

A simple and efficient incompressible Navier–Stokes solver for unsteady complex geometry flows on truncated domains



Yann T. Delorme^{a,*}, Kunal Puri^a, Jan Nordstrom^b, Viktor Linders^b, Suchuan Dong^c, Steven H. Frankel^a

^a Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa, Israel

^b Department of Mathematics, Linköping University, Linkoping, Sweden

^c Department of Mathematics, Purdue University, West Lafayette, IN, USA

ARTICLE INFO

Article history:

Received 11 July 2016

Revised 4 February 2017

Accepted 31 March 2017

Available online 5 April 2017

Keywords:

Artificial compressibility method

EDAC

High-order numerical methods

Large Eddy simulation

ABSTRACT

Incompressible Navier–Stokes solvers based on the projection method often require an expensive numerical solution of a Poisson equation for a pressure-like variable. This often involves linear system solvers based on iterative and multigrid methods which may limit the ability to scale to large numbers of processors. The artificial compressibility method (ACM) [6] introduces a time derivative of the pressure into the incompressible form of the continuity equation creating a coupled closed hyperbolic system that does not require a Poisson equation solution and allows for explicit time-marching and localized stencil numerical methods. Such a scheme should theoretically scale well on large numbers of CPUs, GPU's, or hybrid CPU-GPU architectures. The original ACM was only valid for steady flows and dual-time stepping was often used for time-accurate simulations. Recently, Clausen [7] has proposed the entropically damped artificial compressibility (EDAC) method which is applicable to both steady and unsteady flows without the need for dual-time stepping. The EDAC scheme was successfully tested with both a finite-difference MacCormack's method for the two-dimensional lid driven cavity and periodic double shear layer problem and a finite-element method for flow over a square cylinder, with scaling studies on the latter to large numbers of processors. In this study, we discretize the EDAC formulation with a new optimized high-order centered finite-difference scheme and an explicit fourth-order Runge–Kutta method. This is combined with an immersed boundary method to efficiently treat complex geometries and a new robust outflow boundary condition to enable higher Reynolds number simulations on truncated domains. Validation studies for the Taylor–Green Vortex problem and the lid driven cavity problem in both 2D and 3D are presented. An eddy viscosity subgrid-scale model is used to enable large eddy simulations for the 3D cases. Finally, an application to flow over a sphere is presented to highlight the boundary condition and performance comparisons to a traditional incompressible Navier–Stokes solver is shown for the 3D lid driven cavity. Overall, the combined EDAC formulation and discretization is shown to be both effective and affordable.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Fluid flows of practical interest span the low Mach number incompressible regime to the high-speed compressible flow regime. For the latter, the fully compressible Navier–Stokes equations are the only option for prediction. For low-speed flows, numerical stiffness due to disparities between the relatively fast acoustic speed and the slow flow speed often results in numerical stability problems, small time steps, and inefficient calculations. Traditionally, the most common recourse is to discretize and solve

the incompressible form of the Navier–Stokes equations using a predictor-corrector approach together with a Poisson equation for a pressure-like variable. The extra cost of solving the Poisson equation at each time step is somewhat offset by the larger time step that can be employed. Optimizing the efficiency of the Poisson solver is crucial to ensure the best performance of the solver. This is because scalability of the algorithm on large numbers of processors essentially depends on the parallel scalability of the method used to solve the system of linear equations. Some applications still require the use of a compressible solver, even though the flow is mostly low speed. This happens when aeroacoustics is of interest, or for example in high angle of attack low speed aerodynamics where localized shocks can occur. In this case, preconditioning methods can be used [32]. The goal here is to improve the con-

* Corresponding author.

E-mail address: delorme.yann@gmail.com (Y.T. Delorme).

dition number of the system of equations before starting the computation, allowing a relaxation of the time step constraint and thus improving the efficiency of the overall solver.

In 1967, Chorin [6] proposed a new method called the Artificial Compressibility Method (ACM) to solve low speed flows. This method is based on a modified version of the incompressible Navier–Stokes equations where a time derivative of the pressure is introduced into the continuity equation, hence obviating the need for a Poisson solver. Upon convergence of the solution to steady state, the time derivative of the pressure vanishes and the solution satisfies the original incompressible Navier–Stokes equations. As a result of this formulation, this approach was at first limited to steady flows only. In 1977, Steger et al. [31] employed this approach to efficiently simulate 2D and 3D vortex wake flows. In 1978, Hafez et al. [16] demonstrated good performance and accuracy for transonic flow over an airfoil using a modified version of the ACM. In 1985, Choi et al. [5] compared ACM and low Mach number approximation to solve steady low speed flows. They showed that in 2D an optimized preconditioning is necessary for the low Mach number formulation to be as efficient as the ACM in terms of convergence rate. In 1988, Hartwich et al. [17,18] used ACM to study 3D sharp edged delta wings at different angle of attacks. The algorithm was implemented using high-order flux reconstruction schemes that allowed for accurate predictions of the performance of the airfoil when compared to experimental data. Similarly, Kwak et al. [20] studied 3D space shuttle engine power head using ACM on a 3D curvi-linear mesh. All the applications cited until now were limited to steady flows.

For application to unsteady flows, a pseudo-time derivative $\partial/\partial\tau$ is added to the continuity and momentum equations. Hence, there are now two time derivatives in the momentum equation: the actual time and the pseudo-time. For each physical time step, the solution is advanced in pseudo-time until convergence to the steady-state, and hence incompressible solution. This results in an iterative process for each time step until the pseudo-time derivative vanishes. This approach was used successfully for many applications over the past 20 years [4,26,28]. In 2006, Madsen et al. [23] discussed the efficiency and accuracy of this approach. Indeed, an artificial parameter β was introduced in the equations, controlling the rate of convergence of the solution to steady state. It was then possible to converge faster to the incompressible solution, but not without affecting the accuracy of the overall solution. It is then very important to select the value of β carefully to obtain the best efficiency while maintaining a desired level of accuracy. This approach is still widely used for simulations of unsteady 3D flows [33].

Recently, Clausen [7] proposed an alternative approach to solve the ACM equations for unsteady flows without using a dual time stepping approach. This method is called Entropically Damped form of the Artificial Compressibility (EDAC) method and is derived directly from the compressible Navier–Stokes equations and uses thermodynamic relationships to obtain an equation for pressure. It assumes an isentropic behavior of the flow and the pressure equation is obtained by minimizing density fluctuations. In [7], the EDAC scheme was successfully tested using both a finite-difference MacCormack's method for the two-dimensional steady lid driven cavity and unsteady periodic double shear layer problem, and a finite-element method for flow over a square cylinder. Scaling of the finite-element method based solver on a large number of processors was also shown to be very good due to the absence of an iterative process, even for unsteady flows.

