

A high-order solver for unsteady incompressible Navier–Stokes equations using the flux reconstruction method on unstructured grids with implicit dual time stepping

Christopher Cox ^{*,1,2}, Chunlei Liang ¹, Michael W. Plesniak ²

Department of Mechanical & Aerospace Engineering, The George Washington University, Washington, DC 20052, USA

ARTICLE INFO

Article history:

Received 12 May 2015

Received in revised form 8 February 2016

Accepted 9 March 2016

Available online 14 March 2016

Keywords:

Navier–Stokes

Artificial compressibility

High-order

Flux reconstruction

Unstructured

Implicit LU-SGS

ABSTRACT

We report development of a high-order compact flux reconstruction method for solving unsteady incompressible flow on unstructured grids with implicit dual time stepping. The method falls under the class of methods now referred to as flux reconstruction/correction procedure via reconstruction. The governing equations employ Chorin's classic artificial compressibility formulation with dual time stepping to solve unsteady flow problems. An implicit non-linear lower-upper symmetric Gauss–Seidel scheme with backward Euler discretization is used to efficiently march the solution in pseudo time, while a second-order backward Euler discretization is used to march in physical time. We verify and validate implementation of the high-order method coupled with our implicit time stepping scheme using both steady and unsteady incompressible flow problems. The current implicit time stepping scheme is proven effective in satisfying the divergence-free constraint on the velocity field in the artificial compressibility formulation within the context of the high-order flux reconstruction method. This compact high-order method is very suitable for parallel computing and can easily be extended to moving and deforming grids.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

In computational fluid dynamics, unstructured high-order methods, i.e. those considered 3rd order and above, are useful for the study of unsteady vortex dominated viscous flows in complex geometries. These methods can provide high accuracy for similar cost as low-order methods [1]. Furthermore, solution acceleration can be achieved with p -adaptivity and p -multigrid methods. However, high-order methods are less robust and more complicated to implement than low-order methods, especially when treating irregular geometries.

Four popular methods have been developed to address the need for high-order accuracy – discontinuous Galerkin (DG), spectral difference (SD), spectral volume (SV) and flux reconstruction/correction procedure via reconstruction (FR/CPR). Discontinuous Galerkin was initially developed for the neutron transport equation by Reed and Hill [2]. The staggered grid spectral method was initially presented by Kopriva [3] and was modified and called spectral difference method by Liu, Vinokur and Wang [4]. The spectral volume method was introduced by Wang, Zhang and Liu [5], where each element is split

* Corresponding author. Tel.: +1 267 567 4375.

E-mail address: ccox@gwmail.gwu.edu (C. Cox).

¹ Computational Aerodynamics & Hydrodynamics Laboratory.

² Biofluid Dynamics Laboratory.

into multiple control volumes. Flux reconstruction was initially developed by Huynh [6,7] and has a close connection to lifting collocation penalty by Wang and Gao [8] (LCP). Consequently, those authors coined the name correction procedure via reconstruction. Further development was made on energy stable flux reconstruction schemes by Vincent, Castonguay and Jameson [9] also known as Vincent–Castonguay–Jameson–Huynh (VCJH) schemes. The FR/CPR methods represent significant development because they have the ability to recover DG, SD and SV for linear problems. Furthermore, they are attractive because of the fact that no surface or volume integrals are required. Also, solution and flux points are arranged in a staggered fashion in SD, while flux points only lie along element interfaces in FR/CPR. As such, the latter method does not require extra flux computations at locations other than interior solution points. Using this fact, Liang, Cox and Plesniak [10] have shown improved computational efficiency of FR/CPR over SD for quadrilateral elements. For a comprehensive review of these methods, see Huynh, Wang and Vincent [11] and Wang [12].

With the development of high-order unstructured methods comes the need to achieve faster convergence, especially for solving large-scale problems using parallel computers. This demand motivates the development of time stepping techniques for which the Courant–Friedrichs–Lewy (CFL) condition is less restrictive, which is hardly the case when explicit (e.g. multi-stage Runge–Kutta) schemes are combined with high-order methods and dual time stepping. In this paper, we present an implicit scheme that overcomes the time step restriction associated with explicit schemes used for solving the unsteady incompressible Navier–Stokes equations. Work done to improve convergence of unsteady incompressible flow can be seen in Liang, Chan and Jameson [13], whereby they use a spectral difference method and Chorin's original artificial compressibility formulation (AC) [14] as well as a p -multigrid method to accelerate the convergence rate of pseudo time stepping for a particular physical time step. However, the p -multigrid method marginally improves the stiffness introduced by the artificial compressibility approach, especially for flows that require high aspect ratio elements near solid walls. As computers become equipped with larger RAM, implicit time stepping schemes are seen as effective drivers to overcome this stiffness. With these implicit schemes much larger time steps can be taken in comparison to explicit schemes, delivering the potential to improve the rate of convergence significantly. Application of the DG method to the incompressible Navier–Stokes equations was performed by Bassi et al. [15], where artificial compressibility was introduced only at the interface flux level. Shahbazi, Fischer and Ethier [16] and Nguyen, Peraire and Cockburn [17] applied DG to these equations as well, using triangular and tetrahedral grids. However, incompressible solvers involving a Poisson solver cannot be easily parallelized according to domain decomposition of the grid. One advantage of the approach taken in our current work lies in the fact that the method is discontinuous and local; as a result, there is no global matrix to split.

In recent years, the lower–upper symmetric Gauss–Seidel (LU-SGS) scheme that was originally developed by Jameson and Yoon [18] with multiple grids for solving the unsteady Euler equations has been used within the high-order CFD community for solving compressible flow problems on unstructured grids using SD [19–21] and SV [22] methods. However, when solving incompressible flows using artificial compressibility, the LU-SGS scheme is more economical because it requires the solution of only three equations in two dimensions as opposed to the four needed for compressible flow. Furthermore, with the introduction of artificial compressibility, pressure and velocity are loosely coupled and the Navier–Stokes equations take on a mixed hyperbolic/parabolic mathematical nature. This loose coupling lends itself to parallel computing as both pressure and velocity are state variables in pseudo time. As such, this paper aims to bring a popular high-order method and time stepping technique for producing high-order accurate solutions for compressible flow to the incompressible flow regime. In this regard the current method is novel, especially if it can be applied to moving and deforming grids needed to solve problems involving fluid–structure interaction (FSI) on massively parallel computers.

2. Governing equations with artificial compressibility

Numerical computation of incompressible flow is challenging because the continuity equation lacks a time-dependent term. To handle this difficulty, consider the two-dimensional unsteady incompressible Navier–Stokes equations with artificial compressibility written in conservation form

$$\frac{\partial \mathbf{U}}{\partial \tau} + \mathbf{I}_D \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) = 0 \quad (1)$$

where terms involving τ and t represent pseudo and physical time derivatives, respectively. The vector of state variables $\mathbf{U}(x, y, t) \in \Omega$, where $\Omega \subset \mathbb{R}^2$ and $t \geq 0$, and the vector of fluxes $\mathbf{F}(\mathbf{U}, \nabla \mathbf{U})$ are

$$\mathbf{U} = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (2)$$

where $\mathbf{I}_D = \text{diag}(0, 1, 1)$. The flux vector contains both inviscid and viscous terms in x and y , where $\mathbf{f} = \mathbf{f}_e - \mathbf{f}_v$ and $\mathbf{g} = \mathbf{g}_e - \mathbf{g}_v$. The inviscid fluxes for the artificial compressibility formulation are

$$\mathbf{f}_e = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \end{bmatrix}, \quad \mathbf{g}_e = \begin{bmatrix} \beta v \\ uv \\ v^2 + p \end{bmatrix} \quad (3)$$

and the viscous fluxes are

$$\mathbf{f}_v = \begin{bmatrix} 0 \\ v \frac{\partial u}{\partial x} \\ v \frac{\partial v}{\partial x} \end{bmatrix}, \quad \mathbf{g}_v = \begin{bmatrix} 0 \\ v \frac{\partial u}{\partial y} \\ v \frac{\partial v}{\partial y} \end{bmatrix}. \quad (4)$$

In the above formulation, v is the kinematic viscosity, $u = u(x, y, t)$ and $v = v(x, y, t)$ are the velocity components, β is the artificial compressibility parameter, and $p = p(x, y, t) = P(x, y, t)/\rho$ is static pressure P divided by density ρ .

Since the governing form of the incompressible equations has been modified to take on a mixed hyperbolic/parabolic nature due to the introduction of artificial compressibility, there exist three characteristics in 2D that can be obtained from the eigensystem of the convective flux Jacobians \mathbf{A} and \mathbf{B}

$$\mathbf{A} = \frac{\partial \mathbf{f}_e}{\partial \mathbf{U}} = \begin{bmatrix} 0 & \beta & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{bmatrix}, \quad \mathbf{B} = \frac{\partial \mathbf{g}_e}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & \beta \\ 0 & v & u \\ 1 & 0 & 2v \end{bmatrix}. \quad (5)$$

The eigenvalues of \mathbf{A} and \mathbf{B} are $\lambda_A = \{u - c_x, u, u + c_x\}$ and $\lambda_B = \{v - c_y, v, v + c_y\}$, where $c_x = \sqrt{u^2 + \beta}$ and $c_y = \sqrt{v^2 + \beta}$ are now termed the *pseudo* speed of sound in each direction. This means that pressure waves having finite speed are introduced into the fluid. However, these pressure waves die out as the solution approaches the divergence-free velocity state – this is termed the *pseudo* steady state. Note that the pseudo speed of sound is a function of both the velocity and the artificial compressibility parameter. Furthermore, the pseudo Mach number $M_p = u/c_x$ is always less than 1 for $\beta > 0$. This is a necessary condition in order for pressure waves to travel upstream and affect the entire flow field. More details on the eigenstructure of the incompressible equations can be found in Elsworth and Toro [23].

2.1. A short note on the artificial compressibility parameter, β

In true incompressible flow, the pressure field is affected instantaneously throughout the domain by any flow disturbances. However, in Chorin's modified formulation, the artificial compressibility parameter β , which is not a function of space or time in these simulations, introduces waves with finite speed into the fluid as a means to distribute the pressure. Consequently, the non-physical effect on the pressure field by any disturbance is delayed by some amount of time. Ideally, the value of β should be chosen as high as possible for a given algorithm such that incompressibility is recovered quickly without degradation of the solution. For steady flow, the artificial compressibility formulation can be very efficient because the divergence-free constraint for the velocity field does not have to be satisfied at each iteration. Only at the steady state does the solution take physical meaning and $\nabla \cdot \mathbf{V} = 0$. Also, computational efficiency is generally better than that for compressible flow solvers at the incompressible limit. However, unsteady computations of incompressible flows are time-consuming because any change in the flow must be propagated throughout the entire flow field. The classic formulation of Chorin has been rather well studied for transient flows, particularly by Rogers and Kwak [24], Rogers, Kwak and Kiris [25], Merkle and Athavale [26] and Soh and Goodrich [27]. Also, application of exact and approximate Riemann solvers to Chorin's formulation has been conducted by Elsworth and Toro [23,28].

In viscous flows, the formation of the boundary layer is sensitive to the streamwise pressure gradient, especially in an environment where an adverse pressure gradient can cause flow separation. If separation occurs, a pressure wave with finite speed will modify the local pressure gradient, which will ultimately affect the position of separation. This phenomenon can have a feedback effect on the pressure field. Thus, convergence issues arise if the wave does not propagate sufficiently fast. For internal flows in particular, the distribution characteristics of pressure waves within a viscous flow become rather important. To achieve good convergence in this solver, β is typically chosen between 1 and 4. For the simulations discussed here $\beta = 1.25$, unless specified otherwise.

2.2. Mapping to reference element

An arbitrary solution domain Ω is partitioned into N_q non-overlapping, conforming, linear quadrilateral elements, each denoted by Ω_q such that

$$\Omega = \bigcup_{q=1}^{N_q} \Omega_q, \quad \bigcap_{q=1}^{N_q} \Omega_q = \emptyset. \quad (6)$$

In considering non-uniformity of the solution domain, one is motivated to transform each physical element Ω_q into a reference element $\Omega_r = \{\xi, \eta | 0 \leq \xi, \eta \leq 1\}$. The transformation from a physical domain to computational domain can be achieved through an iso-parametric mapping

$$\begin{bmatrix} x \\ y \end{bmatrix} = \sum_{k=1}^K \Psi_k(\xi, \eta) \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (7)$$

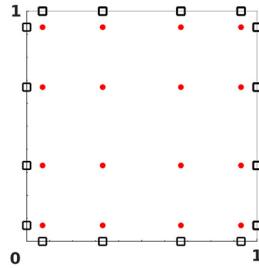


Fig. 1. Distribution of solution points (SP •) and flux points (FP □) for a 4th order FR method inside a reference element Ω_r .

where K is the number of nodes per element Ω_q , (x_k, y_k) are nodal cartesian coordinates, and $\Psi_k(\xi, \eta)$ are shape functions. For the case where Ω_q is a linear quadrilateral element, $K = 4$ and node 1 is located at $(\xi, \eta) = (0, 0)$. After transformation into the computational domain, the governing equations in Eq. (1) can be re-written in the form

$$\frac{\partial \hat{\mathbf{U}}}{\partial \tau} + \mathbf{I}_D \frac{\partial \hat{\mathbf{U}}}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{F}} = 0 \quad (8)$$

where

$$\hat{\mathbf{U}} = |\mathbf{J}| \mathbf{U}, \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{bmatrix} = |\mathbf{J}| \mathbf{J}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \quad (9)$$

In this form, (\cdot) represents values in transformed, computational space. For example, the operator $(\hat{\nabla})$ represents a gradient in the reference element Ω_r , whereas previously (∇) was a gradient in the physical element Ω_q . For a stationary grid, the metric Jacobian \mathbf{J} is independent of time and can be computed for each 2D reference element as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (10)$$

where the metrics of the Jacobian are obtained from the relationship between the non-uniform physical element and the reference element. However, to simplify notation used throughout the remaining sections, we re-write Eq. (8) as

$$\frac{\partial \hat{\mathbf{U}}}{\partial \tau} + \hat{\mathbf{S}} + \hat{\mathbf{R}} = 0 \quad (11)$$

where $\hat{\mathbf{S}}$ represents the physical time derivative and $\hat{\mathbf{R}}$ is the divergence of the flux vector, also known as the residual. These two terms combined can be called the *pseudo* residual $\hat{\mathbf{R}}_\tau = \hat{\mathbf{S}} + \hat{\mathbf{R}}$.

