{filtered} = P + \alpha\nabla^4 P \tag{19}$$

in which α is a small constant. It was, however, necessary to determine the optimal α on a case-by-case basis, and the filter violates conservation.

Here we eliminate high-frequency oscillations and maintain physical structures and conservation by implementing an extra FCT step [29]. The monotone property of FCT allows for it to act as a high-frequency filter. We use this extra step by passing the conservative variables calculated from step 4 (ρ^n, $\rho^n\mathbf{V}^n$ and E^n) into the FCT routine as inputs, while forcing the convective velocities and all pressure and source terms to be zero. This means no additional convection or pressure work in a computational cell is performed. Furthermore, because all the conservative variables are filtered, this procedure preserves conservation. In our usage, we find that this filter is applicable to most cases without extra tuning or optimizing.

3. Coupling BIC with diffusion processes

So far, we have shown how to use BIC to solve for convective fluxes. To obtain the complete solution to the full set of Navier–Stokes equations, the diffusion processes need to be modeled and included. Here we couple the diffusion with the convection process through time-step splitting procedure mentioned above. This means that, in one time step, the diffusion and convection fluxes are calculated independently, and each process uses the solution from the previous process as initial conditions. There are three major considerations for using time-step splitting, especially when the convection process uses an implicit method. These are: (1) time-step control, (2) the order in which of each physical process is computed, and (3) when and how to update variables. The exact way the processes are updated is explained in detail below.

3.1. Governing equations

We consider the time-dependent, compressible Navier–Stokes equations :

$$\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{V}) \tag{20}$$

$$\frac{\partial(\rho\mathbf{V})}{\partial t} = -\nabla\cdot(\rho\mathbf{V}\mathbf{V}) - \nabla P - \nabla\cdot\hat{\tau} \tag{21}$$

$$\frac{\partial E}{\partial t} = -\nabla\cdot((E+P)\mathbf{V}) - \nabla\cdot\left(\mathbf{V}\cdot\hat{\tau}\right) - \nabla\cdot(K\nabla T) \tag{22}$$

$$\hat{\tau} = \rho\nu\left(\frac{2}{3}(\nabla\cdot\mathbf{V})\mathbf{I} - (\nabla\mathbf{V}) - (\nabla\mathbf{V})^{\dagger}\right) \tag{23}$$

where T is temperature, K is thermal conductivity, I is the identity matrix, $\hat{\tau}$ is the stress tensor. Superscript † denotes the transpose for a matrix. We assume Newtonian fluids and ν is the kinematic viscosity. These equations are closed with the ideal gas equation of state:

$$P = \rho\frac{R}{M_w}T \tag{24}$$

The total energy is calculated using Eq. (4).

3.2. Temporal integration procedure

Fig. 1 summarizes the integration process in one computational time-step, in which the subscript o represents the starting (or "old") value, and subscripts 1, 2, and n denote stages where the variables need to be updated. Accordingly, Table 1 lists the values of all the variables at the end of each stage. The total procedure is now explained step by step.

Step (1) Calculate the global time step, Δt_g.

The global time step is the overall Δt_g that all the physical processes use to advance to the new time-step ($t^o + \Delta t_g$). To ensure the stability of the time integration, this time step is constrained by both the physical process and the choice of numerical algorithm. The time-step limit for each process can be calculated by:

$$\Delta t_{conv} = CFL_{wave}\ \min\left(\frac{\Delta x}{|v| + a}\right), \tag{25}$$

$$\text{or}\quad \Delta t_{conv} = CFL_{fluid}\ \min\left(\frac{\Delta x}{|v|}\right) \tag{26}$$

$$\Delta t_{cond} = C_{cond}\ \min\left(\frac{\Delta x^2}{2\lambda/\rho c_p}\right) \tag{27}$$

Table 1
Update of the variables at each stage.

Stage	Old time step: o	After diffusion: 1	Intermediate stage: 2	New time step: n
Density	ρ_o	ρ_1	$\rho_2 = \rho'$	$\rho_n = \rho_2$
Momentum	$\rho_o V_o$	$\rho_1 V_1$	$\rho_2 V_2 = \rho' V'$	$\rho_n V_n = \rho_2 V_2 - \Delta t \nabla \delta P$
Energy	$E_o = \frac{P_o}{\gamma - 1} + \frac{1}{2}\rho_o V_o^2$	$E_1 = \frac{P_o}{\gamma - 1} + \frac{1}{2}\rho_1 V_1^2$	$E_2 = \bar{E}$	$E_n = \frac{\omega P_o + \delta P}{(\gamma - 1)\omega} + \frac{1}{2}\rho_n V_n^2$
Pressure	P_o	$P_1 = P_o$	$P_2 = P_o$	$P_n = P_o + \delta P/\omega$
Temperature	T_o	$T_1 = P_o/R\rho_1$	$T_2 = T_1$	$T_n = P_n/R\rho_n$
Velocity	V_o	V_1	$V_2 = V_1$	$V_n = \rho_n V_n/\rho_n$

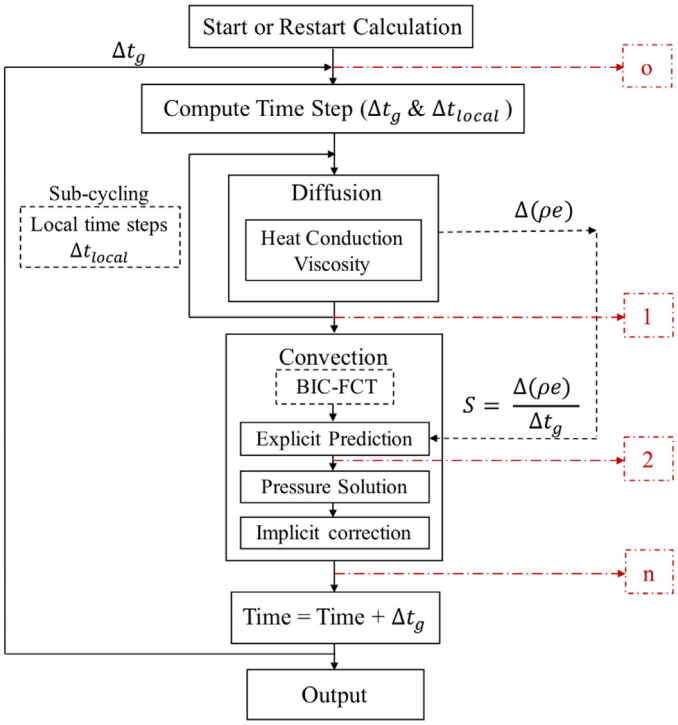

Fig. 1. Flowchart of the coupling of the physical processes in one time-step.

$$\Delta t_{visc} = C_{visc} \min\left(\frac{\Delta x^2}{2\mu/\rho}\right) \tag{28}$$

where v is the fluid velocity, a is the speed of sound, λ is the thermal conductivity, μ is the dynamic viscosity. Here Δx is the computational mesh size and the subscripts *conv, cond*, and *visc* stand for convection, conduction and viscosity. Two *CFL* conditions for the convection process are defined here, in which CFL_{wave} includes the acoustic velocity and CFL_{fluid} is governed only by the fluid velocity. Equivalent conditions for conduction and viscous diffusion processes are defined as C_{cond} and C_{visc}. These *CFL* and the C_{cond}, C_{visc} conditions are dependent on the algorithms chosen for these processes.

If each of these processes, that is convection, thermal conduction, and viscous diffusion, is integrated using an explicit algorithm, then the global time step is determined by the smallest time-step required to ensure stability. When using BIC algorithm for convection, however, the implicit time-step Δt_{conv} could be larger than the other required time-steps. In order to preserve the computational efficiency, the Δt_g is chosen as the implicit time step Δt_{conv} and the other processes are subcycled using local time steps Δt_{local} within the required stability limit.

At this stage, all of the variables have values from the previous time step. This stage is denoted as 'o' which stands for 'old' in Fig. 1 and all the variables are listed in the second column in Table 1 with subscript 'o'.

Step (2) Compute the diffusion effects

After obtaining the local time-step limits Δt_{cond} and Δt_{visc}, subcycle the integration of the heat conduction and viscous diffusion from the old time step, t^o, to the new time step, $t_n = t^o + \Delta t_g$, using a local time-step Δt_{local} for n times, where $\Delta t_{local} = \Delta t_g/n$. The number of the subcycles is determined so that the local time step is not larger than the smallest of these two limits, ($\Delta t_{local} \le \min(\Delta t_{cond}, \Delta t_{visc})$), while the number of the subcycles is minimized. Here the diffusion process is calculated using second-order spatial differencing and high-order Runge-Kutta time integration.

There is a special treatment for updating of variables at this stage. In the new BIC procedure described in Section 2.2, the change of total internal energy due to the diffusion process needs to be extracted and stored temporarily in the variable $S = \Delta(\rho e)/\Delta t_g$, which will be passed into the energy equation Eq. (11) that is solved in the next convection step. At this next convection step, the effect of the change of the total internal energy due to diffusion is included as a pressure effect, by redistributing the pressure correction back into the flow field through the elliptic Eq. (8). Therefore to avoid redundancy, the total internal energy at this stage should temporarily keep the value it had before the diffusion process. This means that the pressure should also stay as P_o, as pressure is a function of the total internal energy. The density, velocity, and momentum should be updated as usual. The total energy should be updated with the old internal energy before the diffusion ($P_o/(\gamma - 1)$) and the new kinetic energy ($\frac{1}{2}\rho_1 V_1^2$). Accordingly, the temperature should be calculated using the old pressure P_o and the new density ρ_1. This is listed in the third column 'After diffusion:1' in Table 1.

Step (3) Compute the convective transport using BIC : explicit predictor

Now perform steps 1 and 2 of the BIC algorithm to calculate the predicted convective fluxes. That is, solve Eqs. (5), (6) and then 11, using the global time step Δt_g and the values updated after the diffusion process. This means the ρ^o, $\mathbf{V}^o$, P^o and E^o in Eqs. (5), (6) and 11 are essentially the values with subscript '1' in Table 1.

This stage after the explicit predictor is denoted as '2' in Fig. 1 and 'intermediate stage: 2' in Table 1. The values ρ', $\rho'\mathbf{V}'$ and $\bar{E}$ in Table 1 are the outputs from the monotone algorithm solver. The notation is kept same as in Eqs. (5), (6) and 11 for consistency. The pressure should have the same value that it had at the beginning of the time step, that is, P_o. Although the density changed at this stage, the temperature and velocity are not updated to save computational efforts as they will not be used in the next corrector step.