In the present paper, the EDAC formulation is discretized by combining a new family of optimized high-order finite-difference schemes for spatial discretization on structured Cartesian grids [22] with the classic fourth-order Runge–Kutta scheme for explicit time stepping. For non-Cartesian geometries a mirroring immersed

boundary method (IBM) [25] is utilized and for turbulent flows an eddy viscosity based subgrid-scale model (SGS) is used to enable large eddy simulations (LES). In addition, to enable simulations on highly truncated domains, a new robust outflow boundary condition for incompressible flows [10] is implemented and tested. All these numerical methods were chosen to obtain a simple and efficient incompressible Navier–Stokes solver. Indeed, the EDAC formulation allows for low-speed flow without the need of a staggered grid or the use of an iterative method to solve the Poisson equation for a pressure-like variable. This results in a simpler implementation of the algorithm as well as a highly scalable solver on a large number of processors. The use of IBM enable simulations of complex geometries while maintaining a simple structured Cartesian mesh needed to implement high-order finite difference schemes. Similarly, the boundary condition by Dong et al. [10] enables shorter computational domains, hence reducing the number of grid points needed for the simulations. In what follows, the mathematical formulation and numerical discretization of the governing equations is presented in detail. The solver is then used for classical 2D studies of the Taylor–Green vortex (TGV), the periodic double-shear layer, and lid-driven cavity problems. This is followed by extension to the 3D TGV problem and the 3D lid-driven cavity problem to test the LES model. Finally, an application for 3D flow over a sphere at different Reynolds numbers is presented and efficiency of the solver is assessed by comparisons to results from a traditional incompressible Navier–Stokes solver.

2. Methods

2.1. Governing equations

The non-dimensional form of the entropically damped artificial compressibility formulation of Clausen [7] is:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial \tau_{ij}}{\partial x_j} \quad (1)$$

where u_i is the i th component of the velocity vector, x_i is the i th spatial coordinate, p is the pressure, Re is the Reynolds number, and τ_{ij} is the viscous stress tensor. Typical projection methods require solving a Poisson equation for the pressure. EDAC employs an entropy balance to close the system of equations. Assuming an isentropic behavior and minimizing density fluctuations, the EDAC equation for pressure is:

$$\frac{\partial p}{\partial t} + u_i \frac{\partial p}{\partial x_i} = -\frac{1}{M^2} \frac{\partial u_i}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 p}{\partial x_i \partial x_i} \quad (2)$$

where M is the Mach number. More details and discussions about the mathematical derivation of Eq. (2) can be found in [7]. In all the results presented in this study, the divergence of the velocity was no larger than 0.2 locally, with a spatially averaged value lower than 10^{-4} .

2.2. Turbulence modeling

For 3D studies of transitional or turbulent flows, the low-pass spatially-filtered EDAC equations are solved enabling LES. Hence, only the large-scale flow features are resolved on the grid and effects of the unresolved small-scales on the large-scales are modeled using an eddy viscosity SGS model. The closed filtered EDAC equations are:

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{1}{Re} \frac{\partial \hat{\tau}_{ij}}{\partial x_j} \quad (3)$$

where $\hat{\tau}_{ij}$ contains the filtered viscous stresses $\tilde{\tau}_{ij}$ and the SGS tensor $(\tau_{ij})_{SGS}$:

$$\hat{\tau}_{ij} = \tilde{\tau}_{ij} - (\tau_{ij})_{SGS} \quad (4)$$

Employing an eddy viscosity based SGS model yields [27]

$$(\tau_{ij})_{SGS} = -2\nu_T S_{ij} \quad (5)$$

where

$$S_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \quad (6)$$

The eddy viscosity ν_T is specified in terms of gradients of the filtered velocity field according to the Vreman model [30]. This model is easy to implement, efficient to use, and applicable to fully inhomogeneous turbulent flows [1,8,9,30]. Following Clausen, the convective derivative in the pressure equation is neglected as small and the filtered pressure equation is simply:

$$\frac{\partial \tilde{p}}{\partial t} = -\frac{1}{M^2} \frac{\partial \tilde{u}_i}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 \tilde{p}}{\partial x_i \partial x_i} \quad (7)$$

2.3. Numerical methods

The filtered EDAC Eqs. (3) and (7) are first written in the following conservative vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{S} \quad (8)$$

where

$$\mathbf{U} = \begin{bmatrix} \tilde{p} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \quad (9)$$

$$\mathbf{F} = \begin{bmatrix} -\frac{1}{Re} \frac{\partial \tilde{p}}{\partial x} \\ \tilde{u}^2 + \tilde{p} - \hat{\tau}_{xx} \\ \tilde{u}\tilde{v} - \hat{\tau}_{xy} \\ \tilde{u}\tilde{w} - \hat{\tau}_{xz} \end{bmatrix} \quad (10)$$

$$\mathbf{G} = \begin{bmatrix} -\frac{1}{Re} \frac{\partial \tilde{p}}{\partial y} \\ \tilde{u}\tilde{v} - \hat{\tau}_{yx} \\ \tilde{v}^2 + \tilde{p} - \hat{\tau}_{yy} \\ \tilde{v}\tilde{w} - \hat{\tau}_{yz} \end{bmatrix} \quad (11)$$

$$\mathbf{H} = \begin{bmatrix} -\frac{1}{Re} \frac{\partial \tilde{p}}{\partial z} \\ \tilde{u}\tilde{w} - \hat{\tau}_{zx} \\ \tilde{v}\tilde{w} - \hat{\tau}_{zy} \\ \tilde{w}^2 + \tilde{p} - \hat{\tau}_{zz} \end{bmatrix} \quad (12)$$

$$\mathbf{S} = \begin{bmatrix} \tilde{p} - \frac{1}{M^2} \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{v}}{\partial y} + \frac{\partial \tilde{w}}{\partial z} \right) \\ \tilde{u} \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{v}}{\partial y} + \frac{\partial \tilde{w}}{\partial z} \right) \\ \tilde{v} \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{v}}{\partial y} + \frac{\partial \tilde{w}}{\partial z} \right) \\ \tilde{w} \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{v}}{\partial y} + \frac{\partial \tilde{w}}{\partial z} \right) \end{bmatrix} \quad (13)$$

The semi-discrete form of Eq. (8) is temporally integrated using the classic 4th-order explicit Runge–Kutta scheme. In order to

enable accurate resolution of high-frequencies on relatively coarse grids, while maintaining low numerical dissipation, the newly developed optimized high-order centered finite-difference schemes of Linders et al. [22] are used. According to this scheme, first derivatives are discretized as:

$$\left(\frac{\partial u}{\partial x} \right)_i + \mathcal{O}(\Delta x^{2p}) = \frac{1}{\Delta x} \sum_{k=1}^{p+n} a_k (u_{i+k} - u_{i-k}) \quad (14)$$

In all the results in this paper, the 4th-order accurate version of the family of schemes is used where $p = 2$, $n = 4$ and the scheme coefficients are:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 0.896607046646854 \\ -0.320910877852970 \\ 0.119465303396051 \\ -0.037162191039544 \\ 0.008242459236975 \\ -0.000957455525961 \end{bmatrix} \quad (15)$$

More details, discussions and proofs regarding these new finite-difference schemes can be found in [22]. In order to reduce high-frequency spurious oscillations, a 4th-order spatial filter [2] is applied to the primitive variables after each time step as shown:

$$(\phi_{filt})_i = \phi_i + 0.1 \left(\sum_{k=1}^6 b_k (\phi_{i+k} + \phi_{i-k}) + b_0 \phi_i \right) \quad (16)$$

with

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} 0.190899511506 \\ -0.171503832236 \\ 0.123632891797 \\ -0.069975429105 \\ 0.029662754736 \\ -0.008520738659 \\ 0.001254597714 \end{bmatrix} \quad (17)$$

