Implicit Solution of the Unsteady Euler Equations for High-Order Accurate Discontinuous Galerkin Discretizations

Li Wang * and Dimitri J. Mavriplis†
Department of Mechanical Engineering
University of Wyoming, Laramie, Wyoming, 82072-3295

Efficient solution techniques for high-order accurate time-dependent problems are investigated for solving the two-dimensional non-linear Euler equations in this work. The spatial discretization consists of a high-order accurate Discontinuous Galerkin (DG) approach. Implicit time-integration techniques are considered exclusively in order to avoid the stability restrictions of explicit methods. Standard Backwards differencing methods (BDF1 and BDF2) as well as a fourth-order implicit Runge-Kutta scheme (IRK4) are considered, in an attempt to balance the spatial and temporal accuracy of the overall approach. The implicit system arising at each time step is solved using a p-multigrid approach, which is shown to produce h independent convergence rates, while remaining relatively insensitive to the time-step size. The higher-order time integration schemes such as fourth-order implicit Runge-Kutta are found to be more efficient in terms of computational cost for a given accuracy level as compared to the lower order BDF1 and BDF2 schemes.

I. Introduction

In recent years, discontinuous Galerkin (DG) methods have been gaining wide acceptance for high-accuracy solutions of systems of conservation laws.From a simple linear advection problem to the more complicated non-linear compressible Euler and Naiver-Stokes equations,[1,2,4,8,9,15,16,21,24] high-order accurate DG methods provide an efficient avenue for achieving high spatial accuracy by means of high-order polynomial approximations within individual mesh elements. Additionally, the resulting discontinuities at element interfaces can be handled using approximate Reimann solver techniques, which are now well understood and have been extensively developed in the context of finite volume methods.

For time-dependent problems, DG methods have generally been used in conjunction with high-order accurate explicit time-integration methods, such as explicit Runge-Kutta DG (RKDG) methods.[7,9,21] While such methods are well suited for problems with similar spatial and temporal scales, they are notoriously inefficient for problems with disparate temporal and spatial scales, such as low reduced frequency problems and steady-state problems. For DG discretizations using polynomials of degree p and a p+1 stage explicit RK method of order p+1, the explicit CFL stability limit is given by $\Delta t < \frac{\Delta x}{c(2p+1)}$,[9] where Δx refers to a characteristic cell size, and c is the maximum eigenvalue or wave speed of the system of equations to be solved. As a consequence, implicit time-integration strategies, which are unconditionally stable[6,11,25] are preferred for low reduced frequency problems, since they allow the selection of the time step based purely on temporal accuracy considerations. On the other hand, implicit methods require the solution of one or more non-linear problems at each time step, thus requiring the use of an efficient solution technique in order to make these schemes competitive.

In this work we focus exclusively on constructing implicit time-integration approaches, i.e. first and second-order implicit Backwards Differencing schemes (BDF1 and BDF2), and an implicit fourth-order Runge-Kutta scheme (IRK4), for solving the Euler equations. In order to provide a competitive approach, the implicit system arising at each time-step in these time-integration schemes must be solved in an efficient

*Ph.D. Candidate, email: wangli@uwyo.edu
†Professor, AIAA Associate Fellow; email: mavripl@uwyo.edu.

manner. This is achieved using a spectral multigrid (p-MG) method.[14,15,19,24] The p-MG method, which has been developed previously for steady-state problems, makes use of coarser multigrid levels constructed by lowering the order of the spatial discretization (p), as opposed to reducing the number of mesh elements (h) in the traditional geometric multigrid method, and is capable of achieving h and p independent convergence rates.[24] In this work, the spectral multigrid approach developed for steady-state DG discretizations is extended to high-order time-implicit unsteady problems.