Step (4) Compute the convective transport using BIC : implicit corrector

Solve the elliptic Eq. (8) for the pressure correction δP. Then correct the momentum, total energy and pressure using Eqs. (12), (17) and 14, respectively. Here we used a multigrid elliptic solver from BoxLib [30]. The default setting of the solver is employed, where red-black Gauss-Seidel smoother is used for relaxation, bi-

conjugate gradient stabilized algorithm is used for the coarse grid exact solver, and V-cycle is used for restriction and interpolation between fine and coarse grids.

At this stage, which is denoted as 'n', all of the physical processes are at the same stage of integration, and all the flow properties are consistent and synchronized appropriately to the new time $t^o + \Delta t_g$. The values of all of the variables at this stage are listed in the fifth column in Table 1 with subscripts 'n'.

Step (5) Apply a high-frequency filter if necessary

If necessary, a high-frequency filter, which here is an extra FCT step, could be applied here after finishing all the integration processes. Since the FCT filter operates on conservative variables, primitive variables need to be updated after the filtering accordingly to avoid synchronization errors.

4. Applications

In the following, the performance of the BIC algorithm is examined for three test problems. In Section 4.1, simulations of a 2D lid-driven cavity flow demonstrate the ability of BIC on solving steady-state swirling flows. The results are compared with a numerical solution using a vorticity-stream-function formulation of the incompressible Navier–Stokes equations [17]. Section 4.2 describes simulations of a 2D doubly periodic shear layer. These are used to examine the behavior of the BIC algorithm when it is used to simulate transient flows with strong vorticity gradients. The results are qualitatively compared with a "Numerical Acoustic Relaxation (NAR)" method [18] and quantatively compared with a pseudospectral method [19]. In Section 4.3, the BIC algorithm is applied to a 3D vortex breakdown problem, which validates its ability of predicting the instabilities that occur in swirling jet flows. The results are qualitatively comparable with a previous DNS simulation [20].

For all of the test problems in this paper, the implicitness parameter $\omega = 1.0$ is used. The original BIC paper tested the prior version of the algorithm for variable ω and showed the damping and dispersion effects of ω on sound waves. For all of the low-Mach number flows of interest to us here, $\omega = 1.0$ is adequate. We leave the investigation of variable ω to future work when we discuss reactive flows.

In this work, the LCPFCT [31] version of the FCT implementation with a fully multidimensional limiter by Zalesak [32] with a modification from DeVore [33] is employed for the convective solver. We have also changed the numerical diffusion and antidiffusion coefficients, ν and μ (see Eq. 3.19 in [31]), of the LCPFCT algorithm to ensure the 3D implementation remains stable. The algorithm is stable in 3D when using

$$\nu = \frac{1}{12} + \frac{1}{3}\epsilon^2 \tag{29}$$

$$\mu = \frac{1}{12} - \frac{1}{6}\epsilon^2 \tag{30}$$

where ϵ is the local CFL_{fluid}. This choice of ν and μ is related to the stability of explicit LCPFCT in 3D and will be discussed in a future work.

4.1. 2D Lid-driven cavity

This simulation considers flow in a 2D square cavity with no-slip boundary conditions on the lower and side walls, and an upper wall moving with a constant uniform velocity. A primary vortex, driven by shear forces, eventurally forms at the center of the cavity. It is accompanied by secondary vortices at corners. The flow pattern is shown in Fig. 2 as streamlines superimposed on the field

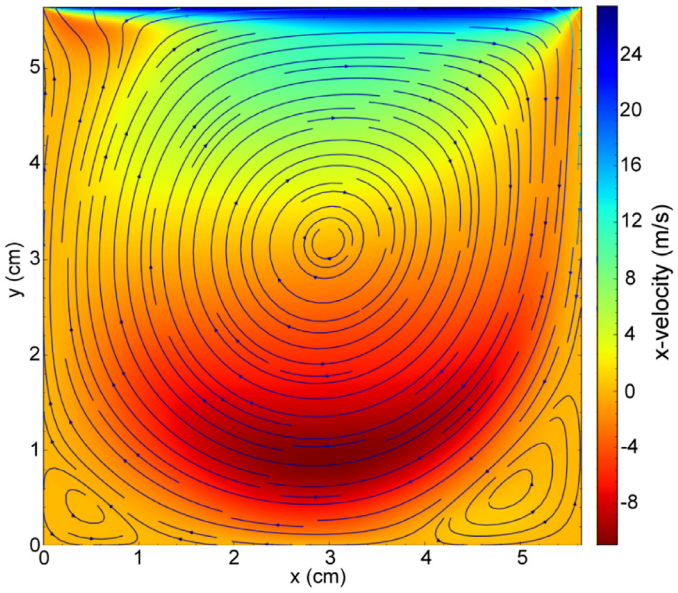

Fig. 2. Streamline superimposed on the contour of x-velocity for case 3 with $Re = 1000$, $CFL_{wave} = 86$.

of velocity in the horizontal direction (x-velocity). The lid-driven flow is a classical test problem for validation of numerical methods and computational codes. Previous numerical results are reviewed in [34]. Here we compare our simulations with the data from Ghia et al. [17], who solved a vorticity-stream-function formulation of the 2D incompressible Navier–Stokes equations using a finite-difference method.

Three different implicit time steps using BIC are tested by holding $CFL_{fluid} = 0.5$ as a constant and varying sound speed through flow temperatures from 300 K to 3, 000 K and 30, 000 K. The resultant CFL_{wave} equals to 13, 38 and 86 respectively. Other than the temperature, the other flow properties are initialized to model dry air at standard condition (300 K, 1 atm). The values of the parameters are summarized in Table 2 below. The test cases are performed for $Re = 1000$. Using all the values in Table 2, we derive the lid velocity $U_{lid} = 27.775$ m/s and the length of the square cavity $H = 0.0565$ m. A non-slip wall boundary condition is applied at all the four boundaries. The upper wall has a constant velocity U_{lid}. All of the calculations are performed on a uniform 256×256 Cartesian mesh.

Fig. 3 presents the velocity profiles (x-velocity and y-velocity) at the vertical and horizontal centerlines of the cavity for cases 1 with $CFL_{wave} = 13$. The effect of the fourth order FCT filter is tested in this calculation. The steady state results show good agreement when compared with the reference data from Ghia et al. [17]. The results in Fig. 3 show that the application of the filter does not affect the accuracy of the calculation, and does not change the flow structure.

Fig. 4 and 5 show the velocity profiles for case 2 with $CFL_{wave} = 38$ and case 3 with $CFL_{wave} = 86$. These calculations with relatively high CFL_{wave} conditions are performed with the filter. The steady state results for both cases agree well with the reference data [17].

4.2. 2D Doubly periodic shear layers

The simulations of 2D double shear layers with periodic boundaries on all sides are performed in a unit domain $[0, 1] \times [0, 1]$. In this domain, a horizontal jet is initialized with small vertical perturbations, and at the jet boundaries, the two shear layers roll up into large vortices as the flow evolves. This configuration was originally introduced by Bell, Colella & Glaz [35], and further examined

Table 2
Properties of the flow for three test cases: case 1 with $CFL_{wave} = 13$, case 2 with $CFL_{wave} = 38$, and case 3 with $CFL_{wave} = 86$. The Re is defined as $U_{lid}H/\nu$, and Ma is defined as $U_{lid}/\sqrt{\frac{\gamma R_u T}{M_{w,air}}}$.

	Re	Ma	T [K]	P [atm]	$M_{w,\ air}$ [kg/mol]	γ	ν [m^2/s]	Pr
Case 1	1000	0.08	300	1.0	28.97	1.4	1.568×10^{-5}	0.711
Case 2	1000	0.0253	3000	1.0	28.97	1.4	1.568×10^{-5}	0.711
Case 3	1000	0.008	30,000	1.0	28.97	1.4	1.568×10^{-5}	0.711

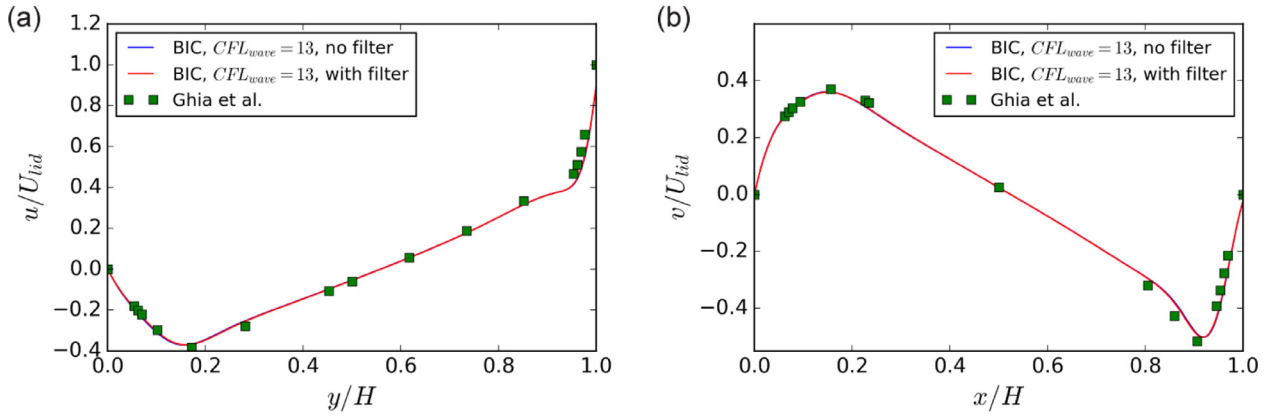

Fig. 3. Comparison of the velocity profiles at the cavity's centerlines for $Re = 1000$, $CFL_{wave} = 13$ with and without filter. (A) x-velocity along the vertical centerline. (B) y-velocity along the horizontal centerline. Squares: data from calculation using $(\omega - \psi)$ formulation (Ghia et al. [17]); Line plots: implicit calculation using BIC (present work). For this calculation, with and without the filter give same results to numerical accuracy.