2.2.1. High-order curved boundaries

In the current solver, general curvature of boundaries in complex geometries is represented through implementation of Bezier curves, similar to those used in Gao, Wang and Liu [29]. The Bezier curve implementation here uses the two end points of a boundary face along with two interior control points whose locations are defined based upon the geometry of neighboring boundary faces. Thus, these Bezier curves are designed to be cubic.

2.3. Compact flux reconstruction

The compact high-order method presented here falls under the umbrella of FR/CPR. In this type of method, the flux points are only required at element interfaces. This avoids the two-grid system and subsequent computational work associated with a staggered solution point/flux point framework. Fig. 1 shows the distribution of solution and flux points for a 4th order flux reconstruction method inside a reference element. In this arrangement, the number of solution points in each coordinate direction is $N_p = 4$.

Along each direction, the solution points are located at the Legendre–Gauss points and the flux points are the two end points, 0 and 1, located along the interfaces. Choosing $P_{-1}(\xi) = 0$ and $P_0(\xi) = 1$, the higher-degree Legendre polynomials can be determined by

$$P_{N_p}(\xi) = \frac{2N_p - 1}{N_p}(2\xi - 1)P_{N_p-1}(\xi) - \frac{N_p - 1}{N_p}P_{N_p-2}(\xi). \quad (12)$$

The locations of these Legendre–Gauss quadrature points are the roots of the equation $P_{N_p}(\xi) = 0$. Using the solution at N_p solution points, a $N_p - 1$ degree polynomial can be built using the Lagrange basis

$$l_i(\xi) = \prod_{s=1, s \neq i}^{N_p} \left(\frac{\xi - \xi_s}{\xi_i - \xi_s} \right) \quad \forall \quad i = 1, \dots, N_p \quad (13)$$

and the reconstructed solution and flux vectors in Ω_r are tensor products of the two one-dimensional polynomials

$$\hat{\mathbf{U}}_r(\xi, \eta) = \sum_{j=1}^{N_p} \sum_{i=1}^{N_p} \hat{\mathbf{U}}_{r|i,j} l_i(\xi) \cdot l_j(\eta) \quad (14)$$

$$\hat{\mathbf{f}}_r^D(\xi, \eta) = \sum_{j=1}^{N_p} \sum_{i=1}^{N_p} \hat{\mathbf{f}}_{r|i,j}^D l_i(\xi) \cdot l_j(\eta). \quad (15)$$

According to Huynh [7], reconstruction for each element involves the jump in flux at each interface. Using this disparity, corrections need to be made to the discontinuous flux function given by Eq. (15) in order to define a flux function that is continuous. Following the initial 1D formulation presented by Huynh, let the polynomial $\hat{\mathbf{f}}_r$ represent a continuous flux function in Ω_r that accounts for data interaction among adjacent elements by taking on common flux values at the two interfaces. The discontinuous flux function $\hat{\mathbf{f}}_r^D$ is then corrected in the following manner

$$\hat{\mathbf{f}}_r(\xi) = \hat{\mathbf{f}}_r^D(\xi) + [\hat{\mathbf{f}}_{r-\frac{1}{2}}^{com} - \hat{\mathbf{f}}_r^D(0)]g_r^{LB}(\xi) + [\hat{\mathbf{f}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{f}}_r^D(1)]g_r^{RB}(\xi), \quad (16)$$

where $(\cdot)^{LB}$ and $(\cdot)^{RB}$ signify quantities at left and right boundary interfaces of Ω_r . This formulation provides two corrections to $\hat{\mathbf{f}}_r^D$ in 1D – one correction due to a jump in flux at the left boundary and a second correction due to a jump in flux at the right boundary of Ω_r . The current choice of correction function for all Ω_r is $g_r = g_{GA}$ from Huynh [7]. This type of correction function is seen as a trade-off between stability and accuracy when considering other choices such as g_2 and g_{DG} . The reason for this trade-off lies in the relative steepness of the correction function g_{GA} – its steepness is greater than that of g_2 but less than g_{DG} . This corresponds to a correction strength between g_2 and g_{DG} , since ultimately we require derivatives of g_r to compute derivatives of the flux. As previously mentioned, the jump in flux at the interface results in a correction to $\hat{\mathbf{f}}_r^D$ evaluated at all solution points. For the left boundary, the correction function in 1D is defined as

$$g_r^{LB}(\xi) = g_{GA}^{LB}(\xi) = \frac{N_p}{2N_p - 1} R_{R,N_p}(\xi) + \frac{N_p - 1}{2N_p - 1} R_{R,N_p-1}(\xi) \quad (17)$$

where $R_{R,N_p}(\xi)$ represents the right Radau polynomial

$$R_{R,N_p}(\xi) = \frac{(-1)^{N_p}}{2} (P_{N_p}(\xi) - P_{N_p-1}(\xi)). \quad (18)$$

The expression for a correction to the right boundary $g_r^{RB}(\xi)$ is obtained simply by reflection of $g_r^{LB}(\xi)$, so that $g_r^{RB}(\xi) = g_r^{LB}(1 - \xi)$ on the interval $\{\xi | 0 \leq \xi \leq 1\}$. The correction functions for left and right boundaries must take on the values

$$g_r^{LB}(0) = 1, \quad g_r^{LB}(1) = 0 \quad (19)$$

$$g_r^{RB}(0) = 0, \quad g_r^{RB}(1) = 1. \quad (20)$$

As stated before, the discontinuous flux function $\hat{\mathbf{f}}_r^D$ is of degree $N_p - 1$. However, both g_r^{LB} and g_r^{RB} are of degree N_p . Therefore, the function $\hat{\mathbf{f}}_r$ is of degree N_p and takes on the two common flux values at the interface

$$\hat{\mathbf{f}}_r(0) = \hat{\mathbf{f}}_{r-\frac{1}{2}}^{com} \quad (21)$$

$$\hat{\mathbf{f}}_r(1) = \hat{\mathbf{f}}_{r+\frac{1}{2}}^{com}. \quad (22)$$

This ultimately ensures flux continuity across interfaces. Once suitable correction functions for g_r^{LB} and g_r^{RB} have been defined for Ω_r , a derivative of the continuous flux function can be computed using

$$\frac{\partial \hat{\mathbf{f}}_r}{\partial \xi} = \frac{\partial \hat{\mathbf{f}}_r^D}{\partial \xi} + [\hat{\mathbf{f}}_{r-\frac{1}{2}}^{com} - \hat{\mathbf{f}}_r^D(0)] \frac{dg_r^{LB}}{d\xi} + [\hat{\mathbf{f}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{f}}_r^D(1)] \frac{dg_r^{RB}}{d\xi}. \quad (23)$$

A derivative of the discontinuous flux function $\partial \hat{\mathbf{f}}_r^D / \partial \xi$ can be computed from a Lagrange polynomial or a chain-rule. In the current implementation, we use the former method.

Now, since the mathematical nature of the governing equations has been modified due to the introduction of artificial compressibility, common inviscid fluxes $\hat{\mathbf{f}}^{com}$ at an interface can be computed using an approximate Riemann solver. Here, we employ the simple but robust local Lax–Friedrichs (LLF) flux [30]

$$\hat{\mathbf{f}}^{com} = \frac{1}{2} [\mathbf{f}_n^R + \mathbf{f}_n^L - \lambda_{max}(\mathbf{U}^R - \mathbf{U}^L)] \quad (24)$$

where the convective flux normal to an interface is $\mathbf{f}_n = \mathbf{f}_e n_x + \mathbf{g}_e n_y$ and n_x and n_y are the interface normal components. Terms written as $(\cdot)^R$ and $(\cdot)^L$ denote right and left states of the interface. The maximum eigenvalue $\lambda_{max} = \max_k |\lambda^k| = |\mathbf{V}_n| + c_n$ falls out of an eigensystem analysis of the convective flux Jacobian \mathbf{A}_n , where

$$\mathbf{A}_n = \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}} = \begin{bmatrix} 0 & \beta n_x & \beta n_y \\ n_x & 2un_x + vn_y & un_y \\ n_y & vn_x & 2vn_y + un_x \end{bmatrix}. \quad (25)$$

The velocity normal to the interface is defined as $\mathbf{V}_n = u^* n_x + v^* n_y$, where $(\cdot)^*$ indicates quantities averaged from the right and left states. The pseudo speed of sound at the interface now takes the form $c_n = \sqrt{\mathbf{V}_n^2 + \beta(n_x^2 + n_y^2)}$.

The common viscous fluxes at the interface can be computed using the first procedure of Bassi and Rebay (BR1) [31], Compact Discontinuous Galerkin (CDG) [32], Local Discontinuous Galerkin (LDG) [33], or the second procedure of Bassi and Rebay (BR2) [34]. In order to guarantee compactness of the stencil in multiple dimensions, the popular BR2 scheme is employed. To compute these terms, the derivative of $\hat{\mathbf{U}}_r$ is needed to evaluate the derivative of the discontinuous flux function $\hat{\mathbf{f}}_r^D$. Following the flux reconstruction concept in Huynh [7] and connecting it with the BR2 scheme, the gradient of the common solution is

$$\frac{\partial \hat{\mathbf{U}}_{r+\frac{1}{2}}^{com}}{\partial \xi} = \frac{1}{2} \left[\frac{\partial \hat{\mathbf{U}}_r^{RB}(1)}{\partial \xi} + \frac{\partial \hat{\mathbf{U}}_{r+1}^{LB}(0)}{\partial \xi} \right] \quad (26)$$

where

$$\hat{\mathbf{U}}_r^{RB}(\xi) = \hat{\mathbf{U}}_r(\xi) + [\hat{\mathbf{U}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{U}}_r(1)]g_r^{RB}(\xi) \quad (27)$$

$$\hat{\mathbf{U}}_{r+1}^{LB}(\xi) = \hat{\mathbf{U}}_{r+1}(\xi) + [\hat{\mathbf{U}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{U}}_{r+1}(0)]g_{r+1}^{LB}(\xi) \quad (28)$$

$$\hat{\mathbf{U}}_{r+\frac{1}{2}}^{com} = \frac{1}{2} [\hat{\mathbf{U}}_r(1) + \hat{\mathbf{U}}_{r+1}(0)]. \quad (29)$$

Therefore,

$$\frac{\partial \hat{\mathbf{U}}_{r+\frac{1}{2}}^{com}}{\partial \xi} = \frac{1}{2} \left\{ \frac{\partial \hat{\mathbf{U}}_r(1)}{\partial \xi} + \frac{\partial \hat{\mathbf{U}}_{r+1}(0)}{\partial \xi} + [\hat{\mathbf{U}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{U}}_r(1)] \frac{dg_r^{RB}(1)}{d\xi} + [\hat{\mathbf{U}}_{r+\frac{1}{2}}^{com} - \hat{\mathbf{U}}_{r+1}(0)] \frac{dg_{r+1}^{LB}(0)}{d\xi} \right\}. \quad (30)$$

Note that Eq. (30) requires data from elements r and $r + 1$ only. As a result, the computational stencil in 2D involves five elements. This compactness ultimately allows for a more economical integration of an implicit time stepping scheme. Furthermore, this stencil lends itself towards a more straightforward implementation on massively parallel computers.

2.3.1. High-order algorithm

In summary, the algorithm used to compute the residual $\hat{\mathbf{R}}$ for each reference element can be described in six major stages:

1. Construct the solution polynomial $\hat{\mathbf{U}}_r$ using the solution at N_p points and a degree $N_p - 1$ polynomial l_i built from a Lagrange basis.
2. Construct the discontinuous flux polynomial $\hat{\mathbf{f}}_r^D$ of degree $N_p - 1$ from the solution at N_p points.
3. Reconstruct the solutions and discontinuous fluxes along the edges of Ω_r from Eqs. (14)–(15).
4. Use the results from stage 3 to form common values $\hat{\mathbf{U}}_{r+1/2}^{com}$ and $\hat{\nabla} \hat{\mathbf{U}}_{r+1/2}^{com}$ needed to compute interface fluxes.
5. Construct the continuous flux polynomial $\hat{\mathbf{f}}_r$ of degree N_p by adding flux corrections based upon the jump in flux along the left and right edges of Ω_r by using correction functions g_r^{LB} and g_r^{RB} of degree N_p . For these simulations, we utilize only one type of correction function; namely $g_r = g_{GA}$.
6. Evaluate the residual $\hat{\nabla} \cdot \hat{\mathbf{F}}$ at each solution point within Ω_r using Eq. (23).

Once the above procedure is complete, a dual time stepping algorithm can be used to march the solution forward in time. We use a second-order backward Euler scheme to discretize the physical time derivative. Also, we implement an implicit backward Euler scheme with non-linear LU-SGS to march the solution in pseudo time to drive the pseudo residual $\hat{\mathbf{R}}_\tau$ to zero and evaluate its effectiveness in achieving a solenoidal velocity field for each physical time step. Results are compared to those from pseudo time marching with an explicit scheme such as Shu and Osher's total variation diminishing (TVD) three-stage third-order Runge–Kutta scheme [35] (RK3). The implementation of explicit time marching in pseudo time follows the general idea from the dual time stepping technique performed by Jameson [36].

