A flux reconstruction solver for unsteady incompressible viscous flow using artificial compressibility with implicit dual time stepping

Christopher Cox,[*] Chunlei Liang[†]
and Michael W. Plesniak[‡]

Department of Mechanical & Aerospace Engineering
The George Washington University, Washington, D.C., 20052, USA

This paper reports development of a high-order flux reconstruction method for solving unsteady incompressible flow on unstructured grids with implicit dual time stepping. The governing equations employ Chorin's classic artificial compressibility treatment with dual time stepping. Implicit non-linear lower-upper symmetric Gauss-Seidel smoothing with backward Euler discretization is used to march the solution in pseudo time, while a second-order backward Euler discretization is used to march in physical time. We verify implementation of the high-order method coupled with our implicit time stepping scheme using incompressible Taylor-Green decaying vortices. We further validate the solver with unsteady laminar flow past a cylinder. 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.

I. Introduction

In computational fluid dynamics, unstructured high-order methods, i.e. those considered 3^{rd} 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-refinement 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). With the development of these 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. In this paper, we implementat 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,[2] whereby they use a spectral difference method and Chorin's original artificial compressibility formulation (AC)[3] 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.

[*] Doctoral Candidate, AIAA Member
[†] Assistant Professor, AIAA Associate Fellow
[‡] Professor & Chairman, AIAA Fellow

In recent years, the lower-upper symmetric Gauss-Seidel (LU-SGS) scheme that was originally developed by Jameson and Yoon[4] has been used within the high-order CFD community for solving the compressible Navier-Stokes equations on unstructured grids with the SD[5,6,7] and SV[8] methods. However, when solving incompressible flows using artificial compressibility, the LU-SGS technique 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. Overall, this paper attempts 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.

II. Mathematical Formulation

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

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

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

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

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

$$\boldsymbol{f}_e = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \end{bmatrix}, \qquad \boldsymbol{g}_e = \begin{bmatrix} \beta v \\ uv \\ v^2 + p \end{bmatrix} \tag{3}$$

and the viscous fluxes are

$$\boldsymbol{f}_v = \begin{bmatrix} 0 \\ \nu \frac{\partial u}{\partial x} \\ \nu \frac{\partial v}{\partial x} \end{bmatrix}, \qquad \boldsymbol{g}_v = \begin{bmatrix} 0 \\ \nu \frac{\partial u}{\partial y} \\ \nu \frac{\partial v}{\partial y} \end{bmatrix}. \tag{4}$$

In the above formulation, ν 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 exists three characteristics in 2D that can be obtained from the eigensystem of the convective flux Jacobians $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}_e}{\partial \boldsymbol{U}} = \begin{bmatrix} 0 & \beta & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{bmatrix}, \qquad \boldsymbol{B} = \frac{\partial \boldsymbol{g}_e}{\partial \boldsymbol{U}} = \begin{bmatrix} 0 & 0 & \beta \\ 0 & v & u \\ 1 & 0 & 2v \end{bmatrix}. \tag{5}$$

The eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ 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.

II.A. 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

$$\Omega = \bigcup_{q=1}^{N_q} \Omega_q, \qquad \bigcap_{q=1}^{N_q} \Omega_q = \emptyset. \tag{6}$$

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

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

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 (ξ,η)=(0,0). After transformation into the computational domain, the governing equations in Eq. 1 can be re-written in the form

$$\frac{\partial \hat{\boldsymbol{U}}}{\partial \tau} + \boldsymbol{I}_D \frac{\partial \hat{\boldsymbol{U}}}{\partial t} + \hat{\nabla} \cdot \hat{\boldsymbol{F}} = 0 \tag{8}$$

where

$$\hat{\boldsymbol{U}} = |\boldsymbol{J}|\boldsymbol{U}, \qquad \begin{bmatrix} \hat{\boldsymbol{f}} \\ \hat{\boldsymbol{g}} \end{bmatrix} = |\boldsymbol{J}|\boldsymbol{J}^{-1} \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}. \tag{9}$$

In this form, $(\hat{\cdot})$ 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 $\boldsymbol{J}$ is independent of time and can be computed for each 2D reference element as

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{10}$$

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

$$\frac{\partial \hat{\boldsymbol{U}}}{\partial \tau} + \hat{\boldsymbol{S}} + \hat{\boldsymbol{R}} = 0 \tag{11}$$

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

II.B. 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. Figure 1 shows the distribution of solution and flux points for a 4^{th} order flux reconstruction method inside a reference element. In this arrangement, the number of solution points in each direction is $N_p = 4$.

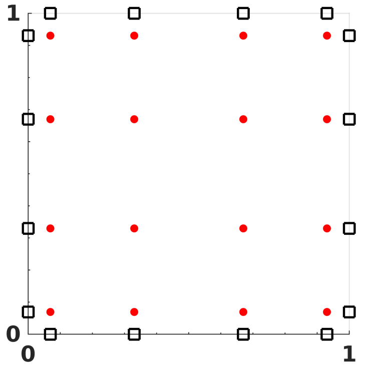


Figure 1: Distribution of solution points (SP(●)) and flux points (FP(□)) for a 4^{th} order FR method inside a reference element Ω_r.