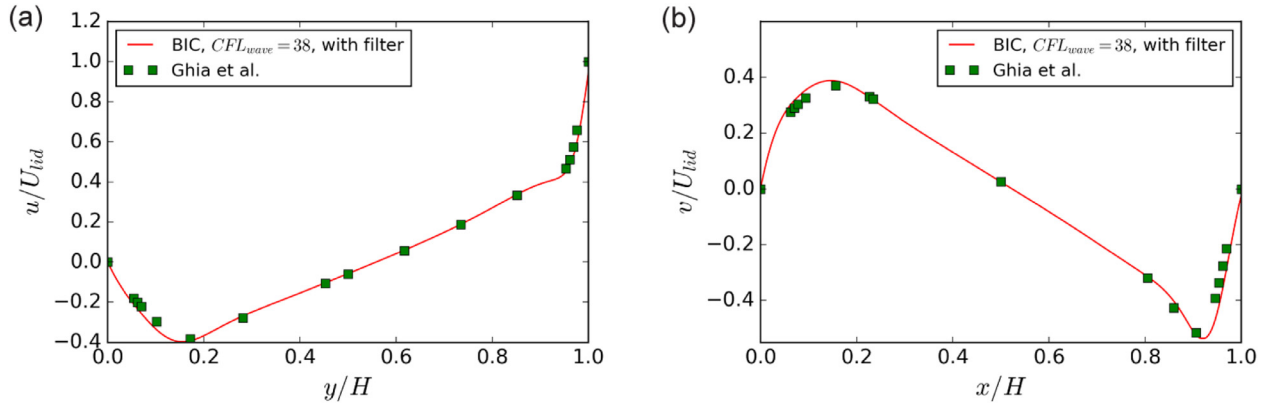

Fig. 4. Comparison of the velocity profiles at the cavity's centerlines for $Re = 1000$, $CFL_{wave} = 38$ with filter. (A) x-velocity along the vertical centerline. (B) y-velocity along the horizontal centerline. Squares: data from calculation using $(\omega - \psi)$ formulation (Ghia et al. [17]); Line plots: implicit calculation using BIC (present work).

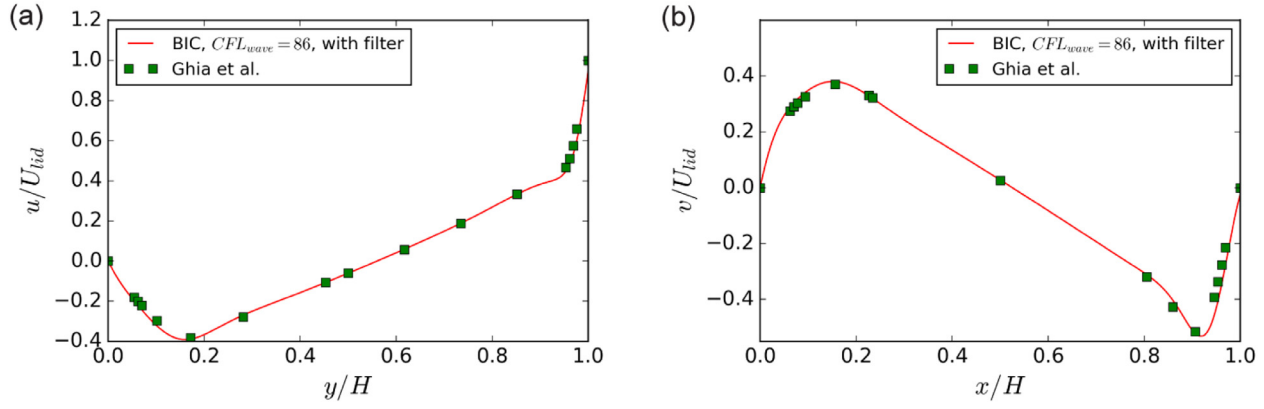

Fig. 5. Comparison of the velocity profiles at the cavity's centerlines for $Re = 1000$, $CFL_{wave} = 86$ with filter. (A) x-velocity along the vertical centerline. (B) y-velocity along the horizontal centerline. Squares: data from calculation using $(\omega - \psi)$ formulation (Ghia et al. [17]); Line plots: implicit calculation using BIC (present work).

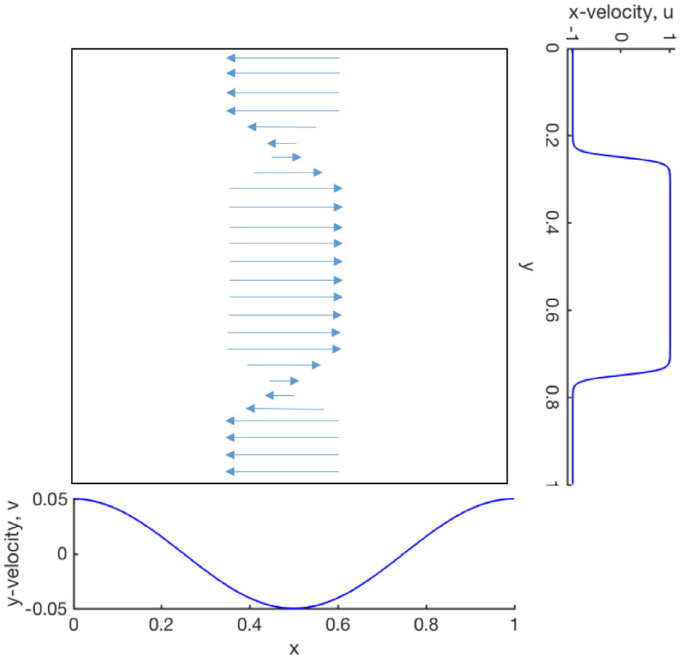

Fig. 6. Initial velocity conditions for doubly periodic shear layers.

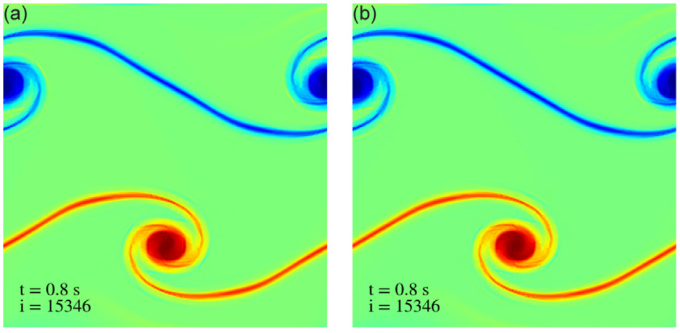

Fig. 7. Vorticity fields for calculations using explicit FCT and implicit BIC with $CFL_{wave} = 0.9$. The implicit BIC calculation is performed without the filter.

by Minion & Brown [19], Nourgaliev, Dinh & Theofanous [18] to compare various numerical methods for stability and accuracy, as the large vorticity gradients in this setup are sensitive to the numerical algorithms. The initial velocity condition is given by

$$u = \begin{cases} \tanh(\vartheta\,(y - 0.25)), & \text{for } \; y \leqslant 0.5 \\ \tanh(\vartheta\,(0.75 - y)), & \text{for } \; y > 0.5 \end{cases}$$
$$v = \gamma \sin(2\pi\,(x + 0.25)) \qquad (31)$$

where ϑ is the parameter that controls the width of the shear layer. The initial perturbation uses the lowest wavenumber with an amplitude γ. The other flow properties are initially uniform throughout the domain. In this work, calculations are performed for a "thin layer" configuration $\vartheta = 80$, with a perturbation strength $\gamma = 0.05$ at $Re = 10,000$. The initial velocity profiles are shown below in Fig. 6:

The different computations are performed on uniform Cartesian grids, 256×256 and 512×512. A summary of the vorticity maps for a time late into the calculation is shown by Figs. 7–9. Quantitative comparisons of the decay of total enstrophy and total kinetic energy with the results obtained by the pseudospectral method of Minion & Brown [19] are shown in Fig. 10. Their pseudospectral method solves the incompressible Navier Stokes equations using a projection method in spectral space and a fourth-order Runge-Kutta method for time integration.

Fig. 7 compares the results of simulations using explicit FCT and implicit BIC, with the same time step governed by $CFL_{wave} = 0.9$. This is close to the explicit stability limit for this low-speed flow problem. The vorticity fields show good agreement between the explicit and implicit results. The vorticity fields are also in qualitative agreement with those shown by Nourgaliev, Dinh & Theofanous [18]. In Fig. 10, the dissipation profiles of the total enstrophy and the total kinetic energy for both the explicit and the implicit simulations agree closely with those shown by the pseudospectral method on a 768×768 grid. The difference between the explicit and the implicit calculation for the total enstrophy is about 0.16%, and in the total kinetic energy is about 0.019%.

Fig. 8(a) shows the result of the calculation using BIC with $CFL_{wave} = 32$ (the corresponding $CFL_{fluid} = 0.1$) on the 256×256 grid. In addition to the two large main vortices, there are "wrinkles" in the shear layers. A closer examination of these shows that they are small vortices. These vortices were also found and discussed by Minion & Brown [19], who showed that when the flow is under-resolved, spurious vortices form due to perturbations with higher wavenumbers imposed by the truncation errors from the numerical discretization. When a finer mesh with 512×512 is used with $CFL_{wave} = 32$, the spurious vortices disappear as shown in Fig. 8(b). Another calculation is performed on the 512×512 grid using an even larger time step with $CFL_{wave} = 58$ (the corresponding $CFL_{fluid} = 0.2$), as displayed in Fig. 8(c). In general, the vorticity fields of all cases in Figs. 7 and 8 agree well with each other, both in terms of the structure of the main vortices and the shear layer thickness. It indicates, however, that the implicit calculations may require a higher resolution than the explicit FCT to resolve the flow features when using large implicit time steps. This is possibly due to the relatively low-order calculation for the pressure correction in BIC compared with the accuracy of the fourth-order FCT (when calculating the pressure correction, a second-order discretization is used for the Laplacian-like term in the elliptic equation (Eq. (8)), and a 3-point stencil for each dimension for the multigrid solver is selected in Boxlib).

As shown in Fig. 10, all of the dissipation curves for calculations using BIC with large time steps agree closely with the pseudospectral method on a 768×768 grid. The good agreement validates the ability of BIC to predict consistent results using various time steps. Some minor differences, however, do appear in the comparison of the cases with different time steps in Fig. 10. With the same resolution, calculations with larger time steps show slightly faster dissipation, which is seen in the comparison of the cases on grid 256×256 with $CFL_{wave} = 0.9$ and $CFL_{wave} = 32$, and the cases on grid 512×512 with $CFL_{wave} = 32$ and $CFL_{wave} = 58$ in Fig. 10.