This filter removes high frequency oscillations but leaves the lower frequencies intact. It is 13 points wide, which is the same as the stencil used for the first derivative. Further, it is optimized to filter wave numbers larger than $\pi/2$, which is desired since the scheme used to compute the first derivative accurately resolves wave numbers up to precisely $\pi/2$ but no further. Fig. 1 shows the performance of the optimized centered scheme with and without filtering, and compared to a regular 4th order centered scheme. For this test, the advection equation is solved:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \quad (18)$$

on a $[0; 5]$ domain with a Gaussian pulse initial condition:

$$u(x, t = 0) = \exp(-3200 \times (x - 0.5)^2) \quad (19)$$

We can see that the 4th order scheme (black) shows high oscillations near the pulse and downstream as well. When using 4th order filtering (green), the oscillations are reduced downstream of the pulse, but are still very present near the pulse. On the other hand, the optimized centered scheme (blue) shows a much better agreement with the exact solution. Only some oscillations are seen downstream of the pulse. After applying filtering (orange), the oscillations downstream of the pulse disappear. These 1D results motivate the use of the optimized centered scheme with filtering for the rest of the current study. Near the boundaries of the computational domain as well as in the vicinity of the immersed body for the fluxes, a fourth order one sided finite difference scheme is used to avoid use of extra layers of ghost points. Finally, the viscous terms are computed by applying twice the centered scheme for first order derivatives.

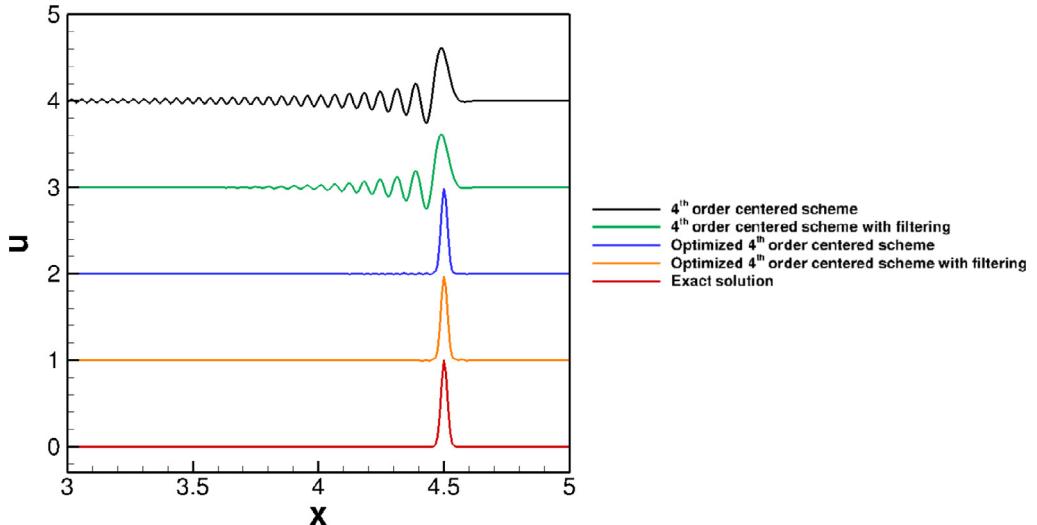


Fig. 1. Comparison of centered schemes for the Gaussian pulse problem ($t = 4$).

2.4. Immersed boundary method

In order to efficiently simulate flows over or through non-Cartesian geometries a mirroring immersed boundary method (IBM) is used [25]. This IBM has been widely used and validated in our previous studies for a number of internal and external flows [1,8,9].

2.5. Boundary conditions

To further enhance numerical efficiency, in particular for external flows on highly truncated computational domains, a newly developed robust energy-stable outflow boundary condition is implemented [10]. Traditionally the use of homogeneous Neumann boundary conditions for velocity and pressure require the computational domain length to be linearly proportional to the Reynolds number for stability. This results in a more expensive simulation since the flow near the outflow is often damped or discarded. Many previous studies have attempted to address this issue [29]. Following the approach of Dong et al. [10], the boundary condition for the pressure at the outlet plane is:

$$-\tilde{p}\mathbf{n} + \frac{1}{Re}\mathbf{n} \cdot \nabla \tilde{\mathbf{u}} - \left(\frac{1}{2}|\tilde{\mathbf{u}}|^2 S_0(\mathbf{n} \cdot \tilde{\mathbf{u}}) \right) \mathbf{n} = 0 \quad (20)$$

with

$$S_0(\mathbf{n} \cdot \tilde{\mathbf{u}}) = \frac{1}{2} \left(1 - \tanh\left(\frac{\mathbf{n} \cdot \tilde{\mathbf{u}}}{U_0 \delta}\right) \right) \quad (21)$$

where \mathbf{n} is the outward pointing normal unit vector, U_0 is the characteristic velocity scale, and δ is a small enough positive constant. In the current study, we used $U_0 = 1$ and $\delta = 1/20$. This formulation compensates for any influx of kinetic energy through the outlet plane. The term $(\frac{1}{2}|\tilde{\mathbf{u}}|^2 S_0(\mathbf{n} \cdot \tilde{\mathbf{u}}))\mathbf{n}$ represents the kinetic energy entering the domain and it is balanced by the effective stress $-\tilde{p}\mathbf{n} + \frac{1}{Re}\mathbf{n} \cdot \nabla \tilde{\mathbf{u}}$ on the outlet plane. Similarly an equation for the velocity gradient at the outlet plane can be derived:

$$\mathbf{n} \cdot \nabla \mathbf{u} = Re \left(p\mathbf{n} + \frac{1}{2}|\mathbf{u}|^2 S_0(\mathbf{n} \cdot \mathbf{u})\mathbf{n} - \frac{1}{Re}(\nabla \cdot \mathbf{u})\mathbf{n} \right) \quad (22)$$

More details about the derivation of the formulation and validation results can be found in [10].

3. Results

3.1. 2D Taylor–Green Vortex (2D TGV)

In order to estimate the order of accuracy of the hybrid scheme, predictions of the 2D TGV are compared to the analytic solution. The initial condition and exact solution for u , v and p are:

$$\begin{aligned} u(x, y, t) &= 1 - \cos(x - t)\sin(y - t)e^{-2t/Re} \\ v(x, y, t) &= 1 + \sin(x - t)\cos(y - t)e^{-2t/Re} \\ p(x, y, t) &= -\frac{1}{4}(\cos(2(x - t)) + \cos(2(y - t)))e^{-4t/Re} \end{aligned} \quad (23)$$

The computational domain is a $[0, 2\pi]^2$ periodic square and $Re = 100$. To estimate the order of accuracy of the solver, comparisons between the numerical and analytical solutions were performed after 10 time steps by computing the L_2 norm of the error as follow:

$$L_2 = \sqrt{\frac{1}{N_x \times N_y} \sum_{j=1}^{Ny} \sum_{i=1}^{Nx} ((u_{ij})_{analytical} - (u_{ij}))^2} \quad (24)$$