According to Huynh,[9] reconstruction for each element involves the jump in flux at each interface. Using this disparity, corrections need to be made to a flux function that is discontinuous in order to define a flux function that is continuous. Following the initial 1D formulation presented by Huynh, let the polynomial $\hat{\boldsymbol{f}}_r$ 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 $\hat{\boldsymbol{f}}_r^D$ is then corrected in the following manner

$$\hat{\boldsymbol{f}}_r(\xi) = \hat{\boldsymbol{f}}_r^D(\xi) + [\hat{\boldsymbol{f}}^{com}_{r-\frac{1}{2}} - \hat{\boldsymbol{f}}_r^D(0)]g_r^{LB}(\xi) + [\hat{\boldsymbol{f}}^{com}_{r+\frac{1}{2}} - \hat{\boldsymbol{f}}_r^D(1)]g_r^{RB}(\xi). \tag{12}$$

where $(\cdot)^{RB}$ and $(\cdot)^{LB}$ signify quantities at right and left boundary interfaces of Ω_r. This formulation provides two corrections 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=$\{\xi, \eta | 0 \leq \xi, \eta \leq 1\}$ is $g_r = g_{GA}$ from Huynh.[9] As mentioned previously, the jump in flux at the interface results in a correction to $\hat{\boldsymbol{f}}_r^D$ evaluated at all solution points. For the left boundary, the correction function in 1D is defined as

$$g_{GA}^{LB}(\xi) = \frac{N_p}{2N_p - 1} R_{R,N_p}(\xi) + \frac{N_p - 1}{2N_p - 1} R_{R,N_p-1}(\xi) \tag{13}$$

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

$$R_{R,N_p}(\xi) = \frac{(-1)^{N_p}}{2}(P_{N_p}(\xi) - P_{N_p-1}(\xi)) \tag{14}$$

and $P_{N_p}(\xi)$ is the Legendre polynomial. The expression for correction to the right boundary $g_{GA}^{RB}(\xi)$ is obtained simply by reflection of $g_{GA}^{LB}(\xi)$. The correction functions g_r for left and right boundaries must take on the values

$$g_r^{LB}(0) = 1, \qquad g_r^{LB}(1) = 0 \tag{15}$$

$$g_r^{RB}(0) = 0, \qquad g_r^{RB}(1) = 1. \tag{16}$$

To construct a $N_p - 1$ degree polynomial in each coordinate direction, the solution at N_p points is required (see Fig. 1). Within each dimension, the solution points are 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

$$P_{N_p}(\xi) = \frac{2N_p - 1}{N_p}(2\xi - 1)P_{N_p-1}(\xi) - \frac{N_p - 1}{N_p} P_{N_p-2}(\xi). \tag{17}$$

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

$$l_i(\xi) = \prod_{s=1, s \neq i}^{N_p} \left(\frac{\xi - \xi_s}{\xi_i - \xi_s} \right) \qquad \forall \;\; i = 1, ..., N_p \tag{18}$$

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

$$\hat{\boldsymbol{U}}_r(\xi,\eta) = \sum_{j=1}^{N_p}\sum_{i=1}^{N_p} \hat{\boldsymbol{U}}_{r|i,j} l_i(\xi)\cdot l_j(\eta) \tag{19}$$

$$\hat{\boldsymbol{f}}^D_r(\xi,\eta) = \sum_{j=1}^{N_p}\sum_{i=1}^{N_p} \hat{\boldsymbol{f}}^D_{r|i,j} l_i(\xi)\cdot l_j(\eta). \tag{20}$$

The discontinuous flux function $\hat{\boldsymbol{f}}^D_r$ is of degree $N_p - 1$. Both g^{LB}_r and g^{RB}_r are of degree N_p. By reflection, $g^{RB}_r(\xi) = g^{LB}_r(1-\xi)$ on the interval $\{\xi | 0 \leq \xi \leq 1\}$. Therefore, the function $\hat{\boldsymbol{f}}_r$ is of degree N_p and takes on the two common flux values at the interface

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

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

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

$$\frac{\partial \hat{\boldsymbol{f}}_r}{\partial \xi} = \frac{\partial \hat{\boldsymbol{f}}^D_r}{\partial \xi} + [\hat{\boldsymbol{f}}^{com}_{r-\frac{1}{2}} - \hat{\boldsymbol{f}}^D_r(0)]\frac{dg^{LB}_r}{d\xi} + [\hat{\boldsymbol{f}}^{com}_{r+\frac{1}{2}} - \hat{\boldsymbol{f}}^D_r(1)]\frac{dg^{RB}_r}{d\xi}. \tag{23}$$

A derivative of the discontinuous flux function $\partial\hat{\boldsymbol{f}}^D_r/\partial\xi$ can be computed from a Lagrange polynomial or a chain-rule. The former method is used in the current implementation.

Now, since the mathematical nature of the governing equations has been modified due to the introduction of artificial compressibility, common inviscid fluxes at an interface $\hat{\boldsymbol{f}}^{com}$ can be computed using an approximate Riemann solver. Here, we employ the simple but robust local Lax-Friedrichs (LLF) flux[10]

$$\hat{\boldsymbol{f}}^{com} = \frac{1}{2}[\boldsymbol{f}^R_n + \boldsymbol{f}^L_n - \lambda_{max}(\boldsymbol{U}^R - \boldsymbol{U}^L)] \tag{24}$$

where the convective flux normal to an interface is $\boldsymbol{f}_n = \boldsymbol{f}_e n_x + \boldsymbol{g}_e n_y$ 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 $\lambda_{max} = \max_k \mid \lambda^k \mid = \mid \boldsymbol{V}_n \mid + c_n$ falls out of an eigensystem analysis of the convective flux Jacobian $\boldsymbol{A}_n$, where

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

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

In order to gaurantee compactness of the stencil in multiple dimensions, the popular BR2 scheme is employed to compute the common viscous fluxes at the interface. To compute these terms, the derivative of $\hat{\boldsymbol{U}}_r$ is needed to evaluate the derivative of the discontinuous flux function $\hat{\boldsymbol{f}}^D_r$. Following the formulation in Huynh[9] and connecting it with the BR2 scheme, the gradient of the common solution is

$$\frac{\partial \hat{\boldsymbol{U}}^{com}_{r+\frac{1}{2}}}{\partial \xi} = \frac{1}{2}\left\{\frac{\partial \hat{\boldsymbol{U}}_r(1)}{\partial \xi} + \frac{\partial \hat{\boldsymbol{U}}_{r+1}(0)}{\partial \xi} + \left[\hat{\boldsymbol{U}}^{com}_{r+\frac{1}{2}} - \hat{\boldsymbol{U}}_r(1)\right]\frac{dg^{RB}_r(1)}{d\xi} + \left[\hat{\boldsymbol{U}}^{com}_{r+\frac{1}{2}} - \hat{\boldsymbol{U}}_{r+1}(0)\right]\frac{dg^{LB}_{r+1}(0)}{d\xi}\right\}. \tag{26}$$

Note that Eq. (26) requires data from elements r and $r+1$ only. As a result, the computational stencil in 2D involves five elements. This stencil lends itself towards a more straightforward implementation on massively parallel computers.