We now consider the effects of further increasing the time step. A relatively large time step with $CFL_{wave} = 150$ (the corresponding $CFL_{fluid} = 0.5$) is used for an implicit simulation on a 512×512 grid without the filter. The result is shown in Fig. 9(a), in which numerical instabilities appear inside and around the outer-edge of the vortices. These instabilities can be seen in the vorticity fields, and are presented more clearly in the vorticity contours. These oscillations cause a faster decay of both the total enstrophy and the kinetic energy compared with the curves for other stable calculations as shown in Fig. 10. Intended to stabilize the calculation, the filter is then applied to re-calculate this case using the same time step with $CFL_{wave} = 150$. As shown in Fig. 9(b), applying the FCT filter helps eliminate the oscillations outside of the vortices, which brings the dissipation curves in Fig. 10 slightly closer towards the other stable solutions than the curves of the case without the filter. There is, however, still a noticeable difference between the dissipation curves. Very small oscillations around the edges of the vortices appear when a closer examination is taken of Fig. 9(b).

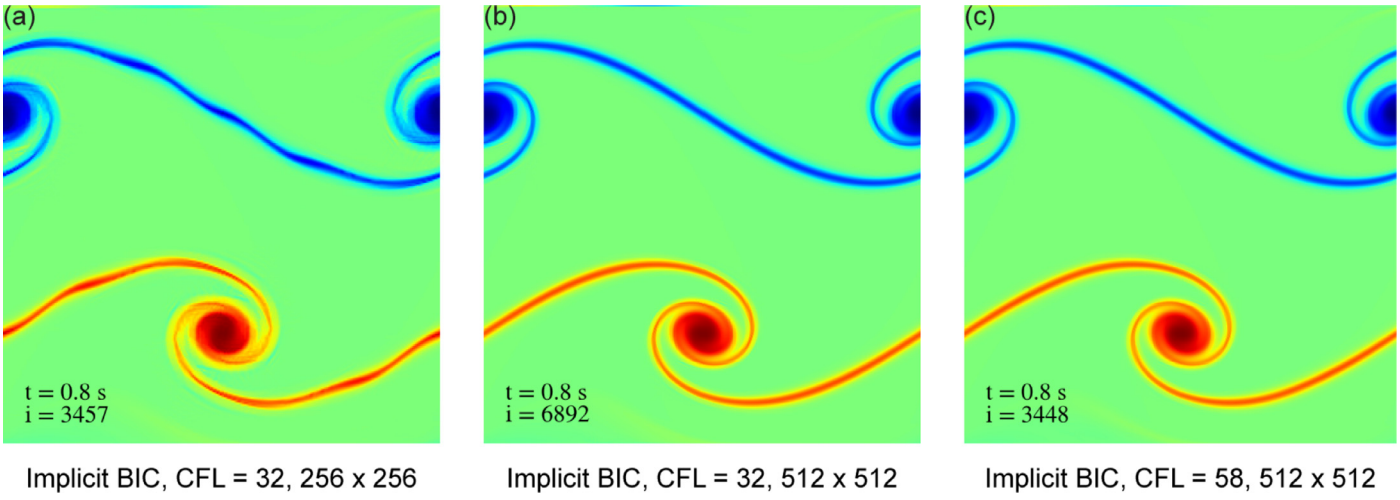

Fig. 8. Vorticity fields for implicit BIC calculations with different time steps and grids. The FCT filter is applied on all of these three cases.

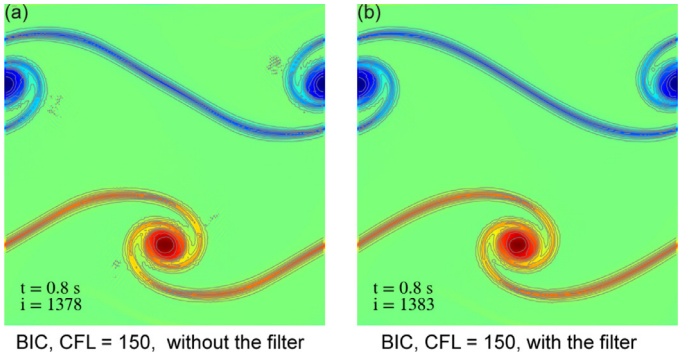

Fig. 9. Vorticity fields with superimposed vorticity contours for implicit BIC calculations with $CFL_{wave} = 150$ on a 512×512 grid, (a) without the filter and (b) with the filter.

Despite these differences, the actual difference of the total kinetic energy in value is only about 0.51% compared with other calculations using smaller time steps. This decrease of the kinetic energy causes a 0.4 K increase in temperature. Nevertheless, this case suggests that the filter can help mitigate spurious oscillations, which allows for more acceptable solutions to be obtained.

4.3. 3D Vortex breakdown

A 3D vortex breakdown problem was computed to test the ability of BIC to predict the instabilities that occur in swirling jet flows. In swirling jets, the vortex structure may be affected by disturbances imposed by the evolution of fluid dynamics, physical boundaries, and temperature. These disturbances can cause adverse pressure gradients on the vortex core. When the axial momentum of the flow is not sufficient to overcome the force generated by the adverse pressure gradient, the vortex structure can form a new stable state. This state is characterized by a stagnation point on the central axis of the vortex with a recirculation zone around it. This change in the vortex structure is referred to as “vortex breakdown.”

Based on distinctive internal structures, vortex breakdown was characterized into three types by Sarpkaya [36]: the spiral mode, the bubble mode, and the double-helix mode. Seven types were reported by Faler & Leibovich [37], which include more intermediate states. Extensive research has been done over the past on this

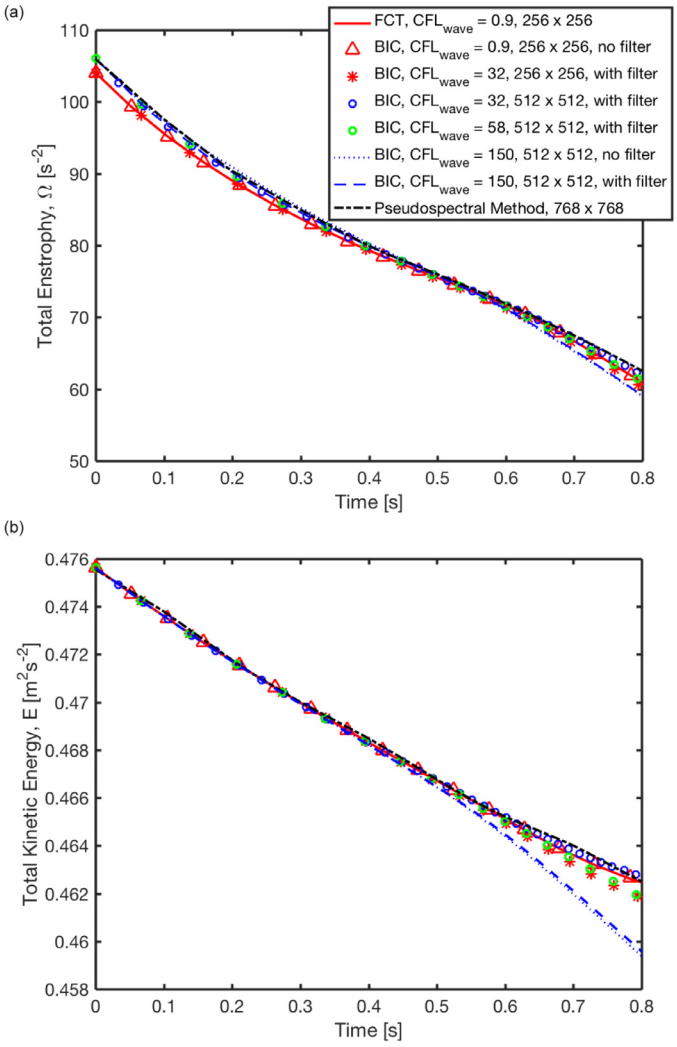

Fig. 10. Time history of (a) total enstrophy, $\sum_{i,j} \frac{\omega_{i,j}^2}{2} \frac{\Delta A_{i,j}}{A_{\text{total}}}$, and (b) total kinetic energy, $\sum_{i,j} \frac{|u|_{i,j}^2}{2} \frac{\Delta A_{i,j}}{A_{\text{total}}}$ for doubly periodic “thin” shear layer tests at $Re = 10,000$. Comparison of explicit FCT, implicit BIC calculations with the solution by the psedospectral method (Minion & Brown [19]).

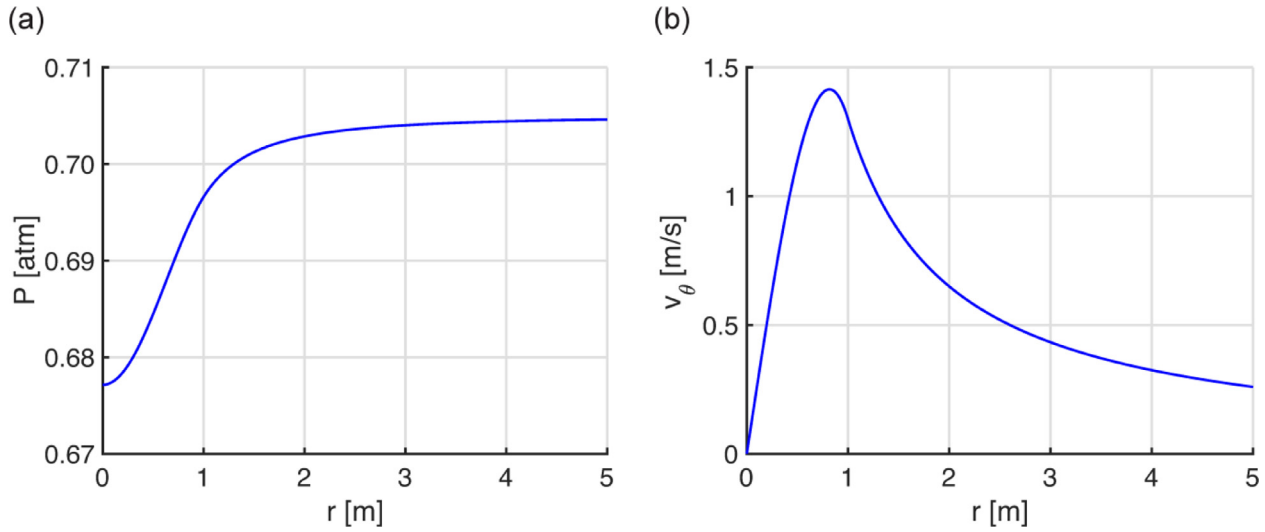

Fig. 11. Grabowski vortex profile with $S = 1.3$, $\alpha = 1$: (a) Pressure distribution for the Grabowski vortex profile; (b) Azimuthal distribution for the Grabowski vortex profile.

phenomenon, which leads to a conclusion that the types of breakdown are mainly controlled by the flow Reynolds number and the swirling level. Here, we adopt one set of the flow parameters from a DNS simulation by Ruith et al. [20]. Under this selected flow condition, the vortex undergoes three major types of breakdown.