2.4. Time stepping

2.4.1. Implicit

In order to relax the restriction on time step size and drive the pseudo residual $\hat{\mathbf{R}}_\tau$ closer and faster to zero to achieve steady state, we use an implicit time stepping scheme. The cost associated with the solution of coupled non-linear equations at each time step motivates linearization of the residual in the governing equations. This resulting linear system of equations is then solved *element-by-element* using an implicit non-linear LU-SGS iterative smoother. For each element r , we can re-write Eq. (11) as

$$\frac{\partial \hat{\mathbf{U}}_r}{\partial \tau} + \hat{\mathbf{S}}_r + \hat{\mathbf{R}}_r = 0. \quad (31)$$

The linearized set of equations is formulated below, where r and nb represent the current and neighboring quadrilateral elements Ω_r and Ω_{nb} , respectively, and n and m represent physical and pseudo iteration indices, respectively. The pseudo time derivative is discretized with a first-order backward Euler scheme

$$\frac{\hat{\mathbf{U}}_r^{n+1,m+1} - \hat{\mathbf{U}}_r^{n+1,m}}{\Delta \tau} + \hat{\mathbf{S}}_r^{n+1,m+1} + \hat{\mathbf{R}}_r^{n+1,m+1} = 0. \quad (32)$$

We can re-write Eq. (32) as

$$\frac{\hat{\mathbf{U}}_r^{n+1,m+1} - \hat{\mathbf{U}}_r^{n+1,m}}{\Delta \tau} + \hat{\mathbf{S}}_r^{n+1,m+1} - \hat{\mathbf{S}}_r^{n+1,m} + \hat{\mathbf{R}}_r^{n+1,m+1} - \hat{\mathbf{R}}_r^{n+1,m} = -\hat{\mathbf{R}}_r^{n+1,m} - \hat{\mathbf{S}}_r^{n+1,m} \quad (33)$$

where a second-order backward difference scheme (BDF2) is used for the physical time derivative,

$$\hat{\mathbf{S}}_r^{n+1,m+1} = \frac{\mathbf{I}_D}{2\Delta t} (3\hat{\mathbf{U}}_r^{n+1,m+1} - 4\hat{\mathbf{U}}_r^n + \hat{\mathbf{U}}_r^{n-1}). \quad (34)$$

Correspondingly, for pseudo time level m the discretization is

$$\hat{\mathbf{S}}_r^{n+1,m} = \frac{\mathbf{I}_D}{2\Delta t} (3\hat{\mathbf{U}}_r^{n+1,m} - 4\hat{\mathbf{U}}_r^n + \hat{\mathbf{U}}_r^{n-1}). \quad (35)$$

Gathering terms

$$\left(\frac{\mathbf{I}}{\Delta \tau} + \frac{3\mathbf{I}_D}{2\Delta t} \right) (\hat{\mathbf{U}}_r^{n+1,m+1} - \hat{\mathbf{U}}_r^{n+1,m}) + \hat{\mathbf{R}}_r^{n+1,m+1} - \hat{\mathbf{R}}_r^{n+1,m} = -\hat{\mathbf{R}}_r^{n+1,m} - \hat{\mathbf{S}}_r^{n+1,m} \quad (36)$$

and then linearizing the residual while dropping the $n+1$ index for convenience

$$\hat{\mathbf{R}}_r^{m+1} - \hat{\mathbf{R}}_r^m \approx \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \Delta \hat{\mathbf{U}}_r + \sum_{nb \neq r} \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_{nb}} \Delta \hat{\mathbf{U}}_{nb} \quad (37)$$

the following expression is obtained

$$\left(\frac{\mathbf{I}}{\Delta \tau} + \frac{3\mathbf{I}_D}{2\Delta t} + \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \right) \Delta \hat{\mathbf{U}}_r^{m+1} = -\hat{\mathbf{R}}_r^m - \hat{\mathbf{S}}_r^m - \sum_{nb \neq r} \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_{nb}} \Delta \hat{\mathbf{U}}_{nb}. \quad (38)$$

To avoid computation and storage of $\partial \hat{\mathbf{R}}_r / \partial \hat{\mathbf{U}}_{nb}$ we can perform another linearization similar to the one done in Kannan [37] and apply the LU-SGS algorithm by replacing m with k on the left hand side of Eq. (38), where k represents one forward/backward sweep through the grid, and introducing (\star) to signify the most recently computed solution. With this modification, Eq. (38) becomes

$$\left(\frac{\mathbf{I}}{\Delta \tau} + \frac{3\mathbf{I}_D}{2\Delta t} + \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \right) \Delta \hat{\mathbf{U}}_r^{k+1} = -\hat{\mathbf{R}}_r^m - \hat{\mathbf{S}}_r^m - \sum_{nb \neq r} \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_{nb}} \Delta \hat{\mathbf{U}}_{nb}^\star \quad (39)$$

and $\partial \hat{\mathbf{R}}_r / \partial \hat{\mathbf{U}}_{nb}$ is eliminated in the following manner:

$$\hat{\mathbf{R}}_r^m + \sum_{nb \neq r} \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_{nb}} \Delta \hat{\mathbf{U}}_{nb}^\star = \hat{\mathbf{R}}_r(\hat{\mathbf{U}}_c^m, \hat{\mathbf{U}}_{nb}^m) + \sum_{nb \neq r} \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_{nb}} \Delta \hat{\mathbf{U}}_{nb}^\star \quad (40)$$

$$\approx \hat{\mathbf{R}}_r(\hat{\mathbf{U}}_c^m, \hat{\mathbf{U}}_{nb}^\star) \quad (41)$$

$$\approx \hat{\mathbf{R}}_r(\hat{\mathbf{U}}_c^\star, \hat{\mathbf{U}}_{nb}^\star) - \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \Delta \hat{\mathbf{U}}_r^\star \quad (42)$$

Table 1
Size of element matrix \mathbf{M}_r for LU decomposition as a function of N_p in 2D.

N_p	\mathbf{M}_r
2	12×12
3	27×27
4	48×48
5	75×75

Substituting Eq. (42) into Eq. (39), we can obtain the final form

$$\left(\frac{\mathbf{I}}{\Delta\tau} + \frac{3\mathbf{I}_D}{2\Delta t} + \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \right) \delta \hat{\mathbf{U}}_r^{k+1} = -\hat{\mathbf{R}}_r(\hat{\mathbf{U}}_r^*, \hat{\mathbf{U}}_{nb}^*) - \hat{\mathbf{S}}_r(\hat{\mathbf{U}}_r^m, \hat{\mathbf{U}}_r^n, \hat{\mathbf{U}}_r^{n-1}) - \left(\frac{\mathbf{I}}{\Delta\tau} + \frac{3\mathbf{I}_D}{2\Delta t} \right) \Delta \hat{\mathbf{U}}_r^* \quad (43)$$

where

$$\delta \hat{\mathbf{U}}_r^{k+1} = \Delta \hat{\mathbf{U}}_r^{k+1} - \Delta \hat{\mathbf{U}}_r^* \quad (44)$$

and

$$\Delta \hat{\mathbf{U}}_r^{k+1} = \hat{\mathbf{U}}_r^{k+1} - \hat{\mathbf{U}}_r^m, \quad \Delta \hat{\mathbf{U}}_r^* = \hat{\mathbf{U}}_r^* - \hat{\mathbf{U}}_r^m. \quad (45)$$

The above linear system is solved element-by-element with direct LU decomposition and the solution is updated with

$$\hat{\mathbf{U}}_r^{k+1} = \hat{\mathbf{U}}_r^* + \delta \hat{\mathbf{U}}_r^{k+1}. \quad (46)$$

After a maximum number of sweeps k_{max} through the grid, the solution reaches pseudo time level $m + 1$, and after a maximum number of pseudo iterations m_{max} the solution reaches physical time level $n + 1$; at this time the fictitious pressure waves introduced into the fluid by artificial compressibility disappear and the solution $\hat{\mathbf{U}}_r$ takes physical meaning. It is important to note the size of the element matrix \mathbf{M}_r , which consists of terms inside the brackets on the left-hand side of Eq. (43). The size of this matrix is shown in Eq. (47) and is a function of the number of equations to solve N_{eq} , number of solution points N_p per Ω_r in 1D, and dimension D of Ω

$$\text{Size of } \mathbf{M}_r = [N_{eq} \cdot N_p^D] \times [N_{eq} \cdot N_p^D]. \quad (47)$$

Table 1 shows the size of \mathbf{M}_r as a function of N_p , where $N_{eq} = 3$ and $D = 2$. For increasingly higher orders of accuracy, the size of this matrix produces a higher memory requirement and increases the total number of operations, rendering the system more computationally expensive to solve.

The element Jacobian $\partial \hat{\mathbf{R}}_r / \partial \hat{\mathbf{U}}_r$ is computed numerically [20] at every physical iteration n by applying a perturbation ϵ to each state variable,

$$\left. \frac{\partial \hat{\mathbf{R}}_r}{\partial \hat{\mathbf{U}}_r} \right|_n = \frac{\hat{\mathbf{R}}_r(\hat{\mathbf{U}}_r^n + \epsilon, \hat{\mathbf{U}}_{nb}^n) - \hat{\mathbf{R}}_r(\hat{\mathbf{U}}_r^n, \hat{\mathbf{U}}_{nb}^n)}{\epsilon} \quad (48)$$

where ϵ is typically $\sqrt{\epsilon_{machine}}$. To help improve convergence rates, however, the computation of $\partial \hat{\mathbf{R}}_r / \partial \hat{\mathbf{U}}_r$ can be frozen without degrading the solution. In other words, this term can be recomputed every 1–5 physical iterations, depending on time step size.

2.4.2. Explicit

The popular TVD RK3 scheme [35] when applied to pseudo time stepping can be written as

$$\begin{aligned} \hat{\mathbf{U}}_k^{n+1,m} &= a_k \hat{\mathbf{U}}_0^{n+1,m} + b_k \hat{\mathbf{U}}_{k-1}^{n+1,m} - c_k \Delta\tau \left[\hat{\mathbf{S}}_{k-1}^{n+1,m} + \hat{\mathbf{R}}_{k-1}^{n+1,m} \right] \\ \forall k &= 1, \dots, 3, \quad m = 1, \dots, m_{max} \end{aligned} \quad (49)$$

where m represents the number of RK3 pseudo iterations. At the start of each round of pseudo iterations, when $m = 1$, $\hat{\mathbf{U}}_0^{n+1,1}$ is the solution from the previous physical time step. However, for $m > 1$, $\hat{\mathbf{U}}_0^{n+1,m}$ represents the solution from the previous pseudo time step. The physical time discretization exists as a source term and is given by

$$\hat{\mathbf{S}}_{k-1}^{n+1,m} = \frac{\mathbf{I}_D}{2\Delta t} (3\hat{\mathbf{U}}_{k-1}^{n+1,m} - 4\hat{\mathbf{U}}^n + \hat{\mathbf{U}}^{n-1}) \quad (50)$$

The index k signifies different stages of the Runge–Kutta scheme for which the coefficients a_k , b_k and c_k are defined as

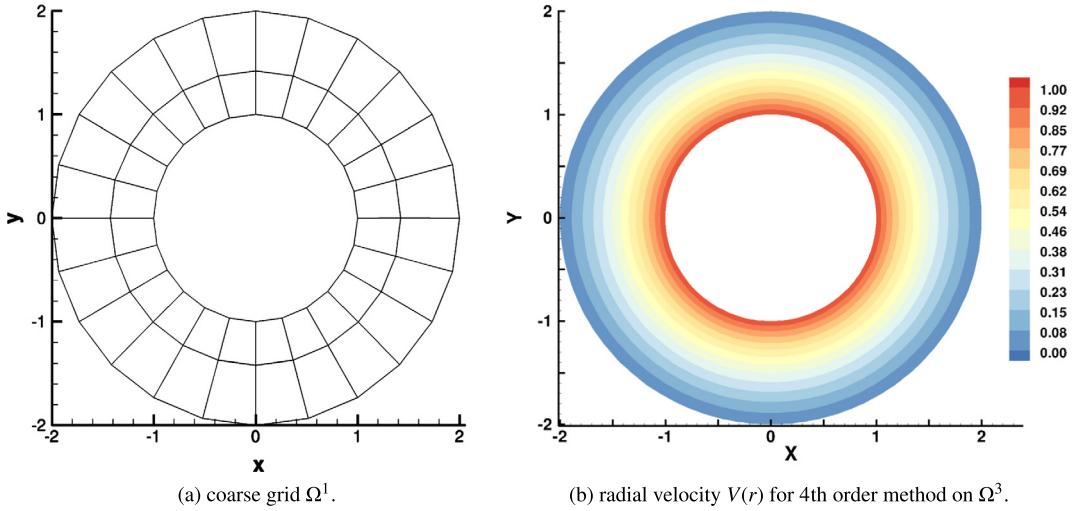


Fig. 2. Incompressible Taylor–Couette flow at $Re = 10$.

$$\begin{aligned} stage \quad k = 1 : & \{a_1, b_1, c_1\} = \{1, 0, 1\} \\ stage \quad k = 2 : & \{a_2, b_2, c_2\} = \left\{ \frac{3}{4}, \frac{1}{4}, \frac{1}{4} \right\} \\ stage \quad k = 3 : & \{a_3, b_3, c_3\} = \left\{ \frac{1}{3}, \frac{2}{3}, \frac{2}{3} \right\} \end{aligned}$$

3. Steady flow: Taylor–Couette flow with curved boundaries

Proper implementation of the numerical method is verified by comparing results to the exact solution for steady incompressible Taylor–Couette flow [38]. The analytical expression for radial velocity $\tilde{V}(r)$ in this problem is

$$\tilde{V}(r) = r_i \omega_i \frac{\frac{r_o}{r} - \frac{r}{r_o}}{\frac{r_o}{r_i} - \frac{r_i}{r_o}} + r_o \omega_o \frac{\frac{r}{r_i} - \frac{r_i}{r}}{\frac{r_o}{r_i} - \frac{r_i}{r_o}} \quad (51)$$

where $r_i = 1$ and $r_o = 2$ are the inner and outer radius and ω_i and ω_o are the inner and outer cylinder angular speeds, respectively. The inner cylinder spins clockwise at a rate $\omega_i = 1$ while the outer cylinder is stationary. At $t = 0$, initial u and v velocity conditions are given based upon the analytical expression and cubic Bezier curves are implemented for curved boundaries, where no-slip velocity conditions are satisfied through specification of exact flux. The Reynolds number $Re = r_i \omega_i d / v$ is 10, where $d = r_o - r_i$. Refinement is done uniformly from the coarsest grid Ω^1 shown in Fig. 2(a) down to the finest grid Ω^4 . The L^p -norm of radial velocity error $\|\epsilon\|_p$ for $N_p = \{2, 3, 4\}$ is computed by Eq. (52)

$$\|\epsilon\|_p = \left(\frac{\sum_{k=1}^{DOF} |\epsilon_k|^p}{DOF} \right)^{\frac{1}{p}} \quad (52)$$

where $\epsilon(r) = \tilde{V}(r) - V(r)$ and $V(r)$ is the radial velocity obtained numerically. Total degrees of freedom are denoted by DOF , which represents the total number of solution points in Ω . The subsequent orders of accuracy \mathcal{P} achieved are computed by Eq. (53)

$$\mathcal{P}_{L^p} = \frac{\log \left[\frac{\|\epsilon\|_p(\Omega^d)}{\|\epsilon\|_p(\Omega^{d+1})} \right]}{\log \left[\frac{h(\Omega^d)}{h(\Omega^{d+1})} \right]} \quad (53)$$

where $p = \{1, 2\}$ to represent different norms and $h(\Omega^i) = 1/\sqrt[DOF]{DOF}$ for $\Omega^i \subset \mathbb{R}^D$. For this benchmark test case, $i = \{1, 2, 3, 4\}$ to represent four uniformly refined domains with number of elements 48, 192, 768 and 3072, respectively. For this two-dimensional benchmark test, $D = 2$.