II.C. Implicit Time Stepping

In order to relax the restriction on time step size and drive the pseudo residual $\hat{\boldsymbol{R}}_\tau$ closer and faster to zero to achieve steady state, an implicit time stepping scheme is employed. The cost associated with the solution of coupled non-linear equations at each time step motivates linearization within 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

$$\frac{\partial \hat{\boldsymbol{U}}_r}{\partial \tau} + \hat{\boldsymbol{S}}_r + \hat{\boldsymbol{R}}_r = 0. \tag{27}$$

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

$$\frac{\hat{\boldsymbol{U}}_r^{n+1,m+1} - \hat{\boldsymbol{U}}_r^{n+1,m}}{\Delta\tau} + \hat{\boldsymbol{S}}_r^{n+1,m+1} + \hat{\boldsymbol{R}}_r^{n+1,m+1} = 0. \tag{28}$$

We can re-write Eq. (28) as

$$\frac{\hat{\boldsymbol{U}}_r^{n+1,m+1} - \hat{\boldsymbol{U}}_r^{n+1,m}}{\Delta\tau} + \hat{\boldsymbol{S}}_r^{n+1,m+1} - \hat{\boldsymbol{S}}_r^{n+1,m} + \hat{\boldsymbol{R}}_r^{n+1,m+1} - \hat{\boldsymbol{R}}_r^{n+1,m} = -\hat{\boldsymbol{R}}_r^{n+1,m} - \hat{\boldsymbol{S}}_r^{n+1,m} \tag{29}$$

where a second-order backward difference scheme (BDF2) is used for the physical time derivative. At time level $m+1$ this term is

$$\hat{\boldsymbol{S}}_r^{n+1,m+1} = \frac{\boldsymbol{I}_D}{2\Delta t}\left(3\hat{\boldsymbol{U}}_r^{n+1,m+1} - 4\hat{\boldsymbol{U}}_r^{n} + \hat{\boldsymbol{U}}_r^{n-1}\right). \tag{30}$$

Gathering terms

$$\left(\frac{\boldsymbol{I}}{\Delta\tau} + \frac{3\boldsymbol{I}_D}{2\Delta t}\right)\left(\hat{\boldsymbol{U}}_r^{n+1,m+1} - \hat{\boldsymbol{U}}_r^{n+1,m}\right) + \hat{\boldsymbol{R}}_r^{n+1,m+1} - \hat{\boldsymbol{R}}_r^{n+1,m} = -\hat{\boldsymbol{R}}_r^{n+1,m} - \hat{\boldsymbol{S}}_r^{n+1,m} \tag{31}$$

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

$$\hat{\boldsymbol{R}}_r^{m+1} - \hat{\boldsymbol{R}}_r^{m} \approx \frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_r}\Delta\hat{\boldsymbol{U}}_r + \sum_{nb\neq r}\frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_{nb}}\Delta\hat{\boldsymbol{U}}_{nb} \tag{32}$$

the following expression is obtained

$$\left(\frac{\boldsymbol{I}}{\Delta\tau} + \frac{3\boldsymbol{I}_D}{2\Delta t} + \frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_r}\right)\Delta\hat{\boldsymbol{U}}_r^{m+1} = -\hat{\boldsymbol{R}}_r^{m} - \hat{\boldsymbol{S}}_r^{m} - \sum_{nb\neq r}\frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_{nb}}\Delta\hat{\boldsymbol{U}}_{nb}^{m+1}. \tag{33}$$

To avoid storage and computation of $\partial\hat{\boldsymbol{R}}_r/\partial\hat{\boldsymbol{U}}_{nb}$ we can perform another linearization and apply the LU-SGS algorithm by replacing m with k, where k represents one forward/backward sweep through the grid, while introducing $(\star)$ to signify the most recently computed solution

$$\hat{\boldsymbol{R}}_r^{\star} - \hat{\boldsymbol{R}}_r^{k} \approx \frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_r}\Delta\hat{\boldsymbol{U}}_r^{\star} + \sum_{nb\neq r}\frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_{nb}}\Delta\hat{\boldsymbol{U}}_{nb}^{\star}. \tag{34}$$

Thus, we can obtain the final form

$$\left(\frac{\boldsymbol{I}}{\Delta\tau} + \frac{3\boldsymbol{I}_D}{2\Delta t} + \frac{\partial \hat{\boldsymbol{R}}_r}{\partial \hat{\boldsymbol{U}}_r}\right)\delta\hat{\boldsymbol{U}}_r^{k+1} = -\hat{\boldsymbol{R}}_r(\hat{\boldsymbol{U}}_r^{\star}, \hat{\boldsymbol{U}}_{nb}^{\star}) - \hat{\boldsymbol{S}}_r(\hat{\boldsymbol{U}}_r^{m}, \hat{\boldsymbol{U}}_r^{n}, \hat{\boldsymbol{U}}_r^{n-1}) - \left(\frac{\boldsymbol{I}}{\Delta\tau} + \frac{3\boldsymbol{I}_D}{2\Delta t}\right)\Delta\hat{\boldsymbol{U}}_r^{\star} \tag{35}$$

where

$$\delta\hat{\boldsymbol{U}}_r^{k+1} = \Delta\hat{\boldsymbol{U}}_r^{k+1} - \Delta\hat{\boldsymbol{U}}_r^{\star} \tag{36}$$

and

$$\Delta\hat{\boldsymbol{U}}_r^{k+1} = \hat{\boldsymbol{U}}_r^{k+1} - \hat{\boldsymbol{U}}_r^{m}, \qquad \Delta\hat{\boldsymbol{U}}_r^{\star} = \hat{\boldsymbol{U}}_r^{\star} - \hat{\boldsymbol{U}}_r^{m}. \tag{37}$$