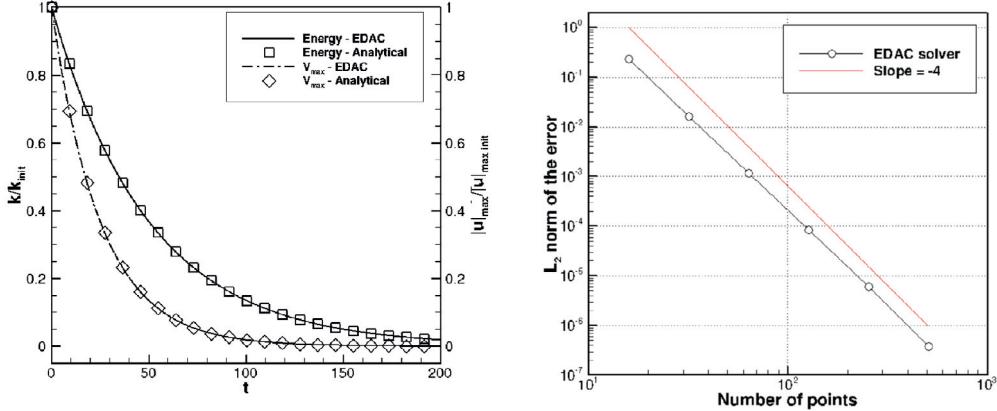
The kinetic energy decay as a function of time is shown in Fig. 2. The agreement between the analytical and numerical results is excellent. The L_2 norm of the error for different grid sizes is shown in Fig. 2-b, where the slope of the curve is very close to (-4) , confirming the global spatial order of accuracy of the solution.

3.2. 2D Double Shear Layer (2D DSL)

In order to assess the role of numerical dissipation in the solver and its ability to resolve sharp spatial gradients at low resolutions, the case of the 2D double shear layer (DSL) is considered. The 2D DSL initial conditions are:

$$\begin{aligned} u(x, y) &= \tanh(80 \times (y - 0.25)) & (y \leq 0.5) \\ u(x, y) &= \tanh(80 \times (0.75 - y)) & (y > 0.5) \\ v(x, y) &= 0.05 \times \sin(2\pi(x + 0.25)) \end{aligned} \quad (25)$$

The computational domain is a $[0, 1]^2$ square periodic domain and $Re = 10000$. Typically, for this case, under-resolved or highly dissipative simulations produce spurious secondary braid vortices which result in an early breakdown of the shear layer [24]. Contour plots of vorticity are shown in Fig. 3 at $t = 1$ for two different



(a) Normalized kinetic energy versus time.
(b) L_2 norm of the error between numerical and analytical solutions. The 4th order accuracy can be clearly seen.

Fig. 2. Validation and accuracy study of the developed solver.

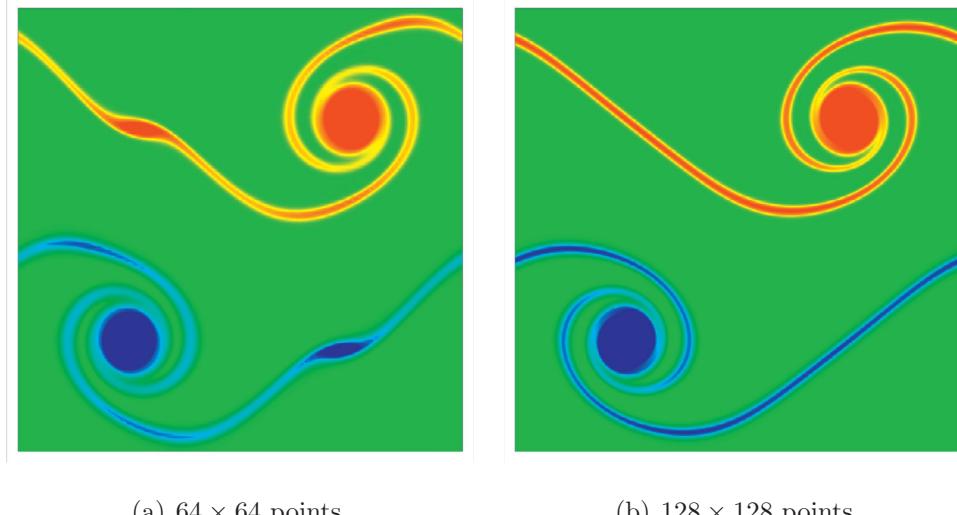


Fig. 3. Contour plots of vorticity at $t = 1$ for the 2D double shear layer.

grid resolutions. Secondary vortices in the braid region are seen to form in Fig. 3-a corresponding to the 64×64 grid due to lack of resolution. In contrast, Fig. 3-b for the 128×128 grid shows no evidence of spurious vortices. Hence, the scheme is able to capture these strong velocity gradients on this relatively coarse grid compared to previously published studies which required finer grids [7,19,24].

3.3. 2D Lid-Driven Cavity (2D LDC)

In order to consider wall-bounded flows, the classic 2D lid-driven cavity problem is studied here for a range of different Reynolds numbers Re ($Re = 400, 1000, 3200, 5000, 10000$). For all cases the computational domain is a unit square and a 256×256 uniformly spaced grid is used. The simulations are run until steady state is achieved. Steady-state streamlines are shown in Fig. 4 for all cases and demonstrate that the solver captures both primary and secondary corner vortices well. As the Reynolds number keeps

increasing, the flow becomes more chaotic and smaller pockets of recirculating flow can be seen (Fig. 4-e). The results are quantitatively compared with the classic numerical data set of Gharib et al. [15] as shown in Fig. 5. Fig. 5-a and b show perfect agreement for low Reynolds numbers. Fig. 5-c-e show that the flow becomes more and more chaotic resulting in sharp velocity gradients near the walls and yet excellent agreement is maintained.

3.4. 3D Taylor Green Vortex (3D TGV)

For the 3D TGV case, the computational domain is a $[-\pi, \pi]^3$ periodic cube with initial conditions:

$$\begin{aligned} u(x, y, z) &= \sin(x)\cos(y)\cos(z) \\ v(x, y, z) &= \cos(x)\sin(y)\cos(z) \\ w(x, y, z) &= 0 \\ p(x, y, z) &= p_0 + \frac{1}{16}(\cos(2x)\cos(2y)\cos(2z) + 2) \end{aligned} \quad (26)$$