In the calculations presented below, the flow is initialized with a "Grabowski vortex" profile, which is adopted from [20], originally introduced by Grabowski & Berger [38]. The azimuthal, radial, and axial velocities vary with the radial location r:

$$v_\theta(0 \leqslant r \leqslant 1) = Sr(2 - r^2) \tag{32}$$

$$v_\theta(1 \leqslant r) = S/r \tag{33}$$

$$v_r(r) = 0 \tag{34}$$

$$v_z(0 \leqslant r \leqslant 1) = \alpha + (1 - \alpha)r^2(6 - 8r + 3r^2) \tag{35}$$

$$v_z(1 \leqslant r) = 1 \tag{36}$$

Here, the swirl number is defined as $S = v_\theta(R)/v_{z,\infty}$, where R is the radius of the vortex core, and the Reynolds number is $Re = v_{z,\infty}R/\nu$. The coflow parameter $\alpha = v_{z,c}/v_{z,\infty}$ describes the axial velocity as a jet-like ($\alpha > 1.0$) or wake-like profile ($\alpha < 1.0$). We set $Re = 300$, $S = 1.3$, $\alpha = 1$, $R = 1$, and $v_{z,\infty} = 1$ m/s, which is one case in [20]. In a swirling flow, the centrifugal force directed outwards should be balanced by the pressure gradient force pointing inwards. Based on the radial force balance,

$$\frac{v_\theta^2}{r} = \frac{1}{\rho}\frac{\partial P}{\partial r} \tag{37}$$

we derive the Grabowski pressure profile outside Eq. (38) and inside Eq. (39) of the vortex core as a function of the radial location:

$$P_{out}(1 \leqslant r) = \rho S^2\left(-\frac{1}{2r^2}\right) + P_\infty \tag{38}$$

$$P_{in}(0 \leqslant r \leqslant 1) = \rho S^2\left[\frac{r^6}{6} - r^4 + 2r^2 - \frac{7}{6}\right] + P_{out}(r = 1) \tag{39}$$

Three simulations are presented in this work:

1. Reference case: Explicit calculation using FCT with $CFL_{wave} = 0.95$. Vortex breakdown in the chosen flow configuration is controlled more by kinematic than thermodynamic effects, in order to maximize the time step, the sound speed was scaled through density while keeping the Mach number below 0.15:
   a) The maximum Mach number is defined as $M_{\max} = v_{\max}/a = v_{\max}/\sqrt{\gamma P/\rho}$, in which $v_{\max}$ is the maximum fluid velocity and a is the sound speed. In order to keep $M_{\max}$ below 0.15, we selected a clean number for ρ as 1000 kg/m^3 so that the resulted far-field pressure $P_\infty \approx 71428$ Pa, which is not too far from the realistic condition. Peak velocity magnitude $v_{\max}$ is obtained from the initial velocity field.
   b) Then the molecular weight is chosen as high as $M_w = 1000$ kg/kmol, so that the temperature is not unrealistically low. The resultant temperature in the initial flow field is $T_{\min} = 8.427$ K and $T_{\max} = 8.573$ K, which is calculated using $T = PM_w/(R_u\rho)$.
2. Implicit calculation using BIC with $CFL_{wave} = 2.0$. This calculation keeps all of the flow properties as the reference case. The corresponding CFL_{fluid} is around 0.2.
3. Implicit calculation using BIC with $CFL_{wave} = 60$. This case intends to test the performance of BIC at higher CFL_{wave} condition. Therefore, the sound speed was scaled up by raising the flow temperature through lowering the density from 1000 kg/m^3 to 1 kg/m^3. This setup results in a similar corresponding CFL_{fluid} around 0.2. Except for the density and temperature, all the other flow properties are same as in the explicit simulation.

The initial pressure and azimuthal velocity profiles are shown in Fig. 11. They are same for all of the cases.

The simulations are performed on a 40 m × 40 m × 20 m domain as shown in Fig. 12, in which we define the z-axis as the axial direction. An inflow condition is imposed on the lower axial $x - y$ plane, as indicated in Fig. 12, with the Grabowski vortex profile as described above. The upper axial $x - y$ plane is treated as non-reflecting. The pressure is controlled at the lateral boundaries by the Bernoulli's equation, since the flow is assumed to be irrotational outside of the vortex core. This boundary pressure is then calculated by $P = P_\infty - \rho V^2/2$, where V is the magnitude of the velocity. The velocity at the lateral boundaries are specified using a first order extrapolation with a zero gradient. For flow leaving the boundaries, we also apply the same first order extrapolation with a zero gradient for the temperature. For flow coming into the

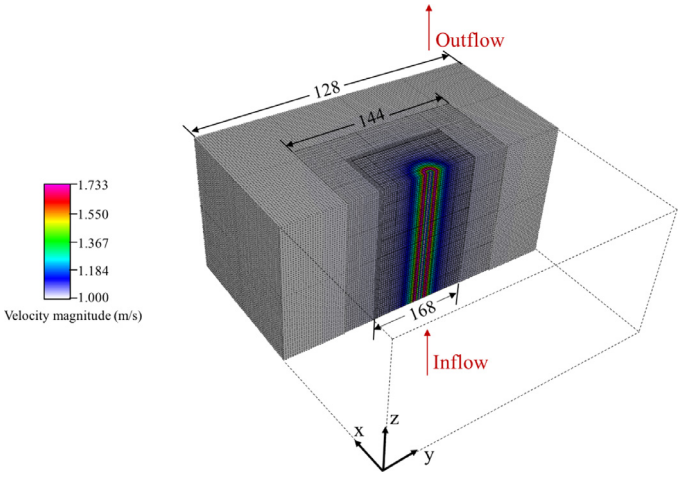

Fig. 12. Computational domain and mesh with superimposed initial velocity field.

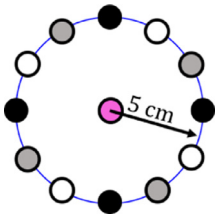

Fig. 13. Layout of the particle injection location.

boundary, we specify the temperature using the ideal gas law with the fixed density discussed earlier.

The flow is initialized with the axisymmetric, columnar Grabowski vortex profile throughout the whole domain, in which the axisymmetry is applied at the center of the $x - y$ plane along the z axis. Cartesian mesh is employed with three levels of refinement. The cell width is halved for each level. The refinement does not change during the calculation. The radial location of refinement is fixed at where the radial pressure gradient is not significant based on Eqs. (38) and (39).

Figs. 14 to 16 show time series for the three simulations. In these figures, the flow streaklines are superimposed on normalized pressure contours, where the dark region indicates higher pressure and the light region indicates low pressure. As the major vortex structure is at the center of the computational domain, only a small central section of the domain of 4 m × 20 m is shown in Figs. 14 to 16. They have been rotated 90° from that shown in Fig. 12, with the flow now going from left to right. The flow particle paths are visualized through streaklines by releasing massless particles at the inflow boundary, from 18 locations distributed evenly along a circle centered at the vortex axis with a 5 cm radius. The particles released on the circle are colored as black, white and grey, which were selected to show different initial locations and the swirling motion. At the center of the vortex, the particles are colored pink. A schematic diagram of these particle injection points is shown in Fig. 13.

In all of the three cases, the flow starts to decelerate in the axial direction at a similar time around 23.0 s, and a similar location on the vortex axis. This deceleration leads to a radial expansion of the vortex core, which eventually results in the formation of the bubble mode. The bubbles in all the cases start to show a conical shape at about 45.0 s. At around 90.0 s, the bubbles are developed to a similar size, and then the downstream instabilities start to occur.

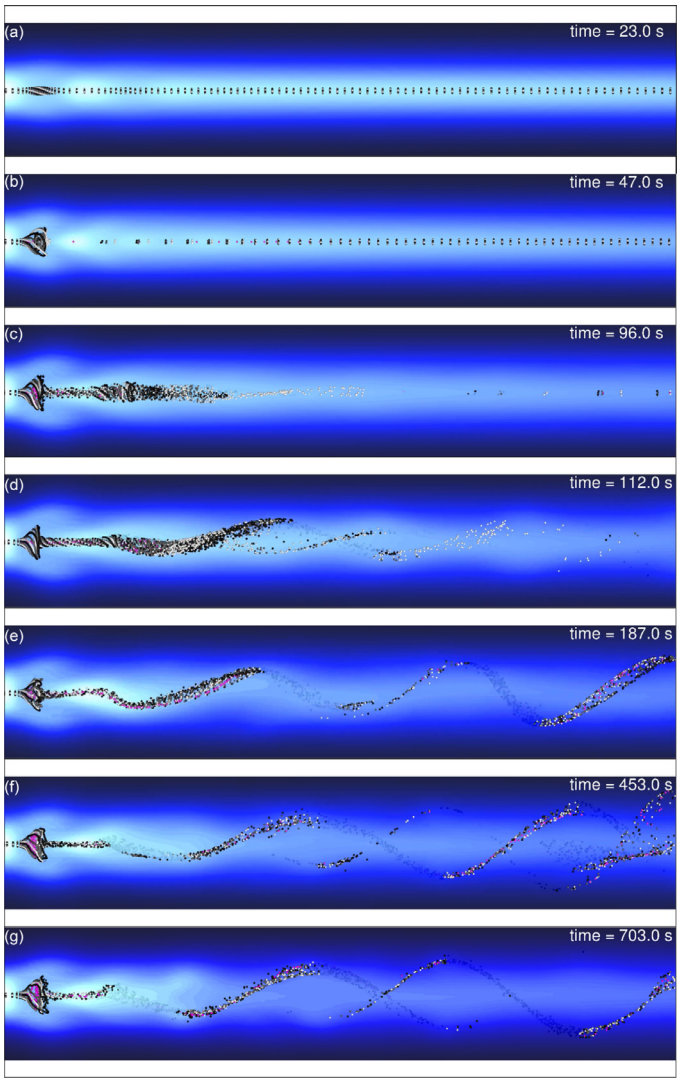

Fig. 14. Streaklines imposed on contours of normalized pressure for the explicit FCT calculation with $CFL_{\text{wave}} = 0.95$. The darker regions indicate higher pressure and the lighter regions indicate lower pressure. Time steps are selected to show the formation and transitions of the instability modes.