The organization of this paper is as follows. Section II describes the governing equations and some preliminary knowledge related to these equations. Section III describes the spatial and temporal discretizations used in this work. Section IV describes the implicit solution techniques developed in this work, including various element Jacobi iterative smoothers, and the p-multigrid approach used to rapidly converge the implicit system of non-linear equations at each time step. In Section V, two-dimensional numerical results are shown for an isentropic convecting vortex case, and for a periodic vortex-shedding problem using an unstructured trianglar mesh. Finally, Section VI summarizes the conclusions of this paper.

II. Governing Equations

The Euler equations which govern two-dimensional unsteady compressible inviscid flow can be written in the following conservative form,

$$\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u}(\mathbf{x},t))}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u}(\mathbf{x},t))}{\partial y} = 0 \quad in \;\; \Omega_T = \Omega \times (0,T) \tag{1}$$

where $T > 0$ is the length of time interval, and Ω is a two-dimensional bounded domain. The conservative state vector $\mathbf{u}$ and the inviscid Cartesian flux component vectors $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ are defined by

$$\mathbf{u} = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho e \end{array} \right\}, \qquad \mathbf{f} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{array} \right\}, \qquad \mathbf{g} = \left\{ \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 \\ (\rho e + p)v \end{array} \right\} \tag{2}$$

where the notations ρ , p, and e denote the fluid density, pressure and specific total energy per unit mass, respectively. u and v are the velocity components of the flow in the x and y coordinate directions. This system of equations is completed by the perfect gas equation of state given by Eq. (3)

$$p = (\gamma - 1)\rho \left[e - \frac{1}{2}(u^2 + v^2) \right] \tag{3}$$

where γ is defined as the ratio of specific heats of the fluid ($\gamma = 1.4$ for air). This represents a hyperbolic system of equations to which are applied the initial and boundary conditions denoted by Eq.(4) and Eq.(5), respectively, where $\partial\Omega$ represents the boundary of Ω.

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}^0(\mathbf{x}) \tag{4}$$

$$B(\mathbf{u}) = 0 \quad on \; \partial\Omega \times (0,T) \tag{5}$$

III. Discretizations

A. Spatial discretization

The Discontinuous Galerkin (DG) method represents a spatial discretization approach based on a finite-element method, which makes use of element based basis functions which are discontinuous across element interfaces.[28] Thus, in this approach, the computational domain Ω is partitioned into an ensemble of non-overlapping elements, i.e. $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_{nElem}$, where *nElem* denotes the number of elements in the domain. The formulation of the integrand over the global computational domain is therefore subdivided into a collection of integrals over *nElem* elements, given in Eq.(6).

$$\sum_{m=1}^{nElem} \int_{\Omega_m} \left[\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u}(\mathbf{x},t))}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u}(\mathbf{x},t))}{\partial y} \right] dV = 0 \tag{6}$$

Assuming that Ω_m is a typical triangulation element of Ω, we consider possible choices of basis functions and then obtain the following weak statement of Eq. (6) by multiplying by a test function $\mathbf{L}$ and integrating by parts:

$$\frac{d}{dt} \int_{\Omega_m} \mathbf{L}_h \mathbf{u}_h dV - \int_{\Omega_m} \left[\frac{\partial \mathbf{L}_h}{\partial x} \mathbf{f}(\mathbf{u}_h) + \frac{\partial \mathbf{L}_h}{\partial y} \mathbf{g}(\mathbf{u}_h) \right] dV + \int_{\partial \Omega_m} \mathbf{L}_h \left[\mathbf{f}(\mathbf{u}_h) n_x + \mathbf{g}(\mathbf{u}_h) n_y \right] dS = 0 \quad \forall \mathbf{L}_h \tag{7}$$

where $\mathbf{u}_h$ and $\mathbf{L}_h$ represent the finite element approximations making use of truncated polynomial expansions combined with N shape functions ζ_k , expressed as:

$$\mathbf{u}_h = \sum_{k=1}^{N} \mathbf{u}_k(t) \zeta_k(\mathbf{x}) \tag{8}$$

$$\mathbf{L}_h = \sum_{k=1}^{N} \mathbf{L}_k \zeta_k(\mathbf{x}) \tag{9}$$

The unit normal vector $\mathbf{n} = (n_x, n_y)$ is outward to the boundary, and points into the computational domain. The current implementation in this work uses a set of hierarchical shape functions on triangles,[18] enabling solutions from 1st order (p=0), up to 5th order (p=4) spatial accuracy. Substituting the expansions into Eq. (7), we then obtain a set of N equations given as:

$$\frac{d\mathbf{u}_i(t)}{dt} \int_{\Omega_m} \zeta_k \zeta_i \, dV - \int_{\Omega_m} \left[\frac{\partial \zeta_k}{\partial x} \mathbf{f}(\mathbf{u}_h) + \frac{\partial \zeta_k}{\partial y} \mathbf{g}(\mathbf{u}_h) \right] dV + \int_{\partial \Omega_m} \zeta_k \left[\mathbf{f}(\mathbf{u}_h) n_x + \mathbf{g}(\mathbf{u}_h) n_y \right] dS = 0 \quad k = 1, 2, \cdots, N \tag{10}$$

where the repeated index i in Eq. (10) denotes the summation convention ranging from 1 to N, and $\mathbf{u}_i(t)$ represents the modal solutions for the conservative state vector, $\mathbf{u}$. The interface flux $\mathbf{f}(\mathbf{u}_h) n_x + \mathbf{g}(\mathbf{u}_h) n_y$, which is common for the neighboring cells on each edge to ensure conservation, can be treated as Riemann flux function $\mathbf{Flux}(\mathbf{u}_h^L, \mathbf{u}_h^R, \mathbf{n})$, where $\mathbf{u}_h^L$ and $\mathbf{u}_h^R$ represent the internal element interface solution and neighboring element interface solution, respectively. In this work, the HLLC approximate Riemann solver[3,27] is employed. All the integrals including the domain and boundary integrals in Eq. (10) are evaluated by use of Gauss quadrature formulae[12,13] with a number of quadrature points corresponding to the required degree of interpolating polynomials. Specifically, the optimal numbers of quadrature points for the first, second and third integral of Eq. (10) are those satisfied with integrals exactly on polynomials of order $2p$, $2p-1$ and $2p$, respectively.

By grouping together the time-dependent and spatial contributions, Eq. (10) can be written as

$$M \frac{d\mathbf{u}}{dt} + \mathbf{R_p}(\mathbf{u}) = 0 \tag{11}$$

where the mass matrix M has identical diagonal blocks, M_d, for the four modal variables of $(\rho, \rho u, \rho v, \rho e)_i$ which can be written as:

$$M_d = \begin{pmatrix} \int_{\Omega_m} \zeta_1 \zeta_1 dV & \int_{\Omega_m} \zeta_1 \zeta_2 dV & \cdots & \int_{\Omega_m} \zeta_1 \zeta_N dV \\ \int_{\Omega_m} \zeta_2 \zeta_1 dV & \int_{\Omega_m} \zeta_2 \zeta_2 dV & \cdots & \int_{\Omega_m} \zeta_2 \zeta_N dV \\ \cdots & \cdots & \cdots & \cdots \\ \int_{\Omega_m} \zeta_N \zeta_1 dV & \int_{\Omega_m} \zeta_N \zeta_2 dV & \cdots & \int_{\Omega_m} \zeta_N \zeta_N dV \end{pmatrix} \tag{12}$$

The mass matrix M can be precomputed and stored ahead of time due to the fact that is remains unchanged during the solution process. The last vector $\mathbf{R_p}(\mathbf{u})$ represents the steady state residual.