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

$$\hat{\boldsymbol{U}}_r^{k+1} = \hat{\boldsymbol{U}}_r^{\star} + \delta\hat{\boldsymbol{U}}_r^{k+1}. \tag{38}$$

After a maximum number of SGS-type 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 $\hat{\boldsymbol{U}}_r$ takes physical meaning. It is important to note the size of the element matrix $\boldsymbol{M}_r$, which consists of terms inside the brackets on the left-hand side of Eq. (35). The size of this matrix is shown in Eq. 39 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

$$Size\ of\ \boldsymbol{M}_r = [N_{eq} \cdot N_p^D] \times [N_{eq} \cdot N_p^D]. \tag{39}$$

Table 1 shows the size of $\boldsymbol{M}_r$ 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.

N_p	$\boldsymbol{M}_r$
2	12x12
3	27x27
4	48x48
5	75x75

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

The element Jacobian $\partial\hat{\boldsymbol{R}}_r/\partial\hat{\boldsymbol{U}}_r$ is computed numerically[6] at every physical iteration n by applying a perturbation ϵ to each state variable,

$$\left.\frac{\partial\hat{\boldsymbol{R}}_r}{\partial\hat{\boldsymbol{U}}_r}\right|_n = \frac{\hat{\boldsymbol{R}}_r(\hat{\boldsymbol{U}}_r^n + \epsilon, \hat{\boldsymbol{U}}_{nb}^n) - \hat{\boldsymbol{R}}_r(\hat{\boldsymbol{U}}_r^n, \hat{\boldsymbol{U}}_{nb}^n)}{\epsilon} \tag{40}$$

where ϵ is typically $\sqrt{\epsilon_{machine}}$. To improve convergence rates, however, the computation of $\partial\hat{\boldsymbol{R}}_r/\partial\hat{\boldsymbol{U}}_r$ can be frozen every 2-5 physical iterations without degrading the unsteady solution.

III. Verification & Validation

III.A. Taylor-Green Decaying Vortices

The Taylor decaying vortex problem is commonly used for verification of a numerical method in simulating unsteady incompressible flow. We solve the unsteady Navier-Stokes equations on a square domain $\{x, y | 0 < x, y < 2\pi\}$ at $Re = 10$. We implement periodic Dirichlet boundary conditions, and the initial condition is set to the exact solution

$$\tilde{p}(x,y,t) = -\frac{cos(2qx) + cos(2qy)}{4} exp(\frac{-4q^2t}{Re}) \tag{41}$$

$$\tilde{u}(x,y,t) = -cos(qx)sin(qy)exp(\frac{-2q^2t}{Re}) \tag{42}$$

$$\tilde{v}(x,y,t) = sin(qx)cos(qy)exp(\frac{-2q^2t}{Re}) \tag{43}$$

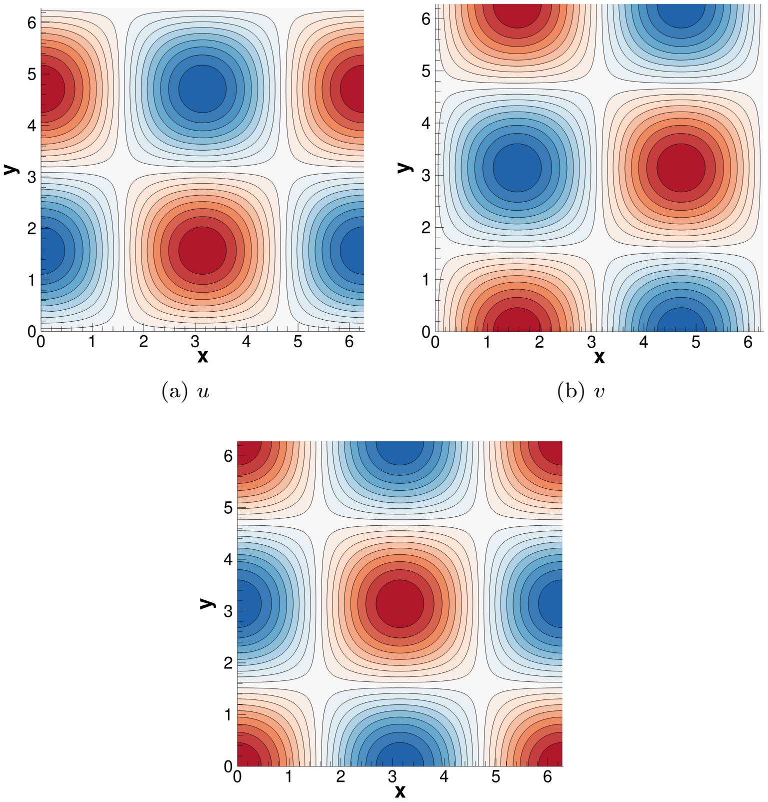
where $q = 1$ is the Taylor array number. Refinement is done uniformly from the coarsest grid Ω^1 down to the finest grid Ω^4. The L^p norm of u-velocity error $||\epsilon||_p$ for $N_p = \{2, 3, 4\}$ is computed by Eq. (44)

$$||\epsilon||_p = \left(\frac{\sum_{k=1}^{DOF} |\epsilon_k|^p}{DOF}\right)^{\frac{1}{p}} \tag{44}$$