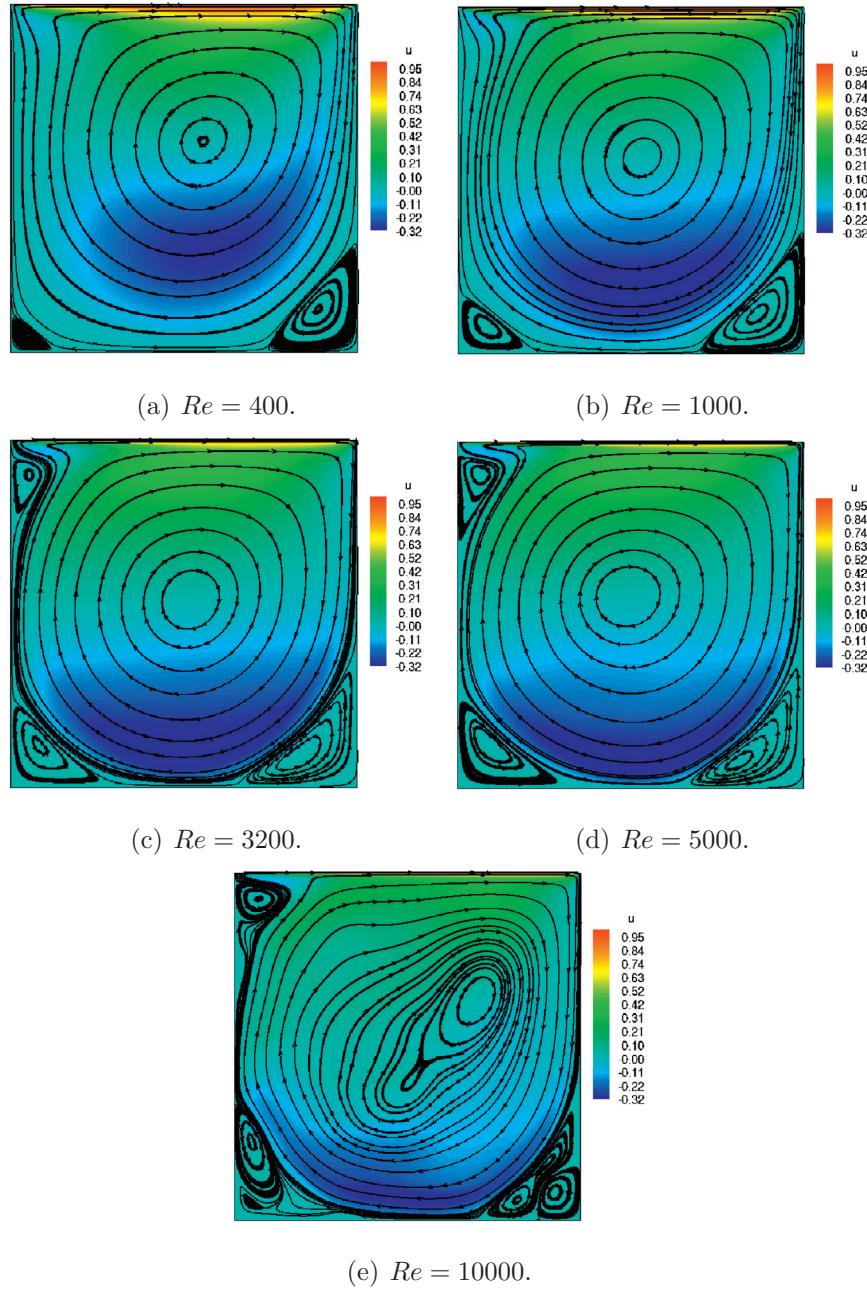


Fig. 4. 2D Lid-Driven cavity. u -velocity contour with selected streamlines inside the cavity.

For adequate resolution and numerical accuracy, these initial vortices will interact and breakdown with time to form a turbulent flow. Comparisons can be made to direct numerical simulation (DNS) data from previous published studies [3]. Fig. 6-a shows the initial vortices inside the cubical computational domain. The simulations are run on a 128^3 grid with $Re = 1600$ and the Vreman SGS model is used enabling LES. As the simulation progresses, the vortices interact with each other and eventually break down into smaller scales resulting in a near-homogeneous turbulent state. Fig. 6-b shows smaller-scale vortices at $t = 10$. The kinetic energy decay rate from the present EDAC-based LES (Fig. 7-a and b) compares well to the DNS data of Brachet [3]. Specifically, on this relatively coarse grid, the results do not show an over-prediction of the decay, confirming the low level of numerical dissipation of the solver.

3.5. 3D Lid-Driven Cavity (3D LDC)

Here the 3D LDC case at $Re = 12000$ is considered. A unit cube computational domain is discretized with an 80^3 grid. The Vreman SGS model is used enabling LES. No-slip boundary conditions are used on all but the top face where the wall velocity is specified as [21]:

$$u(x, 1, z) = (1 - (x - 1)^{18})^2 (1 - (z - 1)^{18})^2 \quad (27)$$

The LES results are compared to DNS data from Leriche et al. [21] and are used to assess accuracy of the numerical scheme, as well as the SGS turbulence model. Fig. 8-a depicts a snapshot of vortical structures within the cavity at $t = 100$. The solver is clearly capable of capturing a wide range of vortical structures with high vorticity magnitude near the top wall and near the left surface where the impact of the moving wall is the strongest. Fig. 8-b