The profile of radial velocity $V(r)$ obtained numerically with a 4th order method on Ω^3 is shown in Fig. 2(b) and reductions in L^1 -norm and L^2 -norm of the error are tabulated in Table 2. These order of accuracy results are obtained exactly using both explicit and implicit pseudo time stepping, with and without inclusion of the source term from physical

Table 2

Taylor–Couette at $Re = 10$: order of accuracy obtained from L^1 -norm and L^2 -norm of errors during h -refinement.

Domain	Elements	DOF	$\ \epsilon\ _1$	\mathcal{P}_{L^1}	$\ \epsilon\ _2$	\mathcal{P}_{L^2}
(a) $N_p = 2$						
Ω^1	48	192	7.97E-2	-	9.55E-2	-
Ω^2	192	768	1.74E-2	2.20	2.16E-2	2.14
Ω^3	768	3072	3.98E-3	2.13	5.05E-3	2.10
Ω^4	3072	12288	9.26E-4	2.10	1.19E-3	2.08
(b) $N_p = 3$						
Ω^1	48	432	6.94E-3	-	8.99E-3	-
Ω^2	192	1728	6.45E-4	3.42	8.81E-4	3.35
Ω^3	768	6912	8.45E-5	2.93	1.07E-4	3.04
Ω^4	3072	27648	2.01E-5	2.07	2.40E-5	2.16
(c) $N_p = 4$						
Ω^1	48	768	1.64E-3	-	1.93E-3	-
Ω^2	192	3072	6.80E-5	4.59	8.16E-5	4.56
Ω^3	768	12288	3.71E-6	4.20	4.57E-6	4.16
Ω^4	3072	49152	2.22E-7	4.06	2.89E-7	3.98

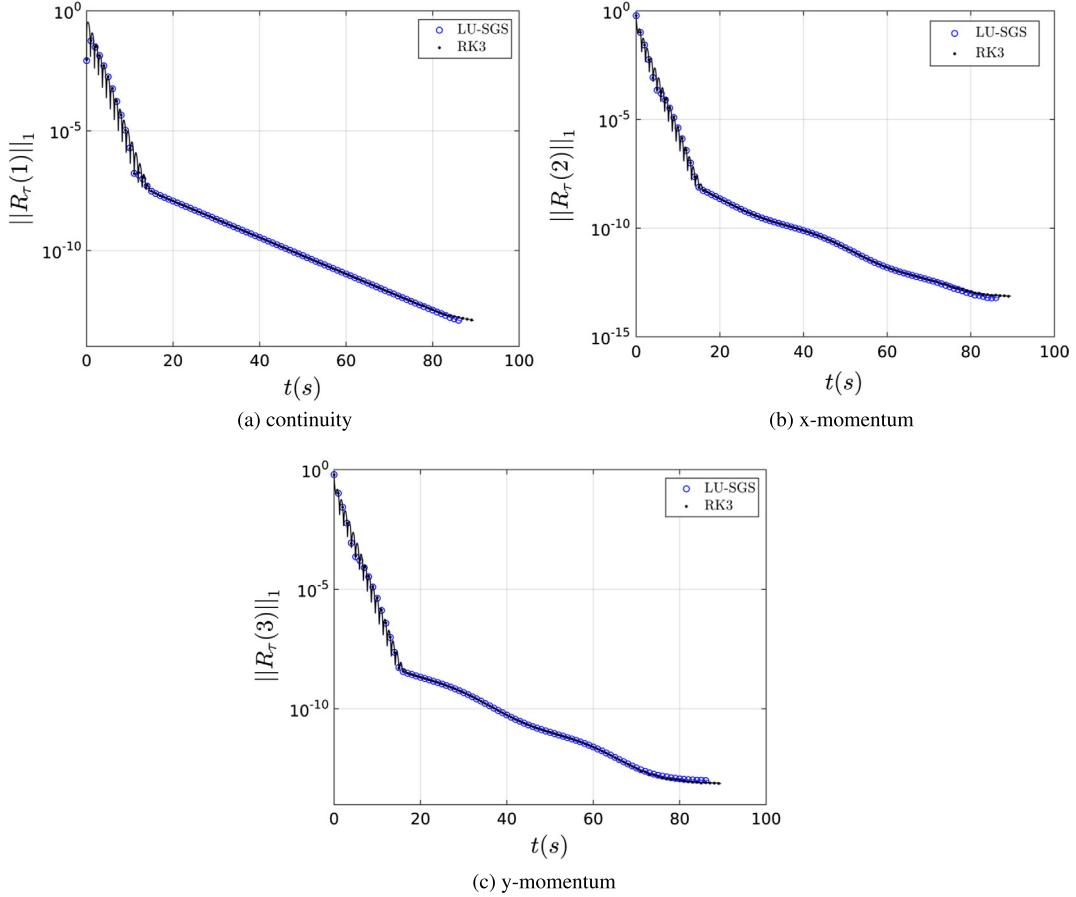


Fig. 3. L^1 -norm of continuity and momentum pseudo residuals, $\|R_\tau(\cdot)\|_1$, for a 4th order method on Ω^3 .

time discretization $\hat{\mathbf{S}}$. Convergence of the L^1 -norm of pseudo residuals $R_\tau(\cdot) = \hat{R}_\tau(\cdot)/|\mathbf{J}|$ on Ω^3 from the continuity and momentum equations are plotted against physical time t in Fig. 3 using a 4th order FR method. The same physical and pseudo time step sizes are used for both RK3 and LU-SGS time stepping schemes and $\|R_\tau\|_1$ is required to decrease one order of magnitude within each physical iteration. As expected, the results from both schemes overlap each other and the level to which each residual drops is 10^{-13} . This analysis verifies that both RK3 and LU-SGS are implemented correctly under dual time stepping. Although both schemes perform well in computing this steady flow, the explicit scheme exhibits difficulty in addressing stiffness introduced by artificial compressibility in the modified continuity equation for unsteady

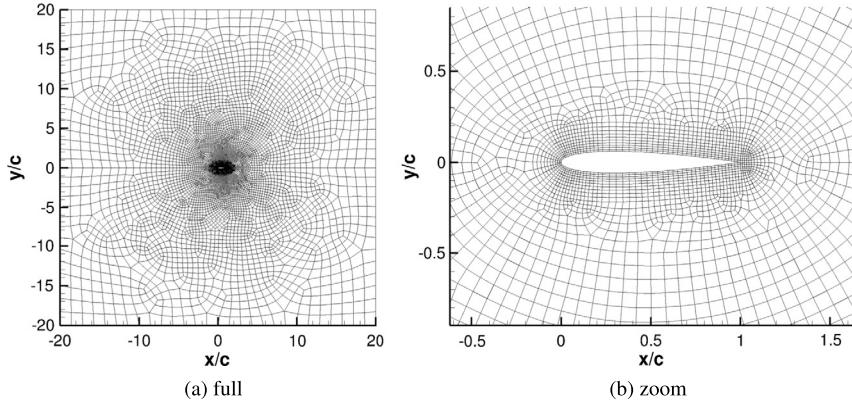


Fig. 4. NACA0012 mesh.

flow. This fact is highlighted in Section 7, where we compare results between the explicit and implicit schemes for unsteady flow past a cylinder and show the latter scheme to be a promising choice.

4. Steady flow: NACA0012 airfoil

In order to demonstrate performance of implicit dual time stepping on a highly non-uniform grid, numerical results were obtained for steady flow past a NACA0012 airfoil at zero angle of attack at $Re = 1850$ and unit chord length. The grid used for this simulation, shown in Fig. 4, contains a total of 8114 linear quadrilateral elements with 100 of these representing the profile of the airfoil, which exhibits a sharp trailing edge. Curvature of the airfoil is captured using our cubic Bezier curve implementation. The blockage in y caused by the presence of the airfoil in the 40×40 domain is 0.3%. Initial conditions are set to $[p, u, v]^T = [1, 1, 0]^T$. At the inlet boundary, freestream velocity conditions are given while at the outlet, only pressure is specified. Far field boundary conditions are applied to the top and bottom boundaries of the domain. The no-slip velocity condition is applied to the airfoil surface. We run 2nd, 3rd and 4th order FR methods, which correspond to 32456, 73026 and 129824 degrees of freedom, respectively. Since we seek a steady state solution, we can remove dual time stepping and ignore the term $\hat{\mathbf{S}}$ resulting from physical time discretization. To demonstrate the effectiveness of sweeping within the LU-SGS scheme in driving $\nabla \cdot \mathbf{V}$ to zero and ultimately the residual $\hat{\mathbf{R}}_\tau$ to zero, we run multiple scenarios for the sweep count k_{max} . The various sweep counts used in this study are $k_{max} = \{1, 3, 6, 10\}$, where one forward/backward movement through the grid constitutes one full sweep.

The implicit scheme offers increased stability and permits highly accelerated convergence to the steady state. We accomplish this by setting the initial maximum CFL at $m = 0$ to 0.5 and increasing it exponentially [22] at every pseudo iteration with $CFL|_m = CFL|_{m=0} \cdot a^m$, where the factor a is set to 1.2. Using this function and the definition for CFL , this equates to increasing $\Delta\tau$ exponentially towards ∞ .

Fig. 5 shows steady state pressure contours around the leading and sharp trailing edge for various orders of accuracy. As expected, improvements in contour shape can be seen with higher orders of accuracy. Contours for u -velocity and convergence rates of $\|\mathbf{R}_\tau\|_1$ are shown in Fig. 6 for all sweep count numbers. The fast reduction in residual using the LU-SGS scheme due to the increased pseudo time step can be seen in this figure. Maximum CFL numbers using RK3 for 2nd, 3rd and 4th order methods are 0.16, 0.14 and 0.07, respectively. This figure also shows the effectiveness of LU-SGS sweeping in dropping the steady state residual by an order of magnitude, from $k_{max} = 1$ to $k_{max} = 10$. Speedup factors for different values of N_p with the various sweep counts are tabulated in Table 3 and are computed at the point where $\|\mathbf{R}_\tau\|_1 = 1.0E-5$. Results show that LU-SGS minimum wall clock times have been achieved for $k_{max} = 10$. For all LU-SGS simulations in this test case, the element Jacobian $\partial\hat{\mathbf{R}}_r/\partial\hat{\mathbf{U}}_r$ is computed every five iterations. Slightly faster convergence can be achieved through optimization of this incremental value. Convergence of the drag coefficient obtained through polynomial order enrichment (p -refinement) is shown in Fig. 7, with a converged value of $C_D = 0.0867$.