where $\epsilon = \tilde{u} - u$ and u is obtained numerically. The 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. (45)

$$\mathcal{P}_{L^p} = \frac{\log\left[\frac{||\epsilon||_p(\Omega^d)}{||\epsilon||_p(\Omega^{d+1})}\right]}{\log\left[\frac{h(\Omega^d)}{h(\Omega^{d+1})}\right]} \tag{45}$$

where $p = \{1, 2\}$ to represent different norms and $h(\Omega^d) = 1/\sqrt[D]{DOF}$ for $\Omega^d \subset \mathbb{R}^D$. For this benchmark test case, $d = \{1, 2, 3, 4\}$ to represent four uniformly refined domains with number of elements 16, 64, 256 and 1024, respectively. For this two-dimensional benchmark test, $D = 2$. The velocity and vorticity fields obtained numerically with a 4^{th} order method on Ω^3 is shown in Fig. 2 and reductions in L^2-norm of the error are tabulated in Tab. 2 for $t = 2s$. A small constant time step of $\Delta t = 10^{-4}$ was chosen to ensure that spatial accuracy did not degrade due to errors resulting from time discretization. The maximum number of pseudo iterations was set to $m_{max} = 25$.



(a) u

(b) v

(c) Vorticity

Figure 2: Incompressible Taylor-Green vortex at $Re = 10$ for 4^{th} order method on Ω^3.

Domain	*Elements*	*DOF*	$\|\|\epsilon\|\|_2$	$\mathcal{P}_{L^2}$
Ω^1	16	64	2.26E-1	-
Ω^2	64	256	5.28E-2	2.09
Ω^3	256	1024	1.25E-2	2.08
Ω^4	1024	4096	2.90E-3	2.09

(a) $N_p = 2$

Domain	*Elements*	*DOF*	$\|\|\epsilon\|\|_2$	$\mathcal{P}_{L^2}$
Ω^1	16	64	2.67E-2	-
Ω^2	64	256	3.02E-3	3.15
Ω^3	256	1024	3.42E-4	3.14
Ω^4	1024	4096	3.81E-5	3.16

(b) $N_p = 3$

Domain	*Elements*	*DOF*	$\|\|\epsilon\|\|_2$	$\mathcal{P}_{L^2}$
Ω^1	16	64	2.13E-3	-
Ω^2	64	256	1.43E-4	3.90
Ω^3	256	1024	9.74E-6	3.87
Ω^4	1024	4096	6.45E-7	3.92

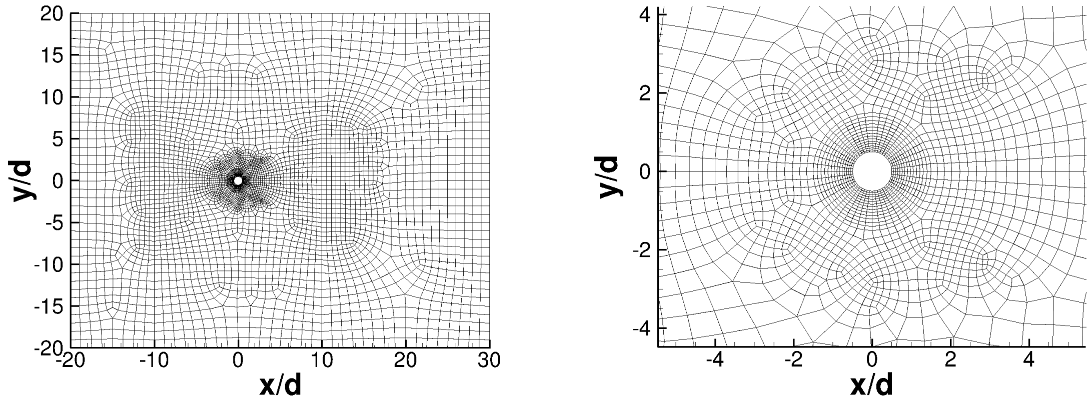
(c) $N_p = 4$

Table 2: Incompressible Taylor-Green vortices: order of accuracy obtained from L^2-norm of errors during uniform grid refinement.

III.B. Cylinder

We report numerical results for unsteady laminar flow at $Re = 100$ past a stationary cylinder. The grid used for this test case shown in Fig. 3 contains 4371 linear quadrilateral elements with 48 elements along the wall of the cylinder, which we represent by cubic Bezier curves. The blockage in y caused by the presence of the cylinder in the $40x50$ domain is 2.5%. Freestream velocity is set at the inlet and pressure is set at the outlet. We run 2^{nd}, 3^{rd} and 4^{th} order flux reconstruction methods. The pseudo time step taken by the implicit scheme is $100x$ larger than that of the explicit scheme. Increased stability offered by this non-linear LU-SGS implementation can also permit a $100x$ larger physical time step, while still capturing the transient process.

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 steady state has been reached. The current criterion for pseudo steady state convergence requires that $||\boldsymbol{R}_\tau||_1$ drop by a factor of 10^2. 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 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, $\partial\hat{\boldsymbol{R}}_r/\partial\hat{\boldsymbol{U}}_r$ is computed at every single physical iteration.