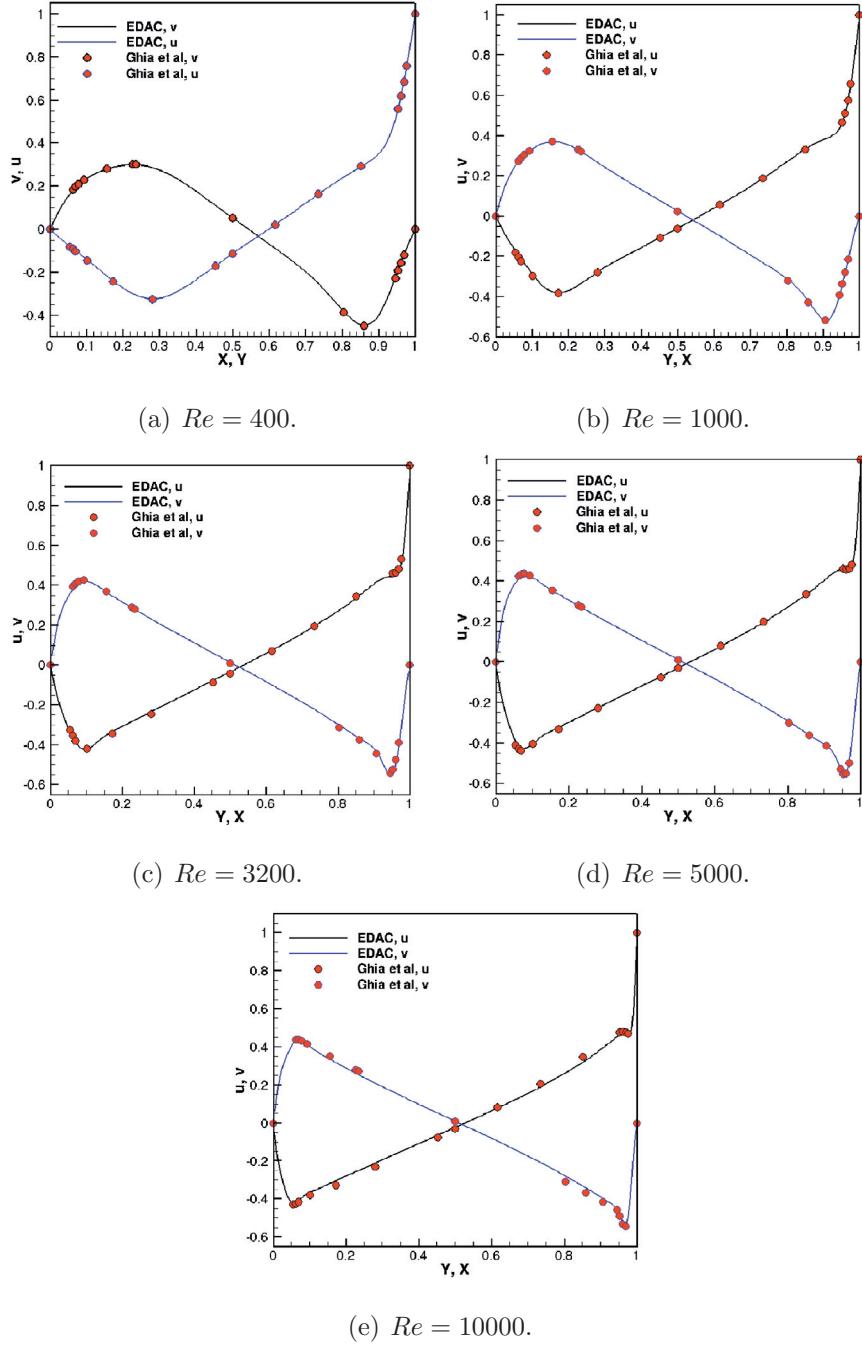


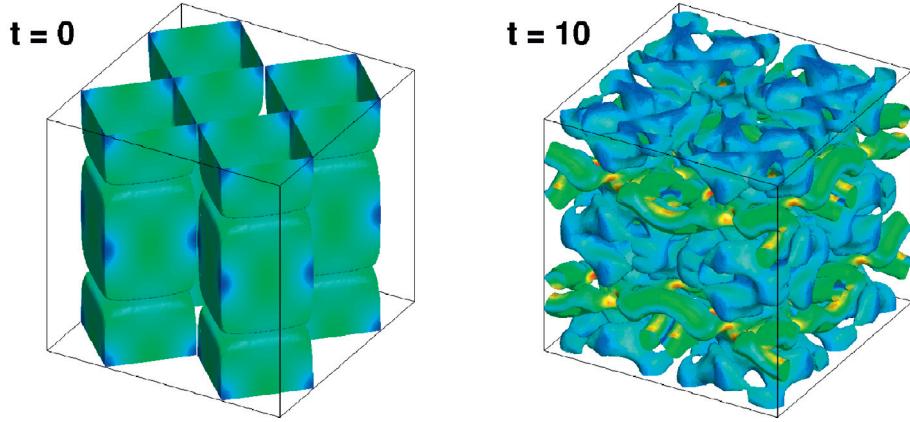
Fig. 5. 2D Lid-Driven cavity. Comparisons between current results and DNS data from Ghia et al.

shows very good agreement for mean velocity profiles between the LES (averaged over 1000 time units) and the DNS data.

3.6. 2D flow over a cylinder

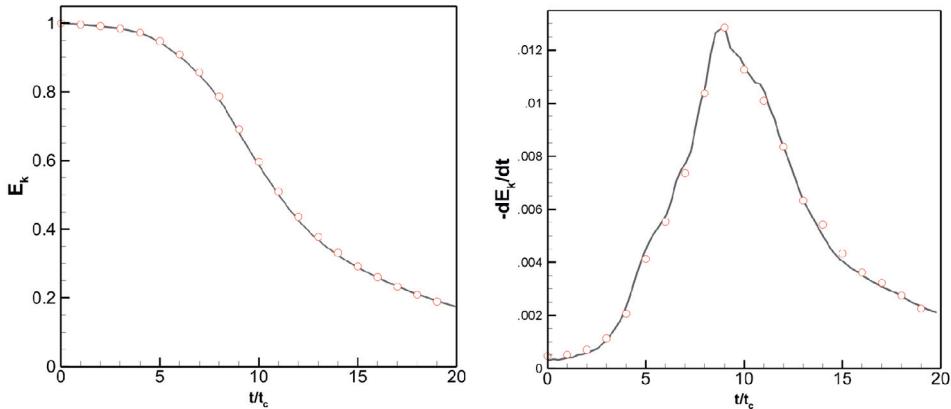
To test the solver using the IBM and the energy-stable outflow boundary condition [10], 2D flow over a cylinder at $Re = 1500$ on a severely truncated domain is first considered. The computational domain is $[-2D, 2.5D] \times [-2.5D, 2.5D]$ and is discretized using a uniform 90×128 grid. The cylinder is placed inside a channel, resulting in no slip boundary condition on the top and bottom boundaries. Instantaneous vorticity contour plots at several times are shown in Fig. 9. The results shown on the left column were obtained using a standard homogeneous Neumann boundary con-

dition at the outlet whereas the results on the right column were obtained using the energy-stable outflow boundary condition from Dong et al. At the beginning of the simulations (Fig. 9-a-d), the two methods show similar solutions, the outlet boundary not affecting the upstream solution at this point since no vortex has crossed the outlet plane yet. At a later time (Fig. 9-e and f), the two solutions are very different. The results obtained using the energy-stable outflow boundary condition show a typical vortex shedding without any re-circulation or inflow coming from the outlet plane. In contrast, the case with the Neumann boundary condition, clearly shows some back-flow at the outlet, which affects the upstream flow and considerably alters the solution. This phenomenon only gets worse with time resulting in additional back-flow at the outlet (Fig. 9-g) and eventual numerical instability of the solver. This is



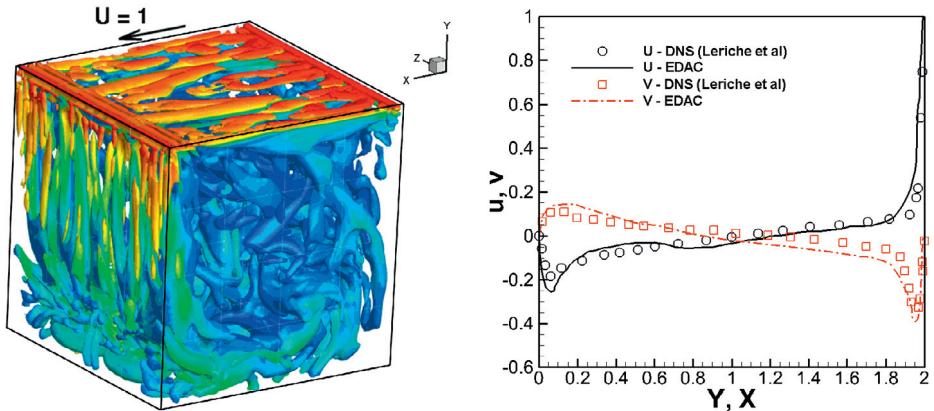
(a) Iso-surface of λ_2 colored by vorticity magnitude ($t = 0$).
(b) Iso-surface of λ_2 colored by vorticity magnitude ($t = 10$).

Fig. 6. 3D Taylor-Green vortices. Vortical structures.



(a) Decay of kinetic energy compared to DNS data of Brachet.
(b) Kinetic energy decay rate compared to DNS data of Brachet

Fig. 7. 3D Taylor-Green vortices. Comparison of kinetic energy decay predicted by the current EDAC solver (black line) with DNS data of Brachet (Red dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Iso-surface of λ_2 colored by vorticity magnitude ($t = 100$).
(b) Comparison between current predictions and DNS data.

Fig. 8. 3D Lid-Driven cavity. Vortical structures and comparisons with DNS data (Leriche et al. [21]).