Under this flow condition and the current setup of the domain, the downstream secondary instability initially forms the double-helix mode, then transitions to the spiral mode, and finally settles back into the double-helix mode. In Fig. 14, this transition appears in the explicit simulation as shown by the distinct flow structures of an early-stage, relatively narrow double-helix mode at time 112.0s (Fig. 14*d*), a spiral mode at time 187.0s (Fig. 14*e*), and a widely expanded double-helix mode at time 703.0s (Fig. 14*g*). Both implicit calculations are able to predict this transition. As shown in Fig. 15 where the $CFL_{wave} = 2.0$, the flow downstream first forms a narrow double-helix mode at time 118.0s (Fig. 15*d*). After a short period, this mode is then replaced by a spiral mode shown at time 165.0*s* in Fig. 15 (*e*). Eventually, the downstream flow settles into a stable, widely expanded double-helix mode shown at time 369.0s (Fig. 15*f*) and a later time 650.0s (Fig. 15*g*). This stable double-helix mode has a thinner flow structure comparing with the earlier transient one in Fig. 15 (*d*). Similarly in Fig. 16 where the $CFL_{wave} = 60$, the early stage double-helix flow structure appears at time 118.4s (Fig. 16*d*), followed by a spiral mode at time 164.0s (Fig. 16*e*), and eventually transitions to a steady state with a widely expanded double-helix mode shown at time 369.3s (Fig. 16*f*) and

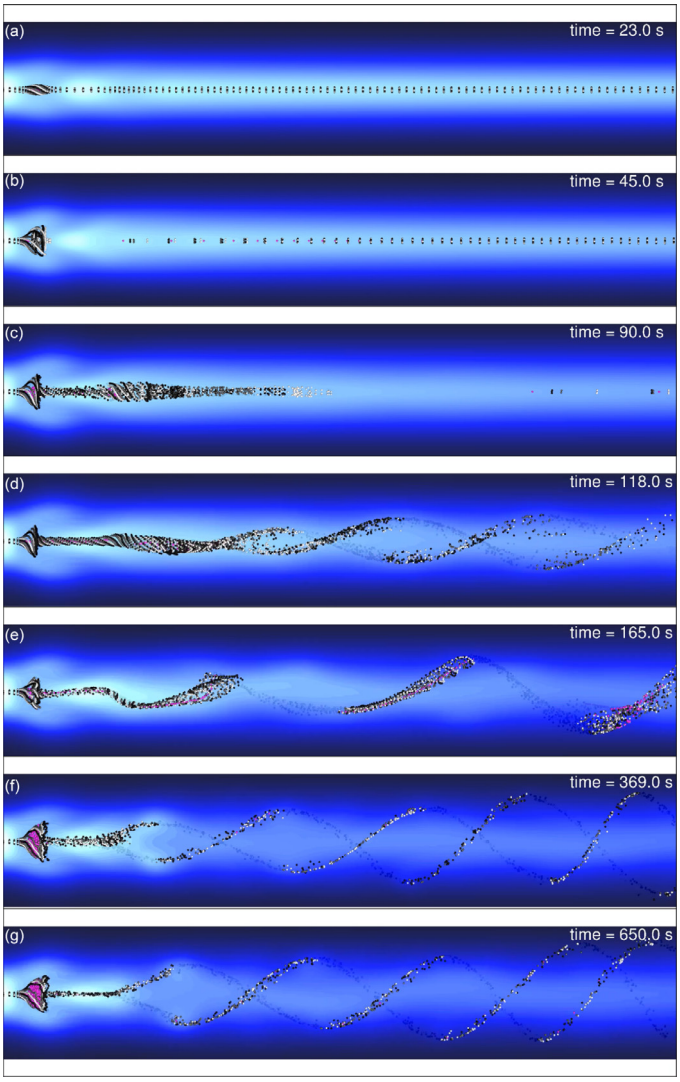

Fig. 15. Streaklines imposed on contours of normalized pressure for the implicit BIC calculation with $CFL_{\text{wave}} = 2.0$. The darker regions indicate higher pressure and the lighter regions indicate lower pressure. Time steps are selected to show the formation and transitions of the instability modes. The FCT filter is applied.

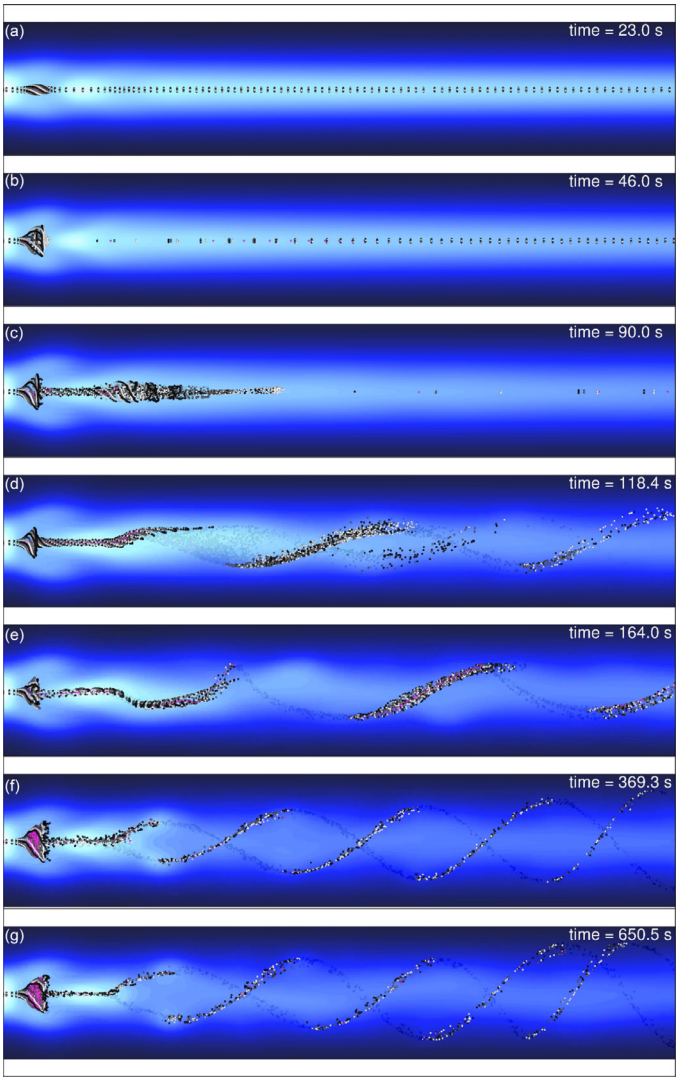

Fig. 16. Streaklines imposed on contours of normalized pressure for the implicit BIC calculation with $CFL_{\text{wave}} = 60.0$. The darker regions indicate higher pressure and the lighter regions indicate lower pressure. Time steps are selected to show the formation and transitions of the instability modes. The FCT filter is applied.

a later time 650.5s (Fig. 16*g*). The results of the two implicit calculations with different CFL_{wave} conditions agree closely with each other, both in terms of the flow structures and the phase accuracy. The flow structures in the explicit calculation and the two implicit calculations also agree quite well with the previous DNS simulation results from Ruith et al. [20]. These computations show the ability of the BIC algorithm to capture and predict the transition of all three major types of the vortex breakdown using large implicit time steps.

Fig. 17 shows a quantitative comparison of the pressure and the axial velocity along the axial direction for the explicit FCT simulation and the two implicit BIC simulations. The profiles are taken at three radial locations, ranging from the vortex axis ($r = 0$ m) to the far field ($r = 8$ m). In Fig. 17 (*a*), the bubble mode in the three calculations is developing while the downstream flow is not affected by the secondary instabilities. The location of the bubble formation is indicated by the minimum velocity on the axial velocity profile at $r = 0$ m. It shows that the location of the bubble is almost the same for all three cases, although the downstream recovery of the vortex is slightly different. This leads to a more pronounced difference in the behavior when the spiral and double-helix modes occur, which is shown in Fig. 17 (*b*). Here, at the centerline $r = 0$ m, we observe a secondary minimum in the axial velocity downstream of the bubble, which corresponds to the bifurcation point of the double helix. This minimum is lower in the implicit calculations than it is in the explicit calculation, which implies the flow experiences a stronger deceleration at the double-helix bifurcation point in the implicit calculations. This is consistent with the stronger adverse pressure gradient near this point in the implicit calculations seen in Fig. 17 (*b*). Although there is a difference between the explicit and implicit simulations, this difference does not grow as CFL_{wave} is increased.

5. Discussion

There are several important features of applying the BIC procedure to an explicit solution of the NS equations. From our experience of using this algorithm, we have identified important points worth discussing. Below, we discuss the use of a spatial high-frequency filter, details of algorithm coupling, the comparison of