5. Unsteady flow: Taylor–Green decaying vortices

The Taylor–Green decaying vortex problem with periodic boundary conditions is commonly used to verify a numerical method for simulation of unsteady incompressible flow. The periodic conditions help to remove influence from boundaries of the domain on the accuracy of the method. We solve the unsteady Navier–Stokes equations on a square domain $\{x, y | 0 < x, y < 2\pi\}$ at $Re = 10$. We set the initial condition to the exact solution

$$p(x, y, t) = -\frac{\cos(2qx) + \cos(2qy)}{4} \exp\left(\frac{-4q^2t}{Re}\right) \quad (54)$$

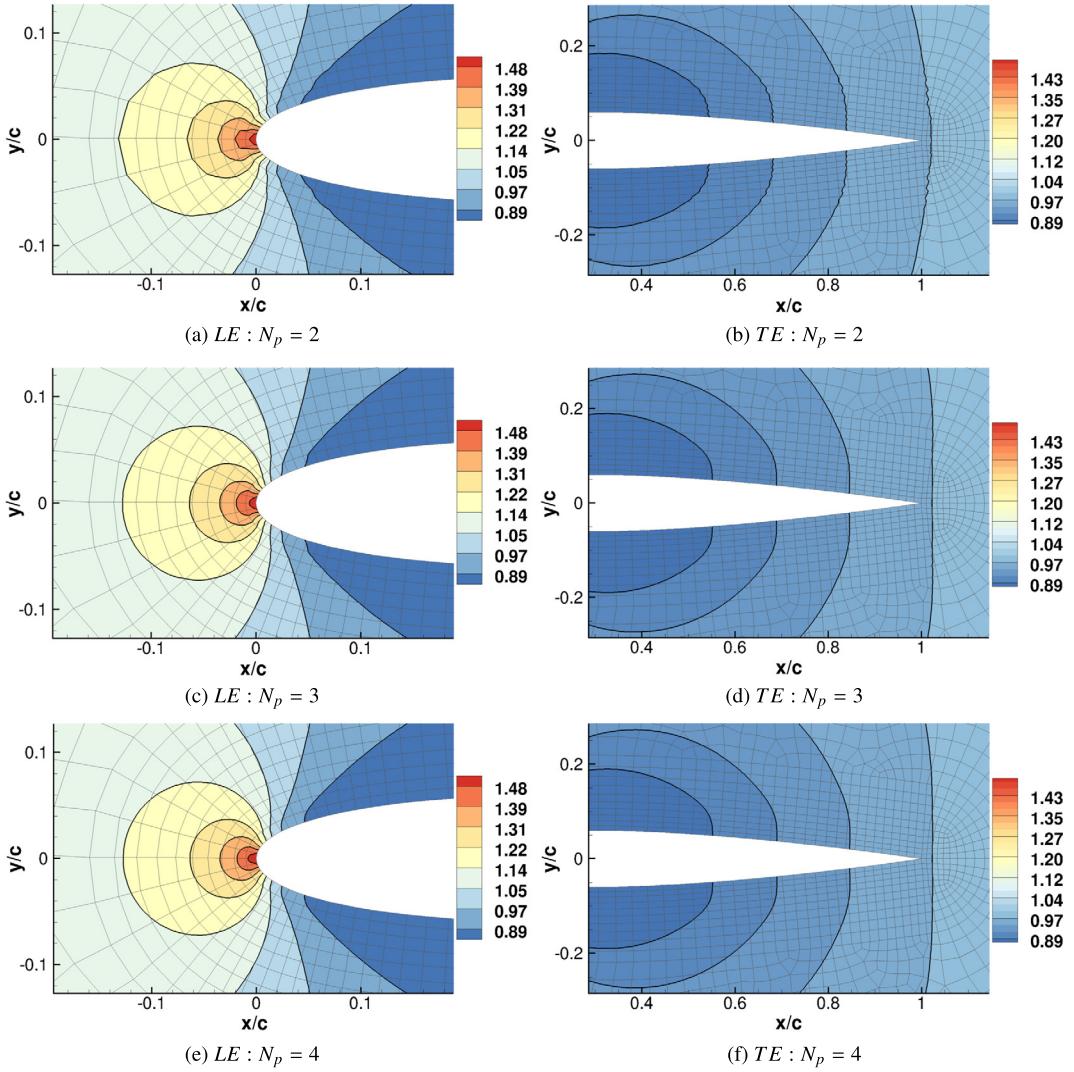


Fig. 5. NACA0012 at $Re = 1850$: pressure at LE (a, c, e) and TE (b, d, f) from LU-SGS pseudo time stepping with $k_{max} = 10$.

$$u(x, y, t) = -\cos(qx) \sin(qy) \exp\left(\frac{-2q^2t}{Re}\right) \quad (55)$$

$$v(x, y, t) = \sin(qx) \cos(qy) \exp\left(\frac{-2q^2t}{Re}\right) \quad (56)$$

at $t = 0$, where $q = 1$ is the Taylor array number. Similar to Section 3, h -refinement is done uniformly from the coarsest grid Ω^1 (4×4) down to the finest grid Ω^4 (32×32); however, to render the elements more irregular a perturbation of 10% is applied to the mesh nodes based on h . The L^p -norms are computed using u -velocity. A small constant time step of $\Delta t = 10^{-4}$ is chosen to ensure that spatial accuracy does not degrade due to errors resulting from time discretization. At this small time step, the maximum number of pseudo iterations is set at $m_{max} = 20$ and the sweep count at $k_{max} = 3$.

The vorticity field obtained numerically with a 4th order FR method using LU-SGS on Ω^3 is shown in Fig. 8, and the L^p -norm of the errors are tabulated in Table 4 at $t = 1$ s. These results demonstrate the order of accuracy for 2nd, 3rd and 4th order methods. Comparing results in Table 4(b) for Ω^3 and Table 4(c) for Ω^2 , we can see that by increasing the order from 3 to 4 similar levels of $\|\epsilon\|_1$ and $\|\epsilon\|_2$ can be achieved with less than half the number of DOFs. Furthermore, order enrichment results gathered from Table 4 and plotted in Fig. 9 demonstrate exponential rate of convergence.

6. Unsteady flow: double periodic shear layers

The second unsteady test case presented is a double periodic pair of horizontal shear layers with a small vertical perturbation in a square domain $\{x, y | 0 < x, y < 1\}$ with periodic boundary conditions. This problem was introduced by Bell,

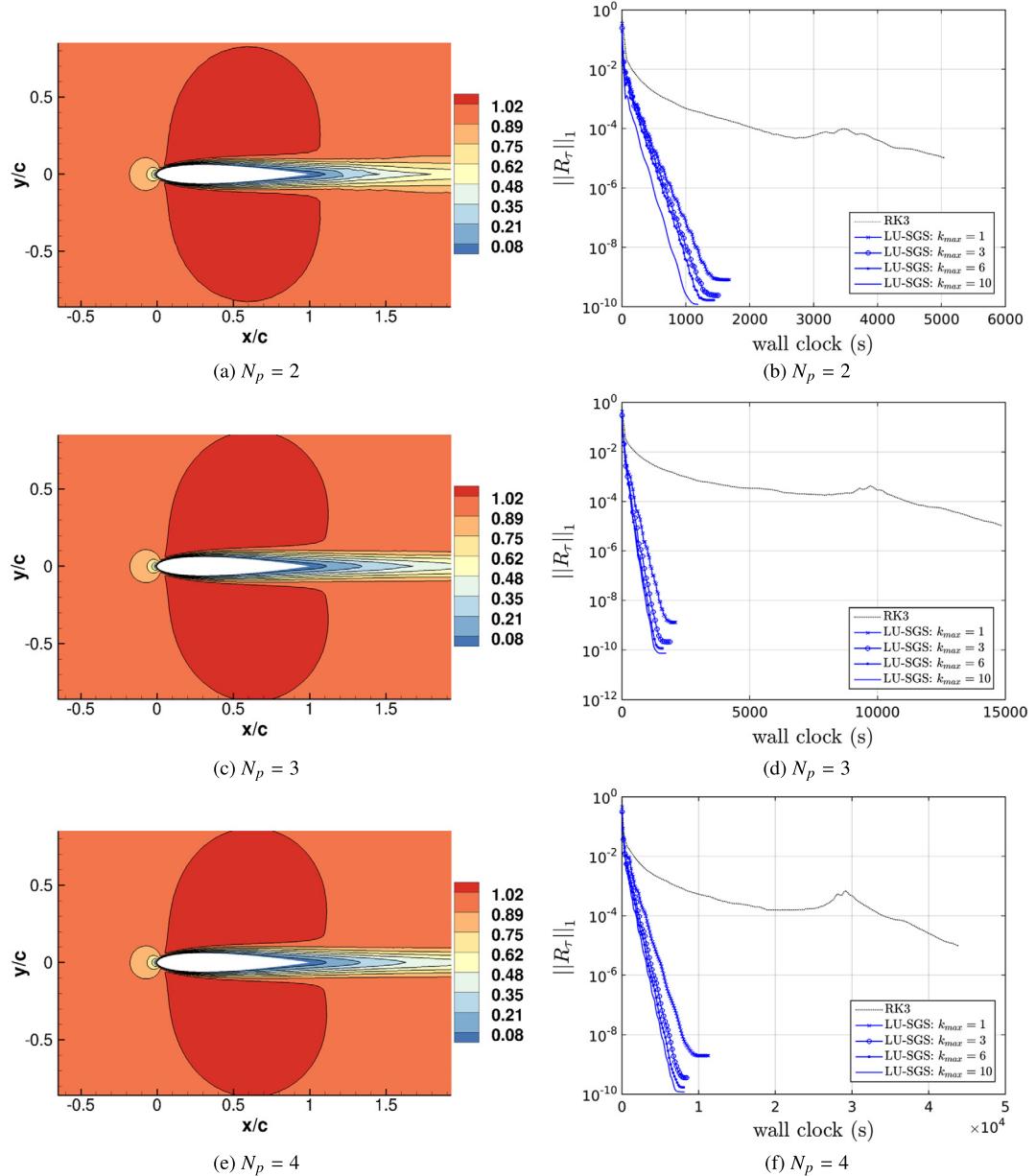


Fig. 6. NACA0012 at $Re = 1850$: u -velocity from LU-SGS pseudo time stepping with $k_{max} = 10$ (a, c, e). L^1 -norm of pseudo residual, $\|R_\tau\|_1$, as a function of wall clock time for various sweep numbers (b, d, f).

Table 3

NACA0012 at $Re = 1850$: speedup.

RK3					LU-SGS					RK3					LU-SGS						
(a) $N_p = 2$	k_{max}	—	1	3	6	10	10	(b) $N_p = 3$	k_{max}	—	1	3	6	10	(c) $N_p = 4$	k_{max}	—	1	3	6	10
Speedup	1	9.0	10.1	10.9	14.4			Speedup	1	19.8	25.6	28.4	28.8		Speedup	1	10.5	13.9	14.6	16.7	

Colella and Glaz [39] to test convergence properties of their projection method and later studied by Minion and Brown [40] to assess the effect of grid resolution on the behavior of various difference approximations for the unsteady incompressible Navier–Stokes equations. This is considered a “thick” shear layer problem with initial conditions given by

$$u(x, y, 0) = \begin{cases} \tanh[(y - 0.25)/\rho], & \text{if } y \leq 0.5 \\ \tanh[(0.75 - y)/\rho], & \text{if } y > 0.5 \end{cases} \quad (57)$$

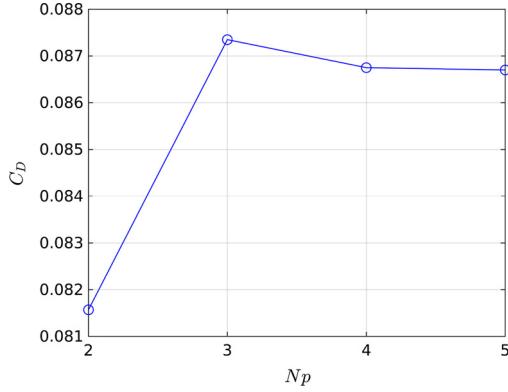


Fig. 7. NACA0012 at $Re = 1850$: C_D convergence using order enrichment (p -refinement) and LU-SGS.

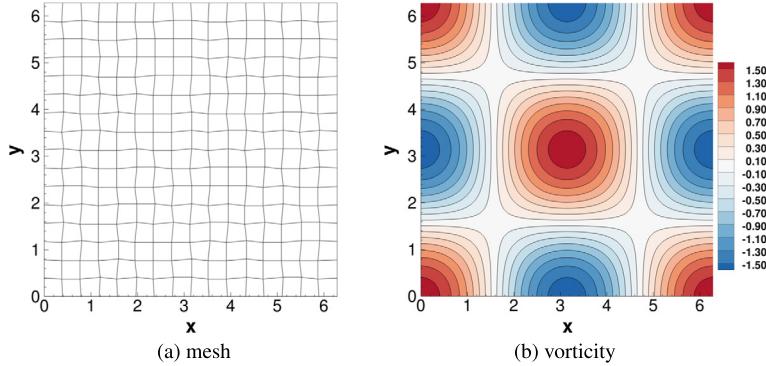


Fig. 8. Taylor-Green at $Re = 10$: 4th order FR method with LU-SGS pseudo time stepping on Ω^3 at $t = 1$ s.

Table 4

Taylor-Green at $Re = 10$: order of accuracy obtained from L^1 -norm and L^2 -norm of errors during h -refinement, with LU-SGS pseudo time stepping.

Domain	Elements	DOF	$\ \epsilon\ _1$	\mathcal{P}_{L^1}	$\ \epsilon\ _2$	\mathcal{P}_{L^2}
(a) $N_p = 2$						
Ω^1	16	64	3.15E-1	–	3.91E-1	–
Ω^2	64	256	1.18E-1	1.42	1.48E-1	1.40
Ω^3	256	1024	1.90E-2	2.63	2.35E-2	2.66
Ω^4	1024	4096	2.68E-3	2.83	3.31E-3	2.83
(b) $N_p = 3$						
Ω^1	16	144	3.27E-2	–	4.05E-2	–
Ω^2	64	576	3.21E-3	3.35	4.05E-3	3.32
Ω^3	256	2304	2.38E-4	3.75	2.95E-4	3.77
Ω^4	1024	9216	2.66E-5	3.16	3.31E-5	3.16
(c) $N_p = 4$						
Ω^1	16	256	3.98E-3	–	4.96E-3	–
Ω^2	64	1024	2.28E-4	4.12	2.79E-4	4.15
Ω^3	256	4096	1.04E-5	4.46	1.30E-5	4.42
Ω^4	1024	16384	6.80E-7	3.93	8.59E-7	3.92

$$v(x, y, 0) = \delta \sin(2\pi x) \quad (58)$$

where $\rho = 1/30$ represents shear layer thickness (smaller ρ produces thinner shear layers) and $\delta = 0.05$ the amplitude of the initial perturbation. The vorticity solution at various times can be seen in Fig. 10. Each shear layer evolves into a periodic arrangement of large symmetric vortices, with the shear layer being thinned by the large straining flow field. The thinning shear layers eventually roll up with the large vortices. To test the convergence of our solver, we compute the solution at $t = 1$ s using 32×32 , 64×64 and 128×128 grids, similar to the grid shown in Fig. 8(a). The Reynolds number is $Re = 500$ and physical and pseudo time steps are $1.0E-4$. Max pseudo iterations were set to $m_{max} = 20$ and the sweep count to $k_{max} = 3$. The convergence rates for 2nd, 3rd and 4th order FR methods are estimated in the same manner as Minion and Brown [40] using a procedure based on Richardson extrapolation and provided in Table 5.