(a) full

(b) zoom

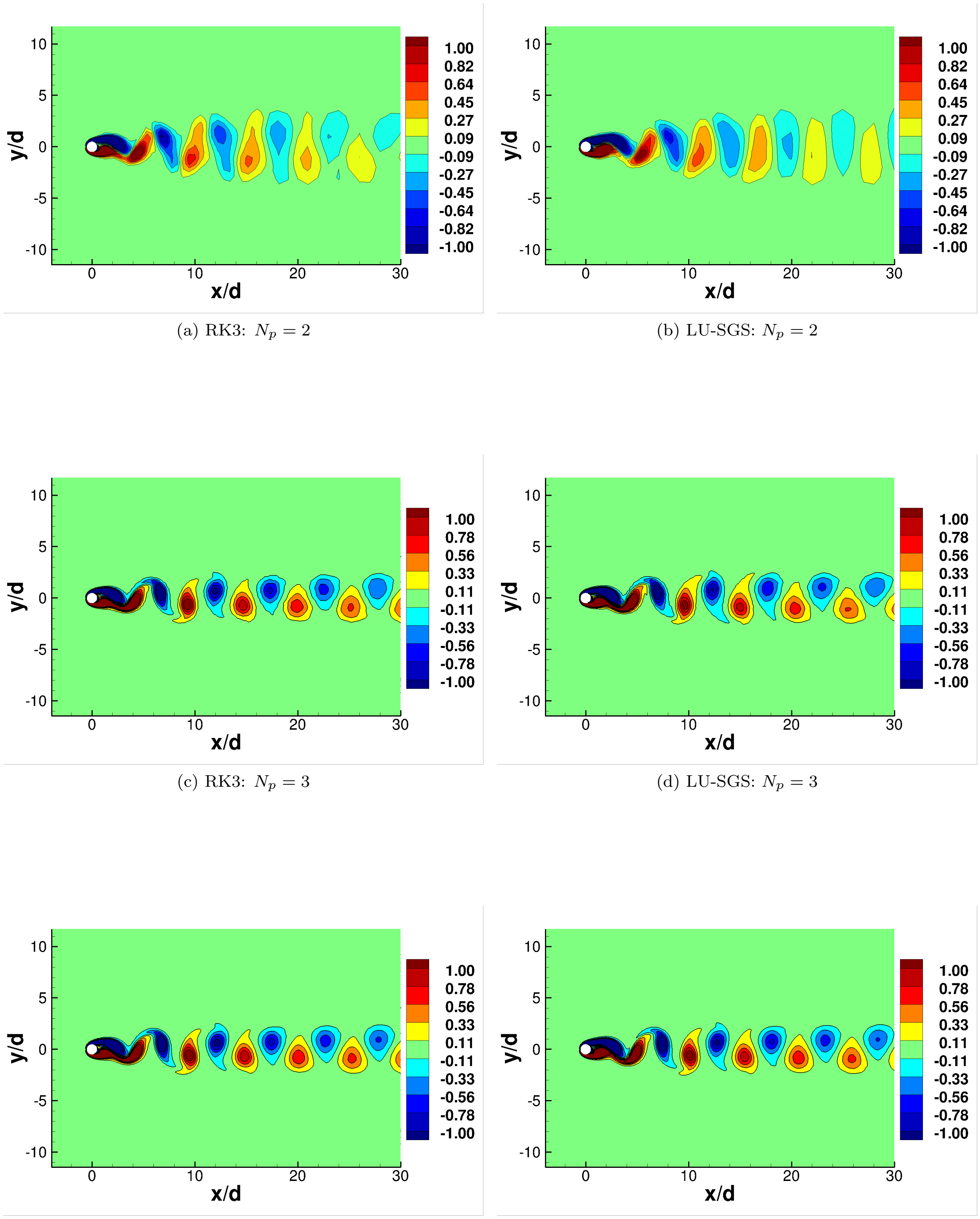
Figure 3: Cylinder grid.

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

Plots of force coefficients C_L and C_D generated by a 3^{rd} order method are shown in Fig. 5 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 Tab. 3 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[11] and Sharman et al.[12] The former authors employed the fractional step method of Choi and Moin[13] while the latter authors utilized the SIMPLE algorithm.[14] Root mean square (RMS) values are indicated by $(\cdot)'$. For a 4^{th} order FR method and implicit pseudo time stepping, 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%.

Side-by-side comparisons of reduction in $||\partial p/\partial \tau||_1$, $||\partial u/\partial \tau||_1$ and $||\partial v/\partial \tau||_1$ between RK3 and LU-SGS are plotted in Figs. 7, 8 and 9, respectively, during the physical time interval $340 \leq t \leq 440$. These plots display the level of pseudo residuals for all three 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 using LU-SGS is shown in Fig. 6. The flat line for RK3 in this plot demonstrates that the criterion for $||\boldsymbol{R}_\tau||_1$ 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. 7a. This fact points to the stiffness associated with pressure. In Fig. 7b, however, we see that the LU-SGS technique is capable of reducing $||\partial p/\partial \tau||_1$, 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 $||\partial u/\partial \tau||_1$ and $||\partial v/\partial \tau||_1$ is given up, as illustrated in Figs. 8 and 9.

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. Within the context of the present high-order flux reconstruction 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 implicit pseudo time stepping with LU-SGS.



(a) RK3: $N_p = 2$

(b) LU-SGS: $N_p = 2$

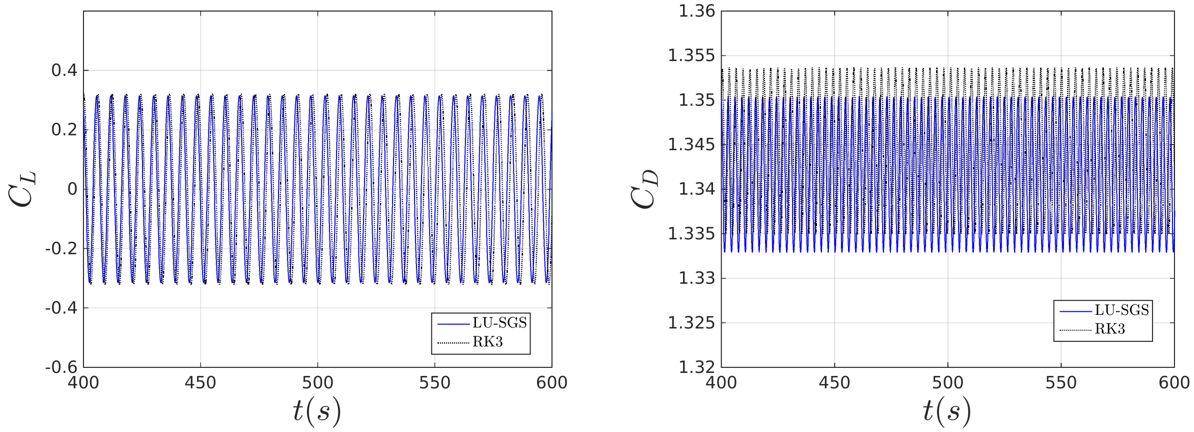
(c) RK3: $N_p = 3$

(d) LU-SGS: $N_p = 3$

(e) RK3: $N_p = 4$

(f) LU-SGS: $N_p = 4$

Figure 4: Cylinder at $Re = 100$: vorticity.