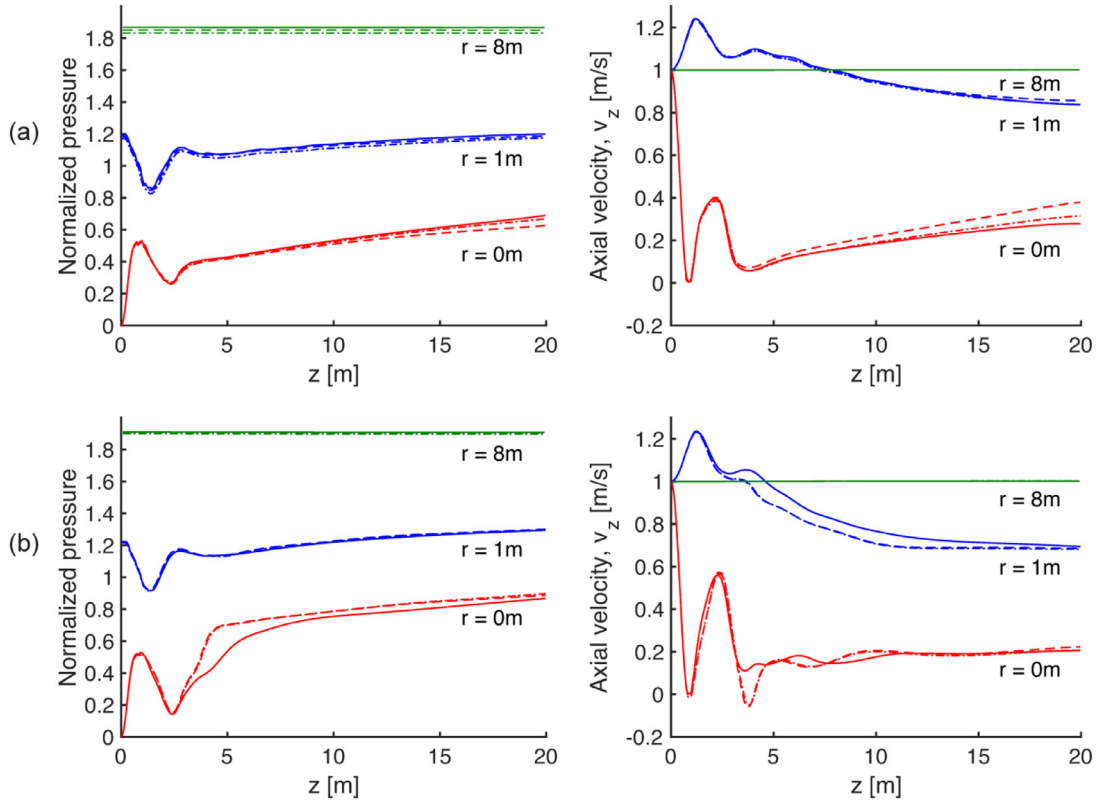

Fig. 17. Time averaged pressure and axial velocity at different radial locations: (a) Time averaged from $t = 85$ to 95 s, when the bubble is developing; (b) Time averaged from $t = 600$ to 650 s, when the downstream instabilities are developed. Solid line: explicit FCT simulation, dashed line: implicit BIC simulation with $CFL_{wave} = 2.0$, and dash-dotted line: implicit BIC simulation with $CFL_{wave} = 60$.

explicit and implicit calculations, the importance of the equation of state, and future directions.

5.1. The filter

We have shown that the new BIC algorithm is stable when the implicit time step is considerably larger than the time step required by the explicit stability limit. Nonetheless, when the time step is too large, numerical oscillations can occur in the solution. A high-frequency filter is necessary in order to control the major numerical instabilities, and therefore to stabilize the calculation while maintaining the accuracy.

The filter used here is an extra FCT step. In previous work [29,39], the FCT algorithm has been used as a post-processing filtering operation to extract a solution from a very noisy direct simulation Monte Carlo calculation. As the effect of the filtering as a post-processing tool is essentially to smooth local peaks in the data, this application inspired us to use FCT as a spatial filter during the calculation to control numerical noise. The details of the filtering process are discussed in [29], and are only briefly explained here to explain why and how it works as a filter.

We begin by explaining some of the basic ideas of the FCT algorithm, which was developed to solve a continuity equation for a variable, ρ:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{V}). \tag{40}$$

The steps of the FCT algorithm globally conserve ρ, do not heighten any existing extrema, and do not introduce any new maxima or minima into the solution. Specifically, the values $\{\rho_i^0\}$ where i indicates the spatial location of ρ, are advanced in time to $\{\rho_i^n\}$ by the following process:

1. The initial values ρ^0 are transported and diffused, giving ρ^{td}. The diffusion here guarantees positivity and stability.
2. Anti-diffusion fluxes $f^{da}(\rho^{td})$ are defined to remove excess diffusion. This antidiffusion stage, however, can introduce negative values or nonphysical overshoots in the solution.
3. The antidiffused fluxes are limited to guarantee positivity and stability, before they are applied to ρ^{td} to find ρ^n. This ensures that no new unphysical maxima or minima are added to the solution.

The effect of these steps is first to smooth local peaks in the data, which alters the local value of ρ and the value of its neighbors. Then, in the limited antidiffusion step, the neighbors keep values closer to their original values, but the peak remains smoothed. The practical result of these steps is a high-frequency filter with some useful properties, as shown in Fig. 18. Here a square wave, propagating at zero velocity ($\mathbf{V} = 0$ in Eq. (40)), is passed through the FCT algorithm using a uniform grid, as shown in Fig. 18*a*. The result is that the square wave is unaltered by FCT, up to numerical round off. Next, if a random spectrum of high-frequency noise is superimposed on the square wave, as shown in Fig. 18*b*, FCT reduces the noise while rigorously conserving ρ. During this process, the quantity ρ is convected at zero velocity. It is the flux limiter and the remnants of the antidiffusion that persist during the filtering. The comparison of the results after 5000 and 20,000 passes shows that the less noisy solution stops changing as the number of filter passes increases. This is because the flux limiting step eliminates local extrema on neighboring points. After these points are smoothed, the flux limiter will no longer affect the solution, thereby stopping the effects of the filtering.

In theory, any high-frequency spatial filter that could help control high-frequency oscillations should work just as well. When the

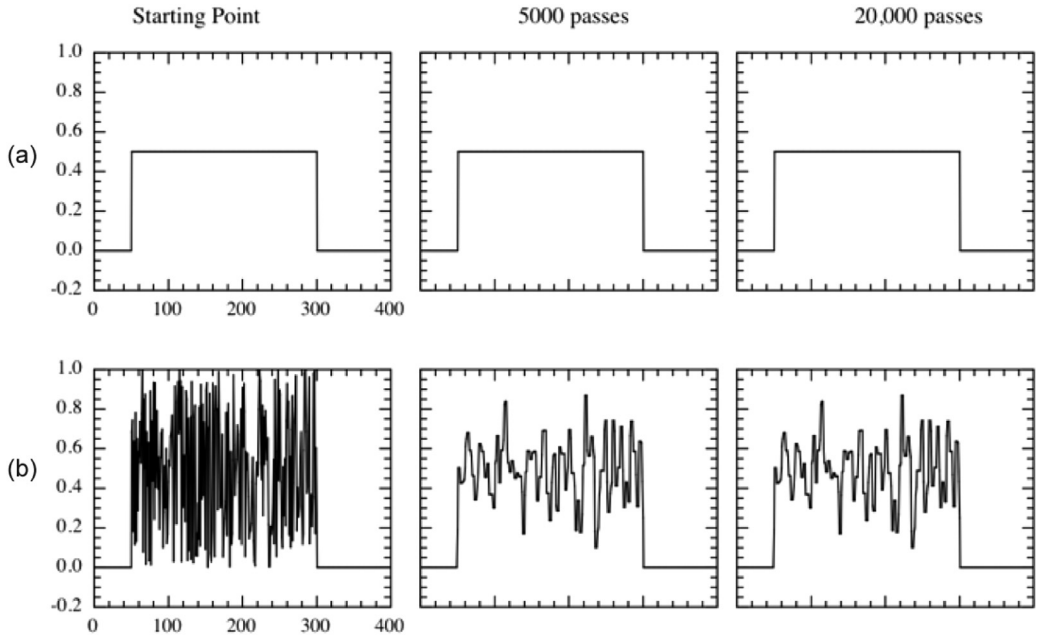

Fig. 18. Reprinted from [29]. Application of FCT filter to square wave. Plain square wave (panel a) and one that has been superimposed with random noise (panel b) are passed through FCT with $\mathbf{V} = 0$. The starting point and the results after 5000 and 20,000 passes through the FCT filter are shown for both cases.

BIC algorithm is combined with FCT, however, using the FCT routine itself as a filter reduces the complexity of implementation.

5.2. The coupling of BIC with explicit algorithms

It is also important to examine the interface between the combined BIC and explicit algorithm, and not only their separate inherent features. An example is the effect of a multiplicative coefficient that is introduced in FCT. This coefficient was introduced to add a very small amount of numerical diffusion in every time integration step, so that no local existing extrema will be enhanced and therefore the monotonicity is ensured. Although the influence of this coefficient was well studied for explicit calculations [40], it affects the results in a different way when BIC is applied.

The amount of the numerical diffusion has been quantified in [40] for explicit FCT. Here, we have seen that when FCT is combined with BIC, the total amount of the numerical diffusion decreases with larger implicit time steps. This is attributed to the inherent benefit of BIC simply having a larger time step: that is, the total number of time integration steps is reduced by using larger time steps to march towards the target physical time. Although the numerical diffusion in FCT imposed by this coefficient is very small, the effect is not negligible when it is close in value to the physical viscosity. Using BIC when it is possible provides a noticeably better result especially when a quantitatively accurate solution is required.

The performance of BIC discussed in this paper is based on the combination of BIC with FCT. When BIC is applied to other methods, the influence of the interface between them should be carefully examined in the analysis of the overall performance and the interpretation of the results.

5.3. Comparison of explicit and implicit calculations

We have compared the results obtained from explicit FCT and implicit BIC calculations side by side for the 2D doubly periodic shear layers problem and the 3D vortex breakdown problem. In both cases, the implicit calculations show excellent agreement with the explicit solutions. The accuracy of the solutions from BIC is shown to be robust using time steps varying from near the explicit stability limit to hundreds of times larger.

Closer examinations of the implicit calculations with large time steps in the 2D doubly periodic shear layers problem show some additional numerical diffusion compared with the explicit solution. In this case, we see a slightly faster decay of the total enstrophy and the kinetic energy when using lager implicit time steps. The additional diffusion observed in the BIC simulations is possibly related to the relatively less accurate elliptic solver for the pressure correction when compared with the fourth-order accurate FCT. This implies that even when a high-order monotone algorithm is used for the explicit prediction, the overall spacial accuracy could be limited by the choice of the elliptic solver for the implicit correction. In this work, we use a second-order discretization for the Laplacian-like term in the elliptic Eq. (8), and a 3-point stencil for each dimension for the elliptic solver from the Boxlib library. A higher order discretization stencil and a more accurate elliptic solver can be used in future applications.

5.4. The equation of state

The BIC algorithm in this paper is derived for ideal gases. For other types of equation of state, the pressure correction equation should be re-derived by finding the corresponding relation of changes of pressure and energy, and then substituting the relation into the implicit forms of the conservation of momentum and energy equations. This derivation should follow the original procedure in [14].

5.5. Future direction

We have demonstrated the ability of the BIC and FCT to provide stable and robust calculations in the range of low-Mach-number flows. The algorithm has been tested on flows with Mach number as low as 0.003. As the Mach number increases to a point where the compressibility effects are significant, the explicit FCT algorithm should be used instead. Future work will develop a method to evolve solutions smoothly between low-Mach-number solution obtained from BICFCT and high-Mach-number solution obtained

from explicit FCT. This could benefit the research of flows which cover a wide range of flow speeds.