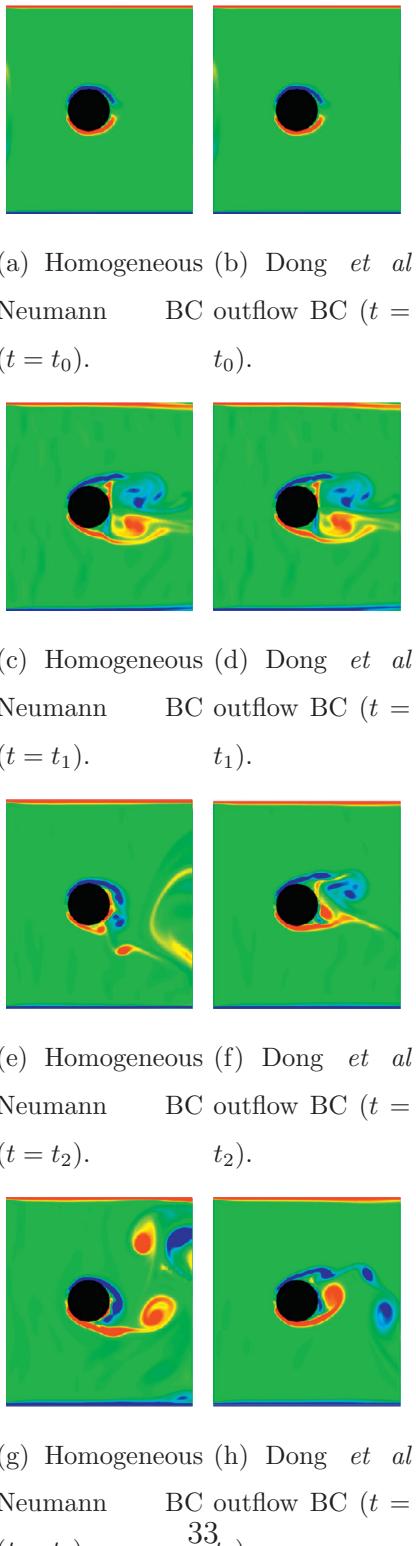
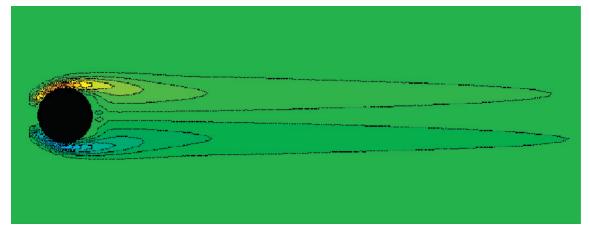
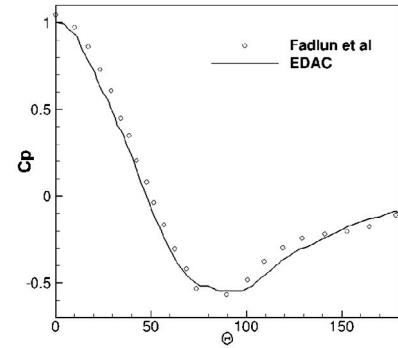


Fig. 9. 2D cylinder on a truncated domain. Out of plane vorticity.

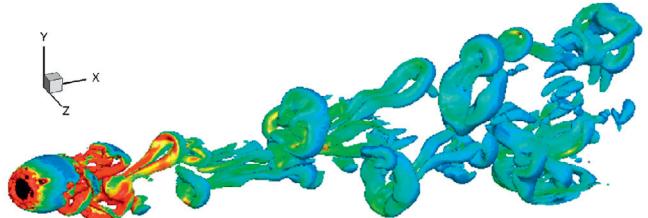
not observed with the other boundary condition (Fig. 9-h): the vortices that are generated by the presence of the cylinder are leaving the computational domain without affecting the rest of the flow, keeping the solution stable. These two simulations validate the implementation of the energy stable outflow boundary condition, and show its positive effect on the stability of the solution. This feature



(a) Out of plane vorticity at a centerplane of the domain ($y=0$). $Re = 200$.



(b) Comparison between current predictions and DNS data from Fadlun *et al.* $Re = 200$.



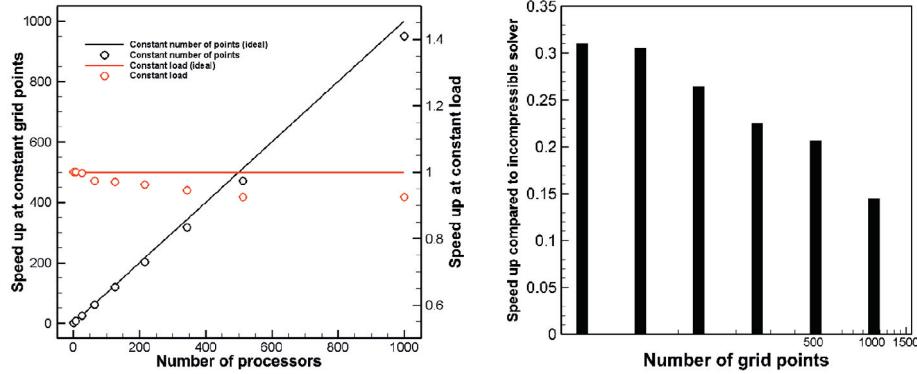
(c) Iso-surface of λ_2 colored by vorticity magnitude. $Re = 1500$.

Fig. 10. 3D flow over sphere (laminar and turbulent).

is critical when trying to improve the computational efficiency of the solver.

3.7. 3D flow over a sphere

Finally, 3D flow over a sphere at $Re = 200$ and at $Re = 1500$ are considered. For both cases, the computational domain size is $[-4D, 25D] \times [-3D, 3D] \times [-3D, 3D]$ and is discretized using a $200 \times 100 \times 100$ uniform grid. The results are shown in Fig. 10. Fig. 10-a shows the out-of-plane vorticity along the $y = 0$ slice for the $Re = 200$ case where the flow is laminar and steady and the vorticity shows top-bottom symmetry in this plane. Fig. 10-b shows very good agreement between the current numerical predictions for C_p at the surface of the surface with the DNS data of Fadlun et al. [11]. This validates the implementation of the IBM in the context of the EDAC formulation. Fig. 10-c shows an iso-surface of λ_2 colored by vorticity magnitude for the sphere case at $Re = 1500$. The solver captures the complex wake structure downstream of the sphere including hairpin-like structures near the sphere which break down further downstream.



(a) Scaling of the solver with a constant number of grid points or a constant load. (b) Comparison of the efficiency between the developed EDAC solver and a traditional predictor / corrector approach.

Fig. 11. Performances of the developed solver.

3.8. Algorithm efficiency assessment

In order to assess the computational efficiency of the overall solver, several tests were run. First, the scalability of the solver was studied for the 3D LDC problem. It is parallelized using Message Passing Interface (MPI). Fig. 11-a shows the scalability for two different tests: in black we can see the scalability of the solver when the number of total grid points is maintained constant (51.2 millions) and the number of processors varies. It can be seen that the solver scales linearly with performances close to the ideal scalability. In red we can see that scalability of the solver when the load of each processor is maintained constant. Once again it can be seen that the performance of the solver stays very good, even as the number of processors increases up to 1000.