(a) C_L

(b) C_D

Figure 5: Cylinder at $Re = 100$: C_L and C_D vs. physical time predicted by a 3^{rd} order method.

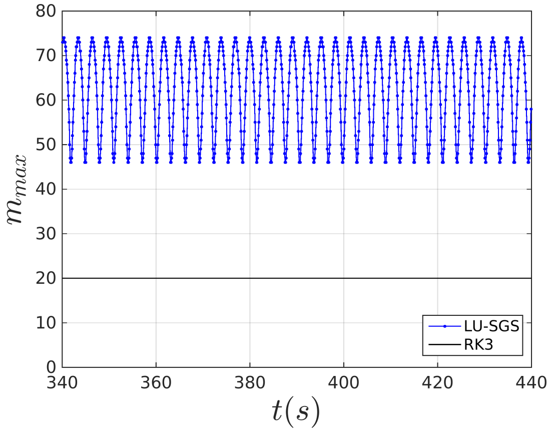
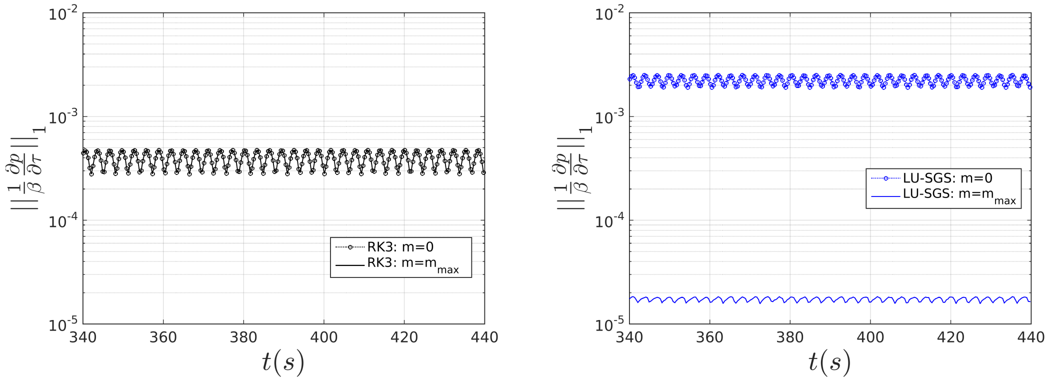


Figure 6: Cylinder at $Re = 100$: total pseudo iterations m_{max} vs. physical time predicted by a 3^{rd} order method.

	Current				Park et al.[11]	Sharman et al.[12]
	RK3	LU-SGS	RK3	LU-SGS		
Blockage	2.5%	2.5%	2.5%	2.5%	1.0%	2.0%
Elements	4371	4371	4371	4371	153600	14441
N_p	3	3	4	4	-	-
St	0.164	0.164	0.164	0.164	0.165	0.164
C'_L	0.227	0.223	0.234	0.230	0.23	0.23
$C'_{L,p}$	0.200	0.197	0.205	0.202	0.21	0.20
$C'_{L,v}$	0.031	0.030	0.031	0.031	0.03	0.03
C_D	1.344	1.342	1.342	1.339	1.33	1.33
$C_{D,p}$	1.002	1.000	1.000	0.997	0.99	0.99
$C_{D,v}$	0.342	0.342	0.342	0.342	0.34	0.34
$C'_{D,p}$	0.0060	0.0056	0.0064	0.0058	0.0058	0.0058
$C'_{D,v}$	7.13E-4	6.50E-4	7.50E-4	6.58E-4	7.00E-4	4.00E-4
C'_D	0.0066	0.0062	0.0068	0.0063	0.0064	0.0064

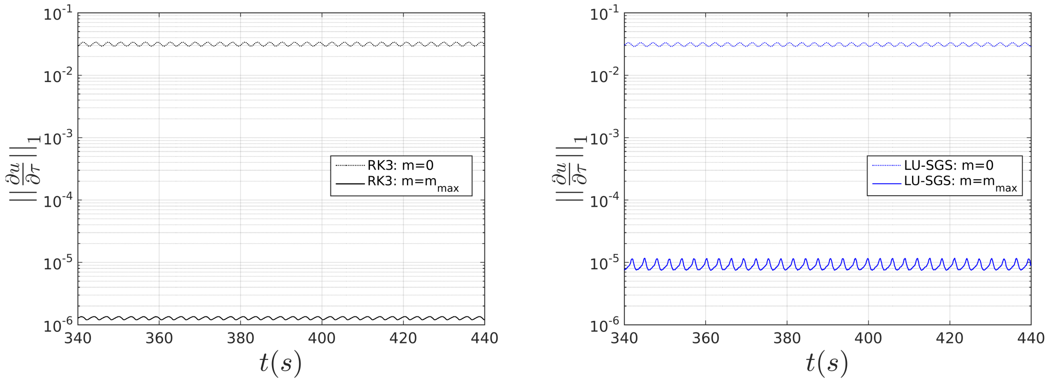
Table 3: Cylinder at $Re = 100$: Strouhal numbers and force coefficients due to pressure $C_{(\cdot),p}$ and viscosity $C_{(\cdot),v}$ predicted by 3^{rd} and 4^{th} order methods. Root mean square results are signified by the superscript $(\cdot)'$.