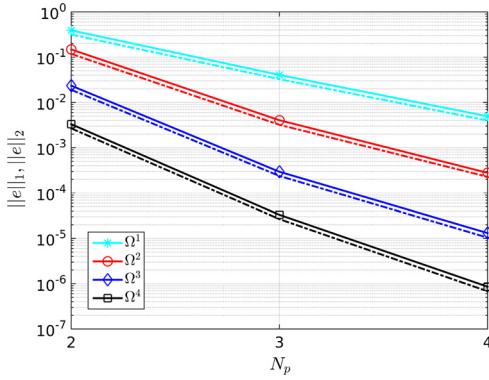


Fig. 9. Taylor–Green at $Re = 10$: p -convergence of $\|\epsilon\|_1$ (dashed) and $\|\epsilon\|_2$ (solid).

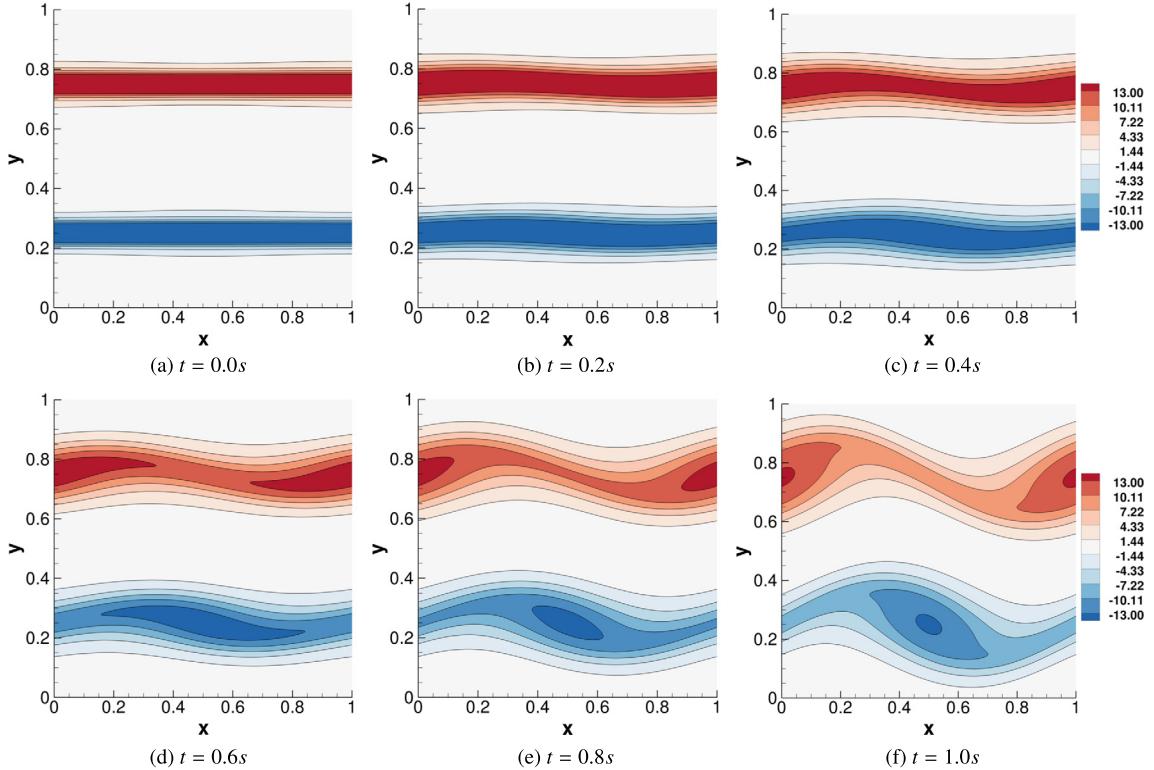


Fig. 10. Double shear layer at $Re = 500$: vorticity for 4th order FR method with LU-SGS pseudo time stepping.

To illustrate the performance of our high-order FR method with implicit pseudo time stepping in resolving this shear layer at higher Reynolds number, we increase Re to 10,000 and march the solution to $t = 1.8$ s on 32×32 , 64×64 and 128×128 regular grids. As the larger physical vortices develop and the shear layer thins, non-physical vortices originate in the shear layer due to insufficient grid and/or order resolution. These vortices are further stretched as they wrap around the larger ones. The spurious nature of these vortices is confirmed by observing their elimination from the solution during h - and p -refinement. This fact can be seen in Fig. 11, which shows existence of spurious vortices using $N_p = \{2, 3, 4\}$ on a 32×32 grid and using $N_p = 2$ on a 64×64 grid. Furthermore, comparing results in Figs. 11(b) and 11(g), which both use 128 DOFs in each direction, we can conclude that performance of a 4th order scheme is greater than a 2nd order scheme in attempting to remove the spurious vortices. If we refine these grids further to contain $DOFs \geq 256$ we can appropriately resolve the flow field and eliminate any non-physical numerical artifact.

7. Unsteady flow: cylinder

The novelty of this paper is the design of an implicit dual time stepping scheme for the high-order FR method to predict unsteady flow in complex geometries. In this section, we report numerical results for unsteady laminar flow at $Re = 100$

Table 5

Double Shear Layer at $Re = 500$: convergence rate obtained from L^2 -norm of errors using Richardson extrapolation at $t = 1$ s.

	$N_p = 2$	$N_p = 3$	$N_p = 4$
$\ u_{\Omega^1} - u_{\Omega^2}\ _2$	2.263E-2	2.569E-3	1.367E-4
$\ u_{\Omega^2} - u_{\Omega^3}\ _2$	6.374E-3	3.817E-4	1.053E-5
Richardson Rate	1.83	2.75	3.70

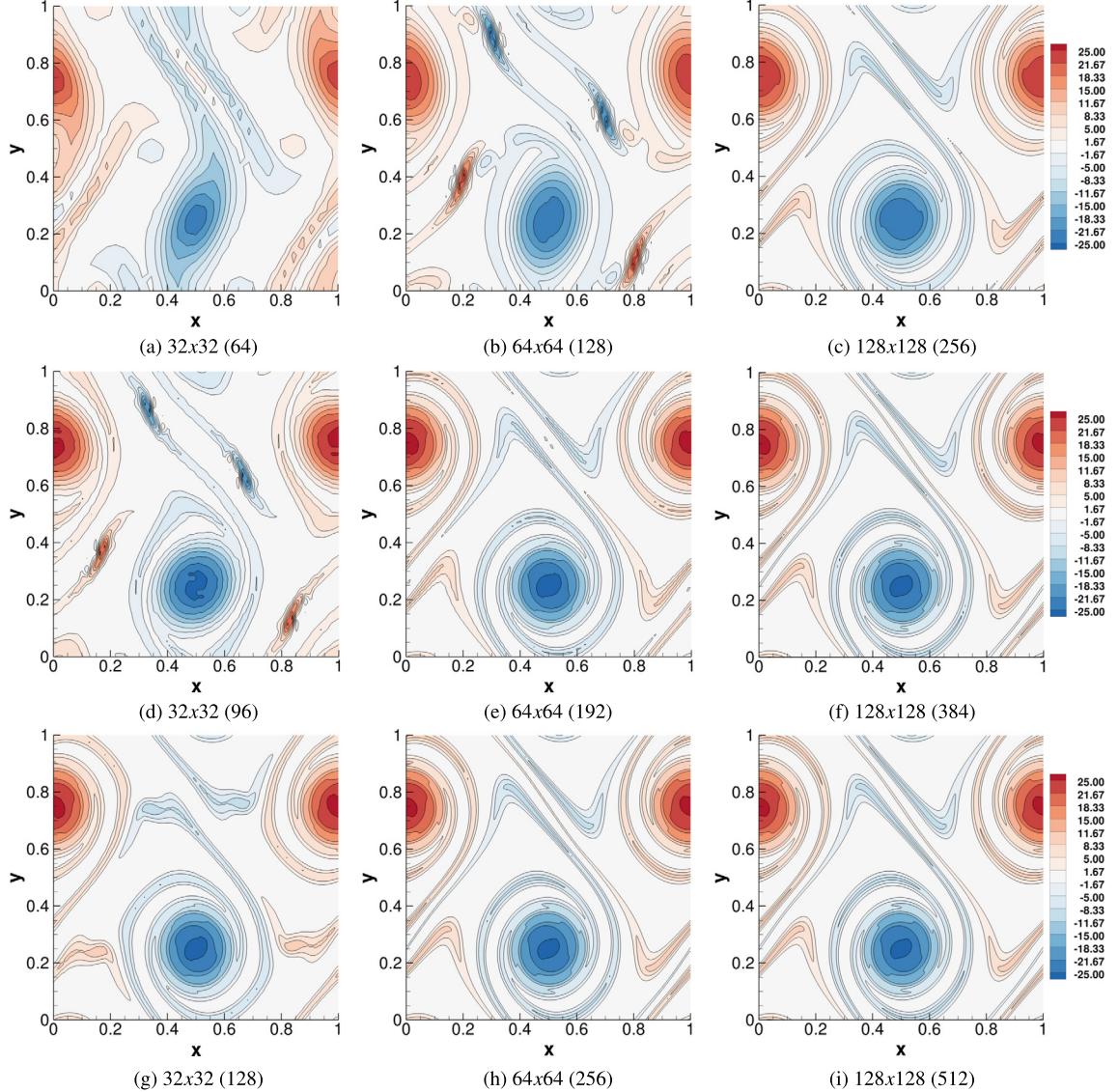


Fig. 11. Double shear layer at $Re = 10,000$: vorticity at $t = 1.8$ s for $N_p = 2$ (a–c), $N_p = 3$ (d–f) and $N_p = 4$ (g–i). DOFs along each coordinate direction are given in parenthesis.

past a stationary cylinder with diameter $d = 1$. The grid used for this test case shown in Fig. 12 contains 4371 linear quadrilateral elements with 48 elements along the wall of the cylinder, which again is represented by our cubic Bezier curves. The blockage in y caused by the presence of the cylinder in the 50×40 domain is 2.5%. The initial and boundary conditions applied to the NACA0012 airfoil were also applied to this geometry. As before, we run 2nd, 3rd and 4th order FR methods, which correspond to 17484, 39339 and 69936 degrees of freedom in Ω , respectively.

The pseudo time step to be taken by the implicit scheme is chosen to be $100\times$ larger than that afforded by the explicit scheme. Furthermore, increased stability offered by the LU-SGS implementation can permit a $100\times$ larger physical time step, while still capturing the transient process. The pseudo and physical time step size used by the explicit scheme is

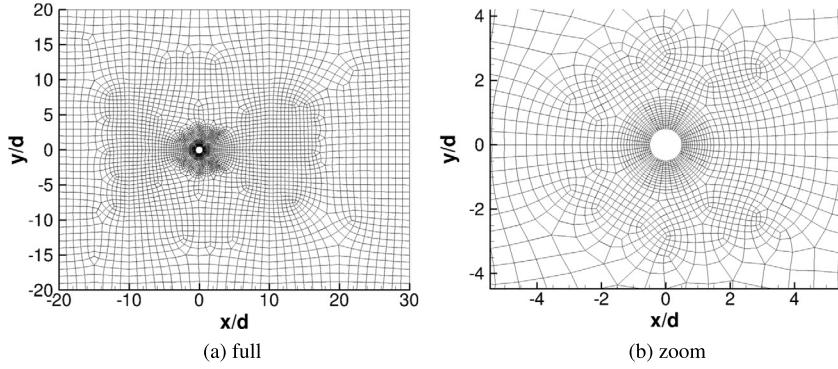


Fig. 12. Cylinder grid.

$\Delta\tau^{RK3} = \Delta t^{RK3} = 1.0E-3$ for a 3rd order method and $\Delta\tau^{RK3} = \Delta t^{RK3} = 6.0E-4$ for a 4th order method. Cost per physical iteration is higher for the implicit scheme and mainly depends upon four factors: (1) the frequency of computing the element Jacobian $\partial\mathbf{R}_r/\partial\mathbf{U}_r$, (2) size of the element matrix \mathbf{M}_r (i.e. polynomial order), (3) number of pseudo iterations, and (4) number of sweeps through the mesh. For unsteady flow calculations, a criterion is used during each physical iteration, whereby the pseudo residual is evaluated at the end of each pseudo iteration to determine if the continuity equation has been satisfied. The criterion for pseudo steady state convergence per physical time step requires that $\|\mathbf{R}_\tau\|_1$ drop by a factor of 10^3 . If this value is not reached, the maximum number of pseudo iterations for explicit and implicit schemes is set to $m_{max}^{RK3} = 20$ and $m_{max}^{LU-SGS} = 200$. The value for m_{max}^{LU-SGS} is made larger by an order of magnitude to account for the increased time step size associated with this scheme. Larger $\Delta\tau$ and Δt cause the element matrix \mathbf{M}_r to be less diagonally dominant and result in more pseudo iterations to achieve convergence to the pseudo steady state than smaller time step sizes. The LU-SGS sweep count number is set to $k_{max} = 3$. In this unsteady flow problem, $\partial\hat{\mathbf{R}}_r/\partial\hat{\mathbf{U}}_r$ is computed at every single physical iteration.

Instantaneous vorticity contours at approximately $t = 360$ s for 2nd, 3rd and 4th order methods can be seen in Fig. 13 for both RK3 and LU-SGS schemes. Figs. 13(a)/13(b) demonstrate that the current grid is inadequate for a 2nd order FR method to maintain the integrity of the vortex shedding downstream of the bluff body. Figs. 13(c)/13(d) and 13(e)/13(f) demonstrate that higher order polynomials have the ability to capture the vortex shedding properly, with the latter set of figures from a 4th order method displaying the smoothest contours of vorticity.