Now, comparisons are also made to a traditional predictor/corrector incompressible Navier-Stokes code. The exact same numerical methods are used in this new solver as in the EDAC solver. Both solvers are parallelized using Message Passing Interface (MPI) and the Poisson equation in the predictor/corrector incompressible code is solved using the Hypre library [12–14]. Both solvers showed linear scaling on up to 800 processors. The efficiency comparison between the two solvers is carried out for the 3D LDC problem on 512 processors. The simulation speed up is defined as:

$$\text{speedup} = \frac{(t_{1s})_{\text{pred/correc}} - (t_{1s})_{\text{EDAC}}}{(t_{1s})_{\text{pred/correc}}} \quad (28)$$

where $(t_{1s})_{\text{EDAC}}$ is the computational time to reach 1s of simulation using the EDAC code and $(t_{1s})_{\text{pred/correc}}$ is the computational time to reach 1s of simulation using the predictor/corrector code. The results are shown in Fig. 11-b where speed up versus grid size is compared. It is clear that on a 64^3 grid, the EDAC code is about 30% faster than the predictor/corrector solver. As the grid size increases, the speed up of the solver deceases slightly to around 27% for the 128^3 grid, 23% for the 256^3 grid, and 15% for the 1024^3 . The improved performance of the EDAC code is primarily due to avoidance of the Poisson solver, which makes the computation time per time step much smaller even though the global time step has to be slightly reduced for stability reasons. As the number of grid points increases, the time step used in the EDAC solver had to be reduced faster than for the incompressible solver, which affects the overall

speed up. But the drop in efficiency is only seen for very large grid (around a billion point mesh), making this approach suitable for practical applications. Moreover, the advantage of the EDAC formulation for simulations of incompressible flows is the fact that no elliptic equations need to be solved, making parallel implementation easier. The use of hybrid grids such as Adaptive Mesh Refinement (AMR), curvilinear meshes or multiple resolution by block is also facilitated. In conclusion, the EDAC formulation when combined with numerical methods selected for this study demonstrate a relatively simple, efficient, and accurate incompressible flow solver.

4. Conclusion

An efficient algorithm was implemented for the simulation of laminar and turbulent incompressible internal and external flows. An inherently unsteady and stable version of the ACM, from Clausen [7] was used. This avoids the need for a Poisson solver or dual-time stepping resulting in a very efficient numerical scheme. Spatial discretization on structured Cartesian grids was achieved using recent high-order centered finite-difference schemes optimized for simulating flows featuring a wide range of wave numbers. For non-Cartesian geometries a mirroring immersed boundary method was used. For turbulent flows, an eddy viscosity based subgrid-scale modeled was used to enable large eddy simulations. For external flows, a recent energy stable outflow boundary condition was used enabling stable high Reynolds number simulations on highly truncated domains. Both 2D and 3D periodic, internal, and external flows were simulated and compared to previously published data for validation. The results show excellent high-resolution capability on relatively coarse grids.

Acknowledgment

The authors would like to acknowledge Dr. Jonathan Clausen for the discussions related to the EDAC method. The authors would also like to acknowledge financial support for this work from the Rosenblatt Chair within the faculty of Mechanical Engineering and Zeff Fellowship Trust.

References

- [1] Anupindi K, Delorme Y, Frankel S, Shetty D. A novel multiblock immersed boundary method for large eddy simulation of complex arterial hemodynamics. *J Comput Phys* 2013;254:200–18.
- [2] Bogeay C, Baily C. A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *J Comput Phys* 2004;194:194–214.
- [3] Brachet M. Direct numerical simulation of three dimensional turbulence in the taylor green vortex. *Fluid Dyn Res* 1991;8:1–9.
- [4] Chen K, Pletcher R. Simulation of three dimensional liquid sloshing flows using a strongly implicit calculation procedure. *AIAA J* 1993;31:901–10.
- [5] Choi D, Merkle C. Application of time iterative schemes to incompressible flow. *AIAA J* 1985;23:1518–24.
- [6] Chorin A. A numerical method for solving incompressible viscous flow problems. *J Comput Phys* 1967;135:118–25.
- [7] Clausen J. Entropically damped form of artificial compressibility for explicit simulation of incompressible flow. *Phys Rev E* 2013;87:13309–21.
- [8] Delorme Y, Anupindi K, Kerlo A, Shetty D, Rodefeld M, Chen J, et al. Large eddy simulation of powered fontan hemodynamics. *J Biomech* 2013a;46:408–22.
- [9] Delorme Y, Anupindi K, Frankel S. Large eddy simulation of FDA's idealized medical device. *Cardiovasc Eng Technol* 2013b;4:4.
- [10] Dong S, Karniadakis G, Chryssostomidis C. A robust and accurate outflow boundary condition for incompressible flow simulations on several truncated unbounded domains. *J Comput Phys* 2014;261:83–105.
- [11] Fadlun E, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed boundary finite difference methods for three dimensional complex flow simulations. *J Comput Phys* 2000;161:35–60.
- [12] Falgout R, Jones J. Multigrid on massively parallel architectures.
- [13] Falgout R, Yang. Hypre: a library of high performance preconditioners. 2002. <http://www.actsc.nerc.gov/hypre/>.
- [14] Ghia U, Ghia K, Shin C. High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J Comput Phys* 1982;48:387–411.
- [16] Hafez M, South J, Murman E. Artificial compressibility methods for numerical solutions of transonic full potential equation. *AIAA J*. 17:838–844.
- [17] Hartwich P, Hsu C. High resolution upwind schemes for the three dimensional incompressible navier stokes equations. *AIAA J* 1988;26:1321–8.
- [18] Hartwich P, Hsu C, Liu C. Vectorizable implicit algorithms for the flux difference split three dimensional navier stokes equations. *J Fluids Eng* 1988;110:297–305.
- [19] Hashimoto T, Tanno I, Yasuda T, Tanaka Y, Morinishi K, Satofuka N. Higher order numerical simulation of unsteady viscous incompressible flows using kinetically reduced local navier stokes equations on a GPU. *Comput Fluids* 2015;110:108–13.
- [20] Kwak D, Chang J, Shanks P, Chakravarthy S. A three dimensional incompressible navier stokes flow solver using primitive variables. *AIAA J* 1986;24:390–6.
- [21] Leriche E, Gavrilakis S. Direct numerical simulation of the flow in a lid-driven cubical cavity. *Phys Fluids* 2000;12:1363–76.
- [22] Linders V, Nordstrom J. Uniformly best wavenumber approximations by spatial central difference operators. *J Comp Phys* 2015;300:695–709.
- [23] Madsen P, Shaffer H. A discussion of artificial compressibility. *Coastal Eng* 2006;53:93–8.
- [24] Minion M, Brown D. Performance of under-resolved two-dimensional incompressible flow simulations, II. *J Comput Phys* 1997;138:734–65.
- [25] Mark A, van Wachem B. Derivation and validation of a novel implicit second order accurate immersed boundary method. *J Comput Phys* 2008;227:6660–80.
- [26] Pan D, Chakravarthy S. Unified formulation for incompressible flows. 1989. AIAA.
- [27] Pope S. *Turbulent flows*. Cambridge University Press; 2000.
- [28] Rogers S, Kwak D, Kiris C. Numerical solution of the incompressible navier stokes equations for steady state and time dependent problems. 1989. AIAA.
- [29] Tsynkov S. Numerical solution of problems on unbounded domains: a review. *Appl Numer Math* 1998;27:465–532.
- [30] Shetty D, Fisher T, Chunekar A, Frankel S. High order incompressible large eddy simulation of fully inhomogeneous turbulent flows. *J Comput Phys* 2010;229:8802–22.
- [31] Steger J, Kulter P. Implicit finite difference procedures for the computation of vortex wakes. *AIAA J* 1977;15:581–90.
- [32] Turkel E. Review of preconditioning methods for fluid dynamics. *Appl Numer Math* 1993;12:257–84.
- [33] Vrahliotis S, Pappou T, Tsangaris S. Artificial compressibility 3d navier stokes solver for unsteady incompressible flows with hybrid grids. *Eng Appl Comput Fluid Mech* 2012;6:248–70.