(a) RK3

(b) LU-SGS

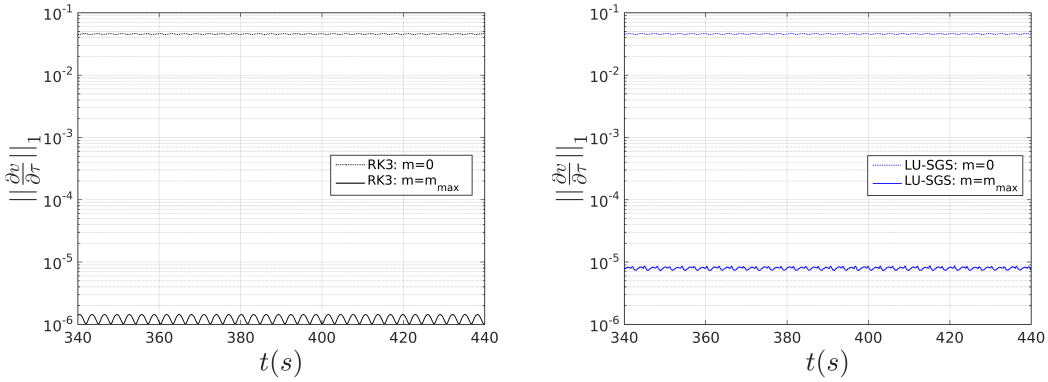
Figure 7: Cylinder at $Re = 100$: L^1-norm of $\partial p/\partial\tau$ vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{max}$) predicted by a 3^{rd} order method.



(a) RK3

(b) LU-SGS

Figure 8: Cylinder at $Re = 100$: L^1-norm of $\partial u/\partial\tau$ vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{max}$) predicted by a 3^{rd} order method.



(a) RK3

(b) LU-SGS

Figure 9: Cylinder at $Re = 100$: L^1-norm of $\partial v/\partial\tau$ vs. physical time before pseudo iterations begin ($m = 0$) and after pseudo iterations end ($m = m_{max}$) predicted by a 3^{rd} order method.

IV. 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 treatment 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 non-linear implicit LU-SGS with backward Euler scheme to deal with pseudo time and a second-order backward Euler scheme for physical time. Implementation of the flux reconstruction method with implicit dual time stepping is verified by the incompressible Taylor-Green vortex problem. Further validation involved unsteady flow past a cylinder. This test case requires that a criterion be provided for the pseudo residual such that the divergence-free condition of velocity can be more accurately satisfied at every physical time level. The Strouhal number and various force coefficients due to pressure and viscosity were computed then compared to two established incompressible solvers. Excellent agreement in lift and drag among these various solvers was demonstrated. Therefore, we conclude that LU-SGS with backward Euler is well suited for iterations within the pseudo time domain in the context of Huynh's high-order flux reconstruction method.

References

[1] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., v. Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 00, 2012, pp. 1–42.

[2] Liang, C., Chan, A. S., and Jameson, A., "A p-multigrid spectral difference method for incompressible Navier-Stokes equations," *Computers and Fluids*, Vol. 51, 2011, pp. 127–135.

[3] Chorin, A. J., "A numerical method for solving incompressible viscous flow problems," *Journal of Computational Physics*, Vol. 135, No. 2, 1997, pp. 118–125, reprinted from Vol. 2(1), 1967, 12-26.

[4] Jameson, A. and Yoon, S., "LU implicit schemes with multiple grids for the Euler equations," AIAA Paper 86-0105, 1986.

[5] Liang, C., Kannan, R., and Wang, Z. J., "A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured triangular grids," *Computers and Fluids*, Vol. 38, 2009, pp. 254–265.

[6] Sun, Y., Wang, Z. J., and Liu, Y., "Efficient implicit non-linear LU-SGS approach for compressible flow computation using high-order spectral difference method," *Communication of Computational Physics*, Vol. 5, No. 2-4, 2009, pp. 760–778.

[7] Parsani, M., Ghorbaniasl, G., Lacor, C., and Turkel, E., "An implicit high-order spectral difference approach for large eddy simulation," *Journal of Computational Physics*, Vol. 229, 2010, pp. 5373–5393.

[8] Haga, T., Sawada, K., and Wang, Z. J., "An implicit LU-SGS scheme for the spectral volume method on unstructured tetrahedral grids," *Communication of Computational Physics*, Vol. 6, No. 5, 2009, pp. 978–996.

[9] Huynh, H. T., "A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion," AIAA Paper 2009-403, 2009.

[10] Lax, P. D., "Weak solutions of nonlinear hyperbolic equations and their numerical computation," *Communications on Pure and Applied Mathematics*, Vol. 7, 1954, pp. 159–193.

[11] Park, J., Kwon, K., and Choi, H., "Numerical solutions of flow past a circular cylinder at Reynolds number up to 160," *KSME International Journal*, Vol. 12, No. 6, 1998, pp. 1200–1205.

[12] Sharman, B., Lian, F. S., Davidson, L., and Norberg, C., "Numerical predictions of low Reynolds number flows over two tandem circular cylinders," *International Journal for Numerical Methods in Fluids*, Vol. 47, 2005, pp. 423–447.

[13] Choi, H. and Moin, P., "Effects of the computational time step on numerical solutions of turbulent flow," *Journal of Computational Physics*, Vol. 113, No. 1, 1994, pp. 1–4.

[14] Caretto, L. S., Gosman, A. D., Patankar, S. V., and Spalding, D. B., "Two calculation procedures for steady, three-dimensional flows with recirculation," *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, edited by L. N. in Physics, Vol. 19, Springer, 1973, pp. 60–68.