B. Temporal discretization

In order to solve the time-dependent problem, the resulting spatially discretized equations must be integrated in time. Although the use of explicit time-integration schemes has been widespread for DG discretizations, in this work we focus on the use of implicit time-integration schemes, which are not restricted by the CFL-stability limit of explicit methods, and thus are capable of using maximum time-steps determined by accuracy considerations, and are more suitable for stiff problems as well. The implicit time-integration schemes currently employed in this work range from first to fourth-order accurate in time, including both first and second order accurate multistep backwards difference formulations (BDF1, BDF2) and a fourth-order accurate implicit multistage Runge-Kutta scheme (IRK4). In previous work,[5,20] it has been shown that even for second-order accurate finite-volume schemes, fourth-order implicit Runge-Kutta schemes can outperform BDF2 schemes for engineering accuracy levels. Therefore, one of the essential tasks of this paper is to address whether high-order implicit RK schemes can be designed to be more efficient than BDF1 and BDF2 schemes when applied with high-order DG methods.

Starting from the ordinary differential equation (11), the formulae for BDF1 and BDF2 schemes are given respectively as:

$$\mathbf{R_e}(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t}\mathbf{u}_h^{n+1} + \mathbf{R_p}(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t}\mathbf{u}_h^{n} \tag{13}$$

$$\mathbf{R_e}(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t}(\frac{3}{2}\mathbf{u}_h^{n+1}) + \mathbf{R_p}(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t}(2\mathbf{u}_h^{n} - \frac{1}{2}\mathbf{u}_h^{n-1}) \tag{14}$$

where Δt represents the integration time-step and $\mathbf{R_e}$ denotes the unsteady-state residual. $\mathbf{u}_h^n$ and $\mathbf{u}_h^{n+1}$ are numerical solutions for the current and the next (unknown) time step, respectively.

These schemes are relatively efficient because they solve only one implicit set of equations per time step. However, they are difficult to use with variable time-steps and are not A-stable beyond second-order temporal accuracy.[5] In the case of multistage IRK schemes, multiple implicit problems are required (each stage works on one implicit system solution) per time step, but these schemes are easily implemented in the presence of variable time steps and can be constructed to be A- and L-stable for any temporal order. The formula for the IRK4 scheme, in the case of the Euler equations, can be written as,

$$\begin{array}{ll} (i) & \mathbf{u}^{(0)} = \mathbf{u}_h^n \\ (ii) & \textit{For } s = 1, \cdots, \mathcal{S} \textit{ compute the intermediate stage solutions, } \mathbf{u}^{(s)} \\ & \mathbf{R_e}(\mathbf{u}^{(s)}) = \frac{M}{\Delta t}\mathbf{u}^{(s)} + a_{ss}\mathbf{R_p}(\mathbf{u}^{(s)}) - \left[\frac{M}{\Delta t}\mathbf{u}^{n} - \sum_{j=1}^{s-1} a_{sj}\mathbf{R_p}(\mathbf{u}^{(j)})\right] \\ (iii) & \mathbf{u}_h^{n+1} = \mathbf{u}^{(\mathcal{S})} \end{array} \tag{15}$$

where $\mathcal{S}$ is the number of stages required for the IRK scheme at each time step and a_{kj} are the Butcher coefficients of the scheme.

In this paper, we utilize the ESDIRK class of RK scheme, which corresponds to Explicit first stage, Single Diagonal coefficient, diagonally Implicit Runge-Kutta. The Butcher table[5,20] for a six stage ESDIRK scheme (fourth-order accurate, $\mathcal{S} = 6$) is shown in Table (1). The set of coefficients, a_{kj}, defines the implicit RK schemes (as shown in Eq.(15)). The first stage is explicit due to $a_{11} = 0$. A single implicit scheme is solved at each individual stage since the set of a_{kj} has the form of a lower triangular matrix. The solution in the last stage is the solution for the next time-step and c_k represents the point in the time interval, $[t, t + \Delta t]$ and satisfies,