Plots of force coefficients C_L and C_D generated by a 3rd order method are shown in Fig. 14 for roughly 32 vortex shedding cycles. These plots show a visual comparison between RK3 and LU-SGS, with the former exhibiting a slightly larger amplitude for C_L (1.6%) and larger mean value for C_D (0.15%). Values for Strouhal number and various force coefficients are tabulated in Table 6 using data collected from approximately 60 data points per shedding cycle over 50 cycles. A fast Fourier transform was performed to capture the shedding frequency f_s needed to compute the Strouhal number $St = f_s d/U_\infty$. Coefficients of lift and drag due to pressure ($C_{L,p}$, $C_{D,p}$) and viscosity ($C_{L,v}$, $C_{D,v}$) are shown and compared to data from Park, Kwon and Choi [41] and Sharman et al. [42]. The former authors employed the fractional step method of Choi and Moin [43] while the latter authors utilized the SIMPLE algorithm [44]. Results from Posdziech and Grundmann [45] are reported as well. Root mean square (RMS) values are indicated by $(\cdot)'$. For a 4th order FR method with LU-SGS, the RMS of lift coefficient ($C'_L = 0.23$) exactly matches the values obtained by Park et al. and Sharman et al., while the RMS of drag coefficient ($C'_D = 0.0063$) is smaller by approximately 1.5% but exactly equal to the value reported by Posdziech et al. Additionally, amplitudes of C_L and C_D are provided in Table 7 along with results from different authors such as Chan et al. [46], Liu et al. [47], Ding et al. [48,49] and Williamson [50]. We compute C_L by taking an average of peak values and we compute C_D by averaging all values over the 50 shedding cycles. The uncertainty in drag is calculated by averaging the differences between all peak values and the mean value. Note that the current incompressible 4th order results using LU-SGS agree extremely well with the result from Chan, who used a low speed compressible flow solver with a 4th order spectral difference method. Values for Strouhal number $St = 0.164$ are consistent across the board of numerical results and match the experimental result from Williamson.

Side-by-side comparisons of reduction in $\|\mathbf{R}_\tau(1)\|_1$, $\|\mathbf{R}_\tau(2)\|_1$ and $\|\mathbf{R}_\tau(3)\|_1$ between RK3 and LU-SGS are plotted in Figs. 15, 16 and 17, respectively, during the physical time interval $340 \leq t \leq 440$. These plots display the level of pseudo residuals for all three governing equations before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{max}$) per physical iteration over multiple shedding cycles. Also, the number of pseudo iterations required for convergence per physical time step for LU-SGS is shown in Fig. 18. The flat line for RK3 in this plot demonstrates that the criterion for $\|\mathbf{R}_\tau\|_1$ is consistently not satisfied due to the inability of the explicit scheme to further improve the divergence of velocity field through rather restricted pseudo time stepping, which is evident in the overlapping lines in Fig. 15(a). This fact points to the stiffness associated with pressure. In Fig. 15(b), however, we see that the LU-SGS scheme is capable of reducing $\|\mathbf{R}_\tau(1)\|_1$, which is required to more accurately satisfy the continuity equation. This reduction is attributed to the larger pseudo time step afforded by the implicit scheme. Though, at the same time, some accuracy in $\|\mathbf{R}_\tau(2)\|_1$ and $\|\mathbf{R}_\tau(3)\|_1$ is

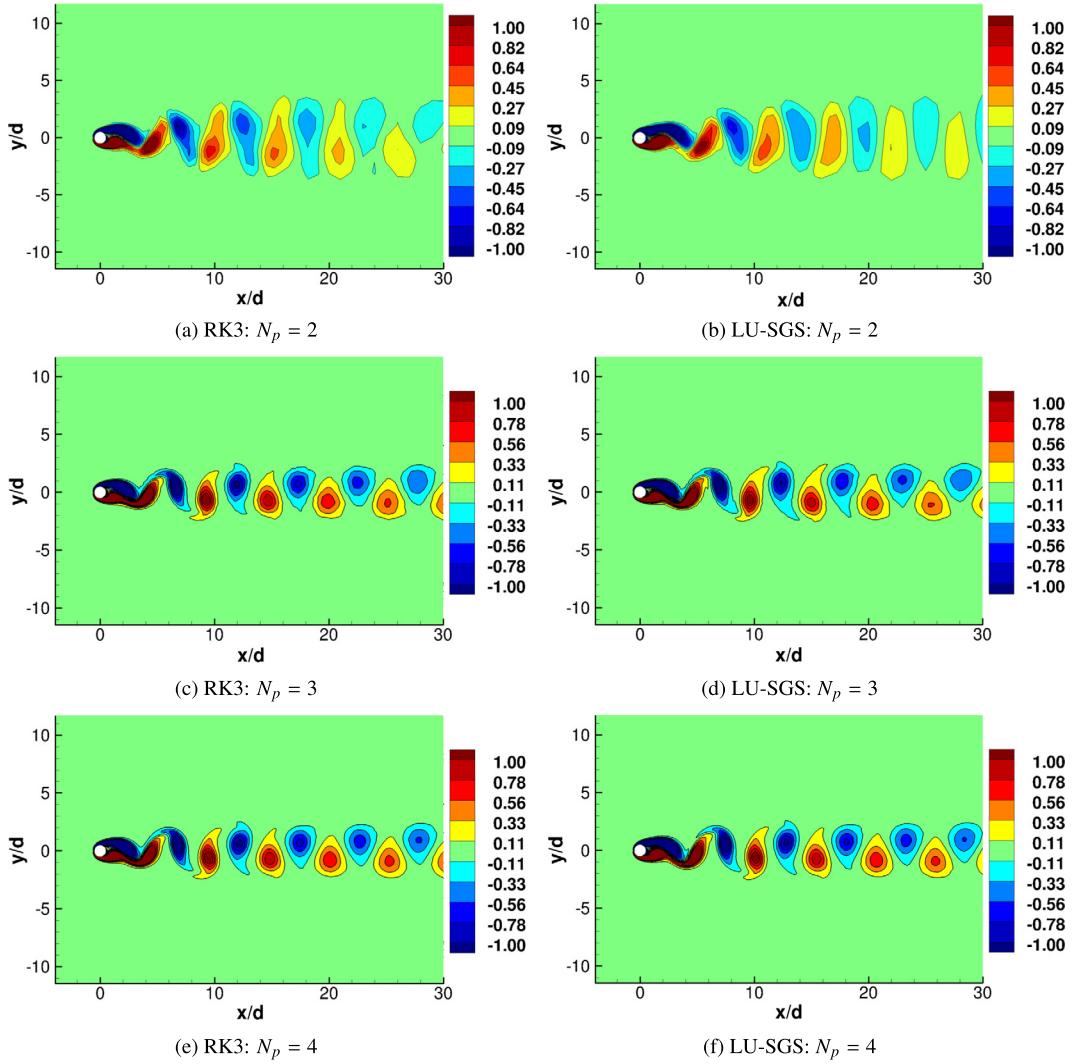


Fig. 13. Cylinder at $Re = 100$: vorticity.

given up under the assigned criterion, as illustrated in Figs. 16 and 17. To assess the effect of time step size on the number of pseudo iterations when using LU-SGS, we reduce the time step by an order of magnitude from $\Delta\tau^{LUSGS} = \Delta t^{LUSGS} = 0.1$ to $\Delta\tau^{LUSGS} = \Delta t^{LUSGS} = 0.01$, and plot maximum pseudo iterations per physical time step in Fig. 19. These results demonstrate that we require approximately $2.5\times$ more pseudo iterations to drop $\|\mathbf{R}_\tau\|_1$ by the same criterion of 10^3 using this smaller time step. So, to simulate the same amount of physical time, the smaller pseudo time stepping requires $25\times$ more CPU time. Therefore, we can conclude that the increased stability offered by large implicit pseudo time stepping improves the convergence rate.

Analysis of this unsteady laminar flow problem highlights that the explicit scheme is not as effective in satisfying the divergence-free constraint on the velocity field as the implicit scheme. The implicit scheme for pseudo time stepping is well suited for accurately solving unsteady incompressible flow using the artificial compressibility method. Within the context of the present high-order FR framework, the restrictions on both pseudo and physical time step sizes experienced by the combination of RK3 and BDF2 for dual time stepping can be effectively alleviated by an implicit backward Euler scheme with non-linear LU-SGS.

8. Conclusion

A high-order compact flux reconstruction method for solving unsteady incompressible flow on unstructured grids with implicit dual time stepping has been presented. The governing equations employ Chorin's classic artificial compressibility formulation such that advanced time stepping techniques developed for solving compressible flow can be used. We implement both a total variation diminishing Runge-Kutta scheme and a backward Euler with non-linear LU-SGS smoothing

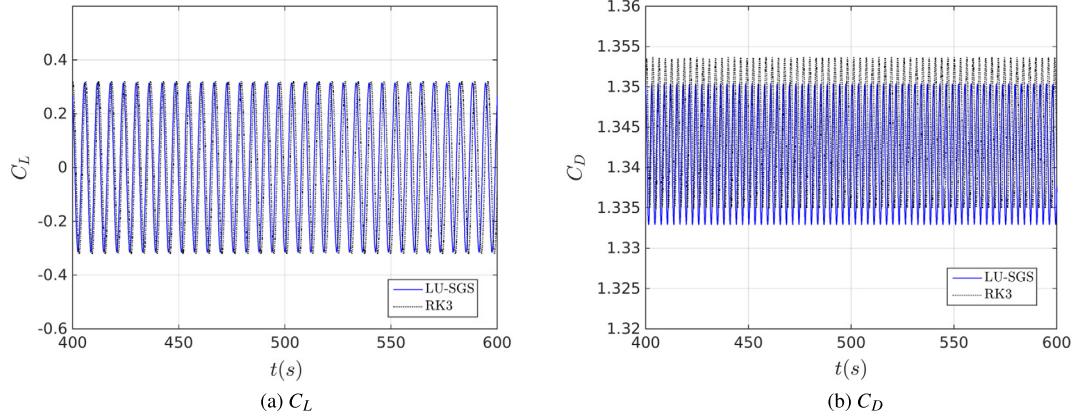


Fig. 14. Cylinder at $Re = 100$: C_L and C_D vs. physical time predicted by a 3rd order method.

Table 6

Cylinder at $Re = 100$: Strouhal numbers and force coefficients due to pressure $C_{(,),p}$ and viscosity $C_{(,),v}$ predicted by 3rd and 4th order methods. Root mean square values are signified by the superscript $(\cdot)^r$.

	Current				Park et al. [41]	Sharman et al. [42]	Posdziech et al. [45]
	RK3	LU-SGS	RK3	LU-SGS			
Blockage	2.5%	2.5%	2.5%	2.5%	1.0%	2.0%	0.71%
Elements	4371	4371	4371	4371	~39000	14441	~19000
N_p	3	3	4	4	—	—	—
St	0.164	0.164	0.164	0.164	0.165	0.164	0.164
C_L	± 0.320	± 0.316	± 0.331	± 0.325	± 0.332	± 0.325	—
C'_L	0.227	0.223	0.234	0.230	0.23	0.23	0.23
$C_{L,p}^r$	0.200	0.197	0.205	0.202	0.21	0.20	0.20
$C_{L,v}^r$	0.031	0.030	0.031	0.031	0.030	0.030	0.031
C_D	1.344	1.342	1.342	1.339	1.33	1.33	1.33
$C_{D,p}$	1.002	1.000	1.000	0.997	0.99	0.99	0.99
$C_{D,v}$	0.342	0.342	0.342	0.342	0.34	0.34	0.34
C'_D	0.0066	0.0062	0.0068	0.0063	0.0064	0.0064	0.0063
$C_{D,p}^r$	0.0060	0.0056	0.0064	0.0058	0.0058	0.0058	0.0058
$C_{D,v}^r$	7.13E-4	6.50E-4	7.50E-4	6.58E-4	7.00E-4	4.00E-4	7.00E-4

Table 7

Cylinder at $Re = 100$: comparison of St , C_L and C_D with various incompressible solvers.

	Method	St	C_L	C_D
Current (4th order w/ LU-SGS)	Artificial compressibility + flux reconstruction	0.164	± 0.325	1.339 ± 0.009
Park et al. [41]	Fractional step	0.165	± 0.332	1.33 ± 0.009
Sharman et al. [42]	SIMPLE	0.164	± 0.325	1.33 ± 0.009
Chan et al. [46] (4th order)	Compressible spectral difference	0.164	± 0.325	1.338 ± 0.009
Liu et al. [47]	Artificial compressibility + RANS	0.165	± 0.339	1.350 ± 0.012
Ding et al. [48]	Hybrid finite difference	0.164	± 0.280	1.325 ± 0.008
Ding et al. [49]	Hybrid finite difference	0.166	± 0.287	1.356 ± 0.010
Williamson [50] (experiment)	—	0.164	—	—

scheme to handle pseudo time and a second-order backward Euler scheme for physical time. Viscous fluxes across each interface are computed using the popular 2nd procedure of Bassi and Rebay. This method makes the computational stencil compact, requiring information from direct neighboring elements only and allowing for inversion of the implicit operator within the LU-SGS framework. Accuracy and convergence of this solver is shown by studying both steady and unsteady laminar flow problems. The current implicit dual time stepping scheme is proven effective in solving the unsteady incompressible Navier-Stokes equations in pseudo time in the context of a discontinuous high-order spatial method, which is limited by time step size restrictions in explicit schemes. The class of discontinuous high-order methods (e.g. DG, SD and FR/CPR) that were designed to solve the unsteady compressible Navier-Stokes equations can be extended to solve incompressible flow using the artificial compressibility formulation with dual time stepping presented in this paper. Furthermore, our incompressible solver is very suitable for parallel computing that uses straightforward domain decomposition for partitioning computational elements, which in turn is friendly towards iterative methods that can be used to update solutions within each element for higher order polynomials. Finally, this solver can be easily extended to handle moving and deforming grids, similar to the implementation in Cox et al. [51].