6. Summary and conclusions

For the computation of low-Mach-number flows, we have presented a new BIC algorithm based on the original BIC introduced in [14,22]. The original BIC algorithm proposed a solution procedure that includes an explicit predictor step to solve the convective portion of the Navier–Stokes equations and an implicit corrector step to remove the acoustic limit on the CFL condition. The explicit predictor uses a high-order monotone algorithm while the implicit corrector solves an elliptic equation for a pressure correction to equilibrate acoustic waves. The modification described in this paper has several new features. These are: (1) a modification of the energy correction, and (2) a filter step that is operated on all of the conserved variables to remove spurious oscillations. Another contribution in this paper is the description of the integration procedure of BIC with the terms that represent physical diffusion processes. The performance of BIC was tested by combining it with a fourth-order monotone FCT algorithm. Three swirling flows with successively increased difficulty are modeled.

First, a flow in a 2D enclosed cavity with a moving upper wall is simulated using various implicit time steps. This case is selected to demonstrate the ability of BIC on solving steady-state swirling flows. The solutions obtained using BIC are compared with results from an incompressible calculation [17]. In this case, we obtain good agreements for CFL_{wave} number of at least 100.

The second test problem is the evolution of two shear layers into large vortices in a 2D periodic domain. This case is simulated to examine the behavior of BIC when solving transient flows with strong vorticity gradients. For this case, comparisons between explicit FCT and implicit BIC calculations are presented side by side. The accuracy of the solutions using BIC are shown to be robust using time steps varying from near the explicit stability limit to hundreds of times larger. The CFL_{wave} number exceeds 100 before there are even small differences. The solutions from the BIC calculations also show excellent agreement when compared to other algorithms [18,19]. A closer examination on the time history of total enstrophy and the total kinetic energy shows a slightly faster decay of both when a larger time step is used. This additional diffusion is possibly from the relatively lower-order solution of the pressure correction term in BIC compared with the fourth-order explicit FCT. This shows that the choice of the multigrid solver for the elliptic equation can limit the overall spatial accuracy of the algorithm, although high-order explicit methods are used.

The third test problem is a 3D vortex breakdown with an inflow-outflow boundary condition, which tests BIC on predicting the instabilities that occur in swirling jet flows. The highlight in this work is the direct comparison of the explicit and implicit calculations. The bubble modes predicted by BIC with different time steps agree closely with the one obtained from the explicit FCT calculation, in terms of both shape and location. The comparison also shows that BIC is able to predict the downstream spiral mode and the double-helix mode, and capture the transition from one to another. BIC, however, predicts a stronger deceleration at the double-helix bifurcation point.

A spatial filter is sometimes necessary to eliminate high-frequency numerical oscillations and therefore stabilizes the calculations when using large implicit time steps. Fourth-order monotone FCT, used here as the routine for solving convective fluxes, is suggested as a convenient choice for such a required filter. The FCT algorithm could serve as a filter which reduces the complexity of implementation as one FCT routine could work for two purposes. Moreover, it is conservative, and does not require extra tuning or optimization for most applications.

The extension of BIC to reactive flows will be introduced in a subsequent paper. Potential future improvements include the generalization of the pressure correction to account for different equations of states, and an algorithm that transitions between BICFCT and explicit FCT to simulate flows that cover a range of Mach numbers.

Acknowledgments

This work was supported by the Army Research Office (grant W911NF1710524) and the National Science Foundation under award CBET 1839510. This work was also supported in part by the University of Maryland through Minta Martin Endowment Funds in the Department of Aerospace Engineering, and through the Glenn L. Martin Institute Chaired Professorship at the A. James Clark School of Engineering. Computations were performed using the University of Maryland Deepthought2 HPC cluster and the Air Force Research Laboratory's Thunder. The second author is grateful to the Laboratory of Computational Physics, Naval Research Laboratory (NRL), for providing the DoD supercomputing resources to access Thunder. The authors gratefully acknowledge the fruitful discussions with Dr. Jay Boris and Dr. Gopal Patnaik at NRL. The authors also thank Dr. Ryan Houim of the University of Florida for assistance and guidance in the implementation of the numerical algorithms and many other helpful suggestions.

References

[1] Grinstein FF, Margolin LG, Rider WJ. Implicit large eddy simulation: computing turbulent fluid dynamics. Cambridge university press; 2007.
[2] MacCormack RW. A numerical method for solving the equations of compressible viscous flow. AIAA J 1982;20(9):1275–81.
[3] Harlow FH, Amsden AA. Numerical calculation of multiphase fluid flow. J Comput Phys 1975;17(1):19–52.
[4] Fryxell BA, Woodward PR, Colella P, Winkler K-H. An implicit-explicit hybrid method for lagrangian hydrodynamics. J Comput Phys 1986;63(2):283–310.
[5] Yee H, Harten A. Implicit TVD schemes for hyperbolic conservation laws in curvilinear coordinates. AIAA J 1987;25(2):266–74.
[6] Wall C, Pierce CD, Moin P. A semi-implicit method for resolution of acoustic waves in low Mach number flows. J Comput Phys 2002;181(2):545–63.
[7] Degond P, Tang M. All speed scheme for the low Mach number limit of the isentropic Euler equations. Commun Comput Phys 2011;10(1):1–31.
[8] Jones WW, Boris JP. Flame and reactive jet studies using a self-consistent two-dimensional hydrocode. J Phys Chem 1977;81(25):2532–4.
[9] Rehm R, Baum H. The equations of motion for thermally driven, buoyant flows. J Res NBS 1978;83:297–308.
[10] Paolucci S. Filtering of sound from the navier-stokes equations. Sandia National Laboratories Livermore, CA; 1982.
[11] Pember RB, Howell LH, Bell JB, Colella P, Crutchfield WY, Fiveland W, Jessee J. An adaptive projection method for unsteady, low-mach number combustion. Combust Sci Technol 1998;140(1–6):123–68.
[12] Nicoud F. Conservative high-order finite-difference schemes for low-mach number flows. J Comput Phys 2000;158(1):71–97.
[13] Thornber B, Mosedale A, Drikakis D, Youngs D, Williams RJ. An improved reconstruction method for compressible flows with low mach number features. J Comput Phys 2008;227(10):4873–94.
[14] Patnaik G, Guirguis R, Boris J, Oran E. A barely implicit correction for flux-corrected transport. J Comput Phys 1987;71(1):1–20.
[15] Patnaik G, Kailasanath K, Oran E, Laskey K. Detailed numerical simulations of cellular flames. In: Symposium (International) on Combustion, 22. Elsevier; 1989. p. 1517–26.
[16] Kaplan CR, Shaddix CR, Smyth KC. Computations of enhanced soot production in time-varying CH4/air diffusion flames. Combust Flame 1996;106(4):392IN3399–84405.
[17] Ghia U, Ghia KN, Shin C. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. J Comput Phys 1982;48(3):387–411.
[18] Nourgaliev R, Dinh T-N, Theofanous T. A pseudocompressibility method for the numerical simulation of incompressible multifluid flows. Int J Multiphase Flow 2004;30(7):901–37.
[19] Minion ML, Brown DL. Performance of under-resolved two-dimensional incompressible flow simulations, II. J Comput Phys 1997;138(2):734–65.
[20] Ruith M, Chen P, Meiburg E, Maxworthy T. Three-dimensional vortex breakdown in swirling jets and wakes: direct numerical simulation. J Fluid Mech 2003;486:331–78.
[21] Casulli V, Greenspan D. Pressure method for the numerical solution of transient, compressible fluid flows. Int J Numer Methods Fluids 1984;4(11):1001–12.

[22] Patnaik G, Laskey K, Kailasanath K, Oran E, Brun T. FLIC - A detailed, two-dimensional flame model. NRL Memorandum report 1989;6555.
[23] Steger JL. Implicit finite-difference simulation of flow about arbitrary two-dimensional geometries. AiAA J 1978;16(7):679–86.
[24] Jameson A, Schmidt W, Turkel E. Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes. In: 14th fluid and plasma dynamics conference; 1981. p. 1259.
[25] Visbal MR, Gaitonde DV. High-order-accurate methods for complex unsteady subsonic flows. AIAA J 1999;37(10):1231–9.
[26] Gaitonde DV, Shang J, Young JL. Practical aspects of higher-order numerical schemes for wave propagation phenomena. Int J Numer Methods Eng 1999;45(12):1849–69.
[27] Bogey C, Bailly C. A family of low dispersive and low dissipative explicit schemes for flow and noise computations. J Comput Phys 2004;194(1):194–214.
[28] Falissard F. Genuinely multi-dimensional explicit and implicit generalized shapiro filters for weather forecasting, computational fluid dynamics and aeroacoustics. J Comput Phys 2013;253:344–67.
[29] Kaplan CR, Oran ES. Nonlinear filtering for low-velocity gaseous microflows. AIAA J 2002;40(1):82–90.
[30] Boxlib users's guide website.
[31] Boris JP, Landsberg AM, Oran ES, Gardner JH. LCPFCT-A flux-corrected transport algorithm for solving generalized continuity equations. Technical Report. NAVAL RESEARCH LAB WASHINGTON DC; 1993.
[32] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. J Comput Phys 1979;31(3):335–62.
[33] DeVore CR. An improved limiter for multidimensional flux-corrected transport. Technical Report. NAVAL RESEARCH LAB WASHINGTON DC; 1998.
[34] AbdelMigid TA, Saqr KM, Kotb MA, Aboelfarag AA. Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using GPU accelerated code. Alexandria Eng J 2017;56(1):123–35.
[35] Bell JB, Colella P, Glaz HM. A second-order projection method for the incompressible Navier–Stokes equations. J Comput Phys 1989;85(2):257–83.
[36] Sarpkaya T. On stationary and travelling vortex breakdowns. J Fluid Mech 1971;45(3):545–59.
[37] Leibovich S. The structure of vortex breakdown. Annu Rev Fluid Mech 1978;10(1):221–46.
[38] Grabowski WJ, Berger S. Solutions of the Navier-Stokes equations for vortex breakdown. J Fluid Mech 1976;75(3):525–44.
[39] Kaplan CR, Oran E, Aggarwal U. Reducing statistical scatter in DSMC solutions of hypersonic ionizing flows. In: 46th AIAA Thermophysics Conference; 2016. p. 3843.
[40] Book DL, Li C, Patnaik G, Grinstein FF. Quantifying residual numerical diffusion in flux-corrected transport algorithms. J Sci Comput 1991;6(3):323–43.