$$c_k = \sum_{j=1}^{k} a_{kj} \qquad (k = 1, 2, \ldots, 6) \tag{16}$$

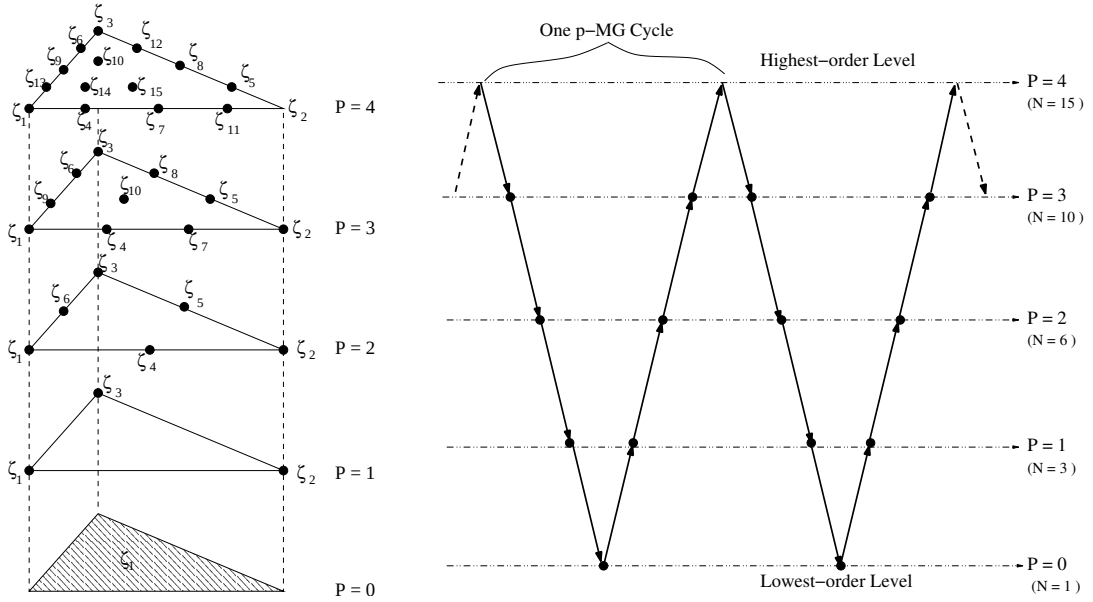
IV. p-MG Approach

As mentioned previously, implicit time-integration methods require the solution of one or more implicit problems per time step. Efficient solvers are required for this task in order to achieve an overall competitive approach. Our approach consists of using a spectral multigrid strategy,[15,19,21,24] in conjunction with an

Table 1. Butcher Tableau for ESDIRK class of RK schemes with six stages

$c_1 = 0$	0	0	0	0	0	0
c_2	a_{21}	a_{66}	0	0	0	0
c_3	a_{31}	a_{32}	a_{66}	0	0	0
c_4	a_{41}	a_{42}	a_{43}	a_{66}	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{66}	0
$c_6 = 1$	$a_{61} = b_1$	$a_{62} = b_2$	$a_{63} = b_3$	$a_{64} = b_4$	$a_{65} = b_5$	a_{66}
$\mathbf{u}^{n+1}$	b_1	b_2	b_3	b_4	b_5	b_6

element Jacobi smoother on each multigrid level, for solving the implicit system at each time step. The spectral multigrid or p-MG algorithm is based on the standard geometric multigrid method,[26] but, instead of using physically fewer elements in coarser levels, lower-order interpolants serve as "coarse" levels while the same spatial grid/element is used by all levels, as shown in Fig. 1(a). Therefore, with this methodology, no additional grid information needs to be stored.



(a) (b) V Cycle of p-MG (N, degree of shape functions)