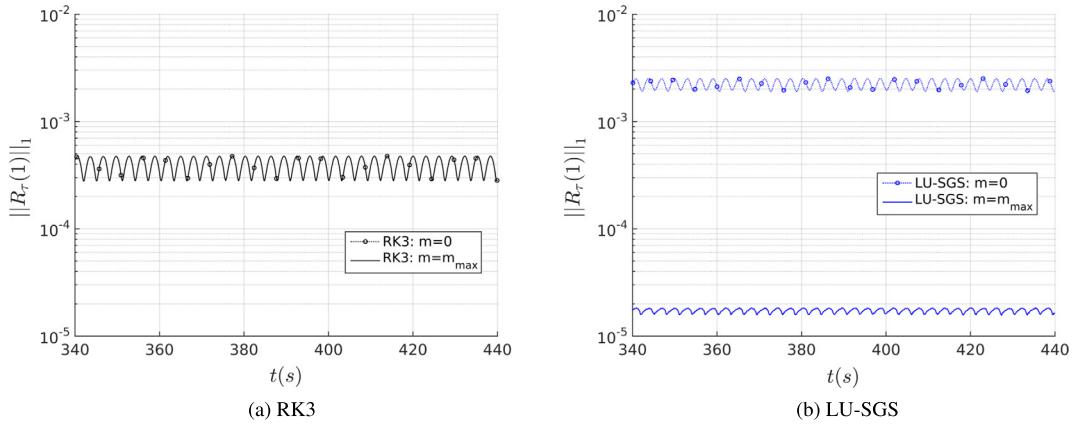


Fig. 15. Cylinder at $Re = 100$: L^1 -norm of continuity residual vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{\max}$) predicted by a 3rd order method.

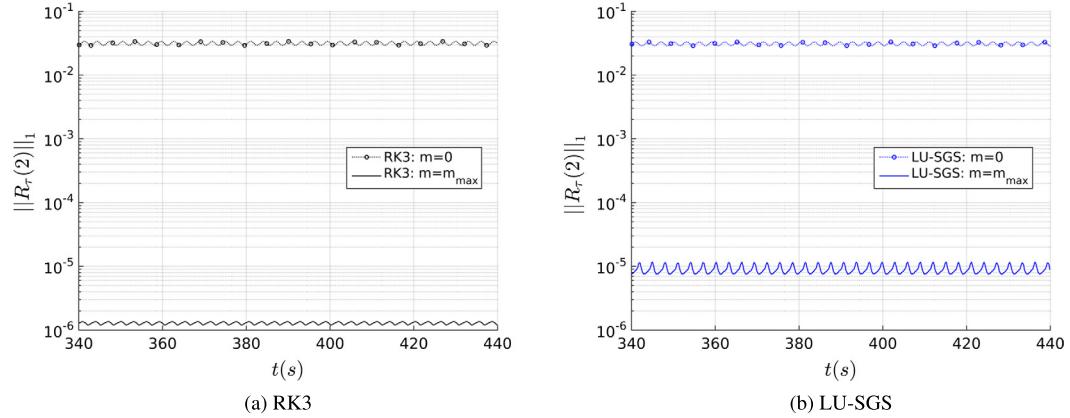


Fig. 16. Cylinder at $Re = 100$: L^1 -norm of x -momentum residual vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{\max}$) predicted by a 3rd order method.

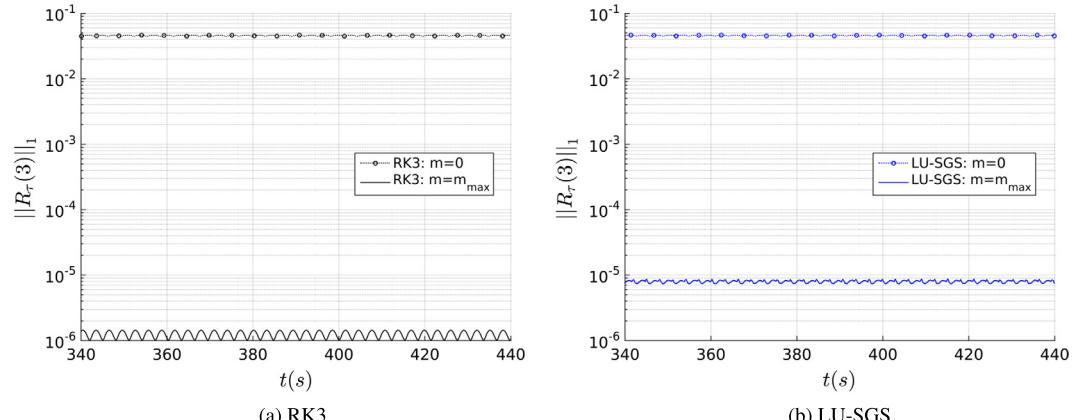


Fig. 17. Cylinder at $Re = 100$: L^1 -norm of y -momentum residual vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{\max}$) predicted by a 3rd order method.

Acknowledgement

The first author would like to greatly acknowledge The George Washington University for financial support under the GW Presidential Merit Fellowship. The second author acknowledges support from the Office of Naval Research Young Investigator Program (award number N000141410367 managed by Dr. Ki-Han Kim).

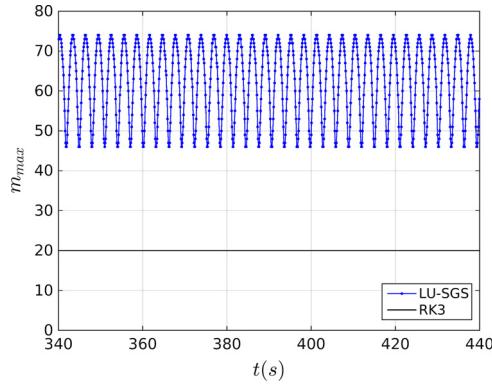


Fig. 18. Cylinder at $Re = 100$: total pseudo iterations m_{max} per physical time step predicted by a 3rd order method.

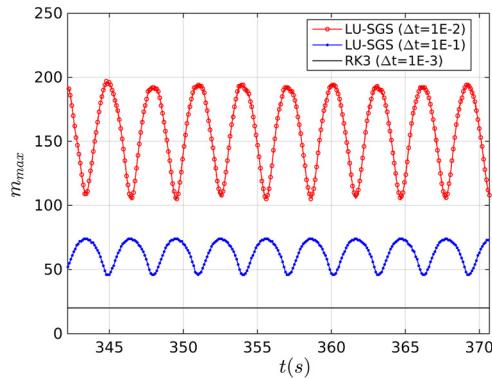


Fig. 19. Cylinder at $Re = 100$: total pseudo iterations m_{max} per physical time step using LU-SGS, highlighting increased iterations with order of magnitude reduction in time step size.

References

- [1] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, *Int. J. Numer. Methods Fluids* 72 (2013) 811–845.
- [2] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [3] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 143 (1998) 125–158.
- [4] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, *J. Comput. Phys.* 216 (2006) 780–801.
- [5] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, *J. Comput. Phys.* 194 (2) (2004) 716–741.
- [6] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper 2007-4079, 2007.
- [7] H.T. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, AIAA Paper 2009-403, 2009.
- [8] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, *J. Comput. Phys.* 228 (21) (2009) 8161–8186.
- [9] P.E. Vincent, P. Castonguay, A. Jameson, A new class of high-order energy stable flux reconstruction schemes, *J. Sci. Comput.* 47 (1) (2011) 50–72.
- [10] C. Liang, C. Cox, M.W. Plesniak, A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction, *J. Comput. Phys.* 239 (2013) 138–146.
- [11] H.T. Huynh, Z.J. Wang, P.E. Vincent, High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids, *Comput. Fluids* 98 (2014) 209–220.
- [12] Z.J. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, *Prog. Aerosp. Sci.* 43 (1–3) (2007) 1–41.
- [13] C. Liang, A.S. Chan, A. Jameson, A p -multigrid spectral difference method for incompressible Navier–Stokes equations, *Comput. Fluids* 51 (2011) 127–135.
- [14] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 135 (2) (1997) 118–125, reprinted from *J. Comput. Phys.* 2 (1) (1967) 12–26.
- [15] F. Bassi, A. Crivellini, D.A.D. Pietro, S. Rebay, An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations, *J. Comput. Phys.* 218 (2) (2006) 794–815.
- [16] K. Shahbazi, P.F. Fischer, C.R. Ethier, A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations, *J. Comput. Phys.* 222 (1) (2007) 391–407.
- [17] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, AIAA Paper 2010-362, 2010.
- [18] A. Jameson, S. Yoon, LU implicit schemes with multiple grids for the Euler equations, AIAA Paper 86-0105, 1986.
- [19] C. Liang, R. Kannan, Z.J. Wang, A p -multigrid spectral difference method with explicit and implicit smoothers on unstructured triangular grids, *Comput. Fluids* 38 (2009) 254–265.

- [20] Y. Sun, Z.J. Wang, Y. Liu, Efficient implicit non-linear LU-SGS approach for compressible flow computation using high-order spectral difference method, *Commun. Comput. Phys.* 5 (2–4) (2009) 760–778.
- [21] M. Parsani, G. Ghorbaniasl, C. Lacor, E. Turkel, An implicit high-order spectral difference approach for large eddy simulation, *J. Comput. Phys.* 229 (2010) 5373–5393.
- [22] T. Haga, K. Sawada, Z.J. Wang, An implicit LU-SGS scheme for the spectral volume method on unstructured tetrahedral grids, *Commun. Comput. Phys.* 6 (5) (2009) 978–996.
- [23] D.T. Elsworth, E.F. Toro, Riemann solvers for solving the incompressible Navier–Stokes equations using the artificial compressibility method, Tech. Rep. 9208, Cranfield Institute of Technology, June 1992.
- [24] S.E. Rogers, D. Kwak, Upwind-differencing scheme for the time-accurate incompressible Navier–Stokes equations, *AIAA J.* 28 (2) (1990) 253–262.
- [25] S.E. Rogers, D. Kwak, C. Kiris, Steady and unsteady solutions of the incompressible Navier–Stokes equations, *AIAA J.* 29 (4) (1991) 603–610.
- [26] C.L. Merkle, M. Athavale, Time accurate unsteady incompressible flow algorithms based on artificial compressibility, *AIAA Paper* 87-1137, 1987.
- [27] W.Y. Soh, J.W. Goodrich, Unsteady solution of incompressible Navier–Stokes equations, *J. Comput. Phys.* 79 (1) (1988) 113–134.
- [28] D.T. Elsworth, E.F. Toro, A numerical investigation of the artificial compressibility method for the solution of the Navier–Stokes equations, Tech. Rep. 9213, Cranfield Institute of Technology, November 1992.
- [29] H. Gao, Z.J. Wang, Y. Liu, A study of curved boundary representations for 2D high-order Euler solvers, *J. Sci. Comput.* 44 (2010) 323–336.
- [30] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Commun. Pure Appl. Math.* 7 (1954) 159–193.
- [31] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys.* 138 (1997) 251–285.
- [32] J. Peraire, P.-O. Persson, The compact discontinuous Galerkin method for elliptic problems, *SIAM J. Sci. Comput.* 30 (2009) 1806–1824.
- [33] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.* 35 (1998) 2440–2463.
- [34] F. Bassi, S. Rebay, A high-order discontinuous Galerkin method for compressible turbulent flows, in: G.K.B. Cockburn, C.-W. Shu (Eds.), *Discontinuous Galerkin Methods: Theory, Computation, and Application*, in: *Lecture Notes in Computational Science and Engineering*, vol. 11, Springer, 2000, pp. 77–88.
- [35] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (2) (1988) 439–471.
- [36] A. Jameson, Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings, *AIAA Paper* 1991-1596, 1991.
- [37] R. Kannan, Z.J. Wang, A study of viscous flux formulations for a p-multigrid spectral volume Navier–Stokes solver, *J. Sci. Comput.* 41 (2009) 165–199.
- [38] G.I. Taylor, Stability of a viscous liquid contained between two rotating cylinders, *Philos. Trans. R. Soc. Lond., Ser. A, Contain. Pap. Math. Phys. Character* 223 (1923) 289–343.
- [39] J. Bell, P. Colella, H. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [40] M. Minion, D. Brown, Performance of under-resolved two-dimensional incompressible flow simulations, II, *J. Comput. Phys.* 138 (1997) 734–765.
- [41] J. Park, K. Kwon, H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds number up to 160, *KSME Int. J.* 12 (6) (1998) 1200–1205.
- [42] B. Sharman, F.S. Lian, L. Davidson, C. Norberg, Numerical predictions of low Reynolds number flows over two tandem circular cylinders, *Int. J. Numer. Methods Fluids* 47 (2005) 423–447.
- [43] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, *J. Comput. Phys.* 113 (1) (1994) 1–4.
- [44] L.S. Caretto, A.D. Gosman, S.V. Patankar, D.B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation, in: *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, in: *Lecture Notes in Physics*, vol. 19, Springer, 1973, pp. 60–68.
- [45] O. Posdziech, R. Grundmann, Tech. rep., Technische Universität Dresden, Germany, 2000.
- [46] A. Chan, P. Dewey, A. Jameson, C. Liang, A. Smits, Vortex suppression and drag reduction in the wake of counter-rotating cylinders, *J. Fluid Mech.* 679 (2011) 343–382.
- [47] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1998) 35–57.
- [48] H. Ding, C. Shu, K.S. Yeo, D. Xu, Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method, *Comput. Methods Appl. Mech. Eng.* 193 (2004) 727–744.
- [49] H. Ding, C. Shu, K.S. Yeo, D. Xu, Numerical simulation of flows around two circular cylinders by mesh-free least square-based finite difference methods, *Int. J. Numer. Methods Fluids* 53 (2007) 305–332.
- [50] C.H.K. Williamson, Vortex dynamics in the cylinder wake, *Annu. Rev. Fluid Mech.* 28 (1996) 477–539.
- [51] C. Cox, C. Liang, M.W. Plesniak, Spectral difference solution of incompressible flow over an inline tube bundle with oscillating cylinder, in: *ASME Pressure Vessels and Piping*, 2012.