Figure 1. Five levels of p-MG approach

A. Element Jacobi smoother

At each p-multigrid level, an element-Jacobi scheme is used as a smoother. Returning to Eq. (13), for example, in order to solve this implicit problem for the $n+1$ time step, let us rewrite it as a Newton scheme given in Eq. (17), where k refers to the subiteration index for the linear system in the Newton scheme. Note that the superscript $n+1$ is omitted in the expression of Newton iterations to avoid notation confusion, and $\mathbf{w}$ represents intermediate solutions in the Newton solver process for the solution of $\mathbf{u}_h^{n+1}$, which is satisfied with $[\partial \mathbf{R_e}/\partial \mathbf{u}_h] = [\partial \mathbf{R_e}/\partial \mathbf{w}]$

$$
\begin{aligned}
&(i) \quad \mathbf{w}^1 = \mathbf{u}_h^n \\
&(ii) \quad \left[\frac{\partial \mathbf{R_e}}{\partial \mathbf{w}}\right]^k \Delta w^{k+1} = -\mathbf{R_e}(\mathbf{w}^k) \\
&\qquad \mathbf{w}^{k+1} = \mathbf{w}^k + \alpha \Delta w^{k+1} \quad , \; k \; = \; 1, \; 2, \; \cdots \mathcal{M} \\
&(iii) \quad \mathbf{u}_h^{n+1} = \mathbf{w}^{\mathcal{M}} \quad , \; when \; \mathbf{R_e}(\mathbf{w}^{\mathcal{M}}) = 0
\end{aligned}
\tag{17}
$$

where α is a relaxation parameter to keep $\|\alpha \Delta \mathbf{w}^{k+1}/\mathbf{w}^k\|_{L_\infty} \leq 10\%$.

The element Jacobi smoother can be viewed as an approximate Newton method, where only the Jacobian entries corresponding to the modal coupling between all modes within an element are retained, resulting in a block diagonal matrix, D_T, which is easily inverted using Gaussian elimination at the block level, and all other entries, O_T are discarded. This solver is generally denoted as non-linear element Jacobi solver (NEJ). The quasi-nonlinear element Jacobi (qNEJ) approach represents a variant of the NEJ solver where the block Jacobians are frozen over a certain number of iterations, and thus only the unsteady residual,$\mathbf{R_e}$, needs to be updated at each subiteration. A third solver variant is denoted as the linearized element Jacobi (LEJ) method, in which the full Jacobian entries consisting of both diagonal (D_T) and off-diagonal components (O_T) are retained, as shown in Eq. (18). The full Jacobian is then approximately inverted using a block Jacobi iteration strategy, where the (D_T) blocks are treated implicitly, and the (O_T) blocks are treated implicitly. This corresponds to the linearization of the NEJ scheme. The fundamentals of these three solver variants are listed in Table 2.

$$
\left[\partial \mathbf{R_e}(\mathbf{w}^k)/\partial \mathbf{w}\right] = \left[D_T^k\right] + \left[O_T^k\right] \tag{18}
$$

Table 2. Fundamentals of NEJ, qNEJ and LEJ solvers

NEJ	1. $\mathbf{w}^1 = \mathbf{u}_h^n$ *get* $\mathbf{R_e}(\mathbf{w}^1)$ *get* $D_T^1(\mathbf{w}^1)$	2. *for* $k = 1, \; 2, \cdots, \mathcal{M}$ $\Delta \mathbf{w}^{k+1} = \left[D_T^k\right]^{-1} \left[\mathbf{S} - \mathbf{R_e}(\mathbf{w}^k)\right]$ $\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha \Delta \mathbf{w}^{k+1}$ *update* $\mathbf{R_e}(\mathbf{w}^{k+1})$ *update* $D_T(\mathbf{w}^{k+1})$	3.$\mathbf{u}_h^{n+1} = \mathbf{w}^{\mathcal{M}}$ *when* $\mathbf{R_e}(\mathbf{w}^{\mathcal{M}}) = 0$
qNEJ	1. $\mathbf{w}^1 = \mathbf{u}_h^n$ *get* $\mathbf{R_e}(\mathbf{w}^1)$ *get* $D_T^n(\mathbf{u}_h^n)$	2. *for* $k = 1, \; 2, \cdots, \mathcal{M}$ $\Delta \mathbf{w}^{k+1} = [D_T^n]^{-1} \left[\mathbf{S} - \mathbf{R_e}(\mathbf{w}^k)\right]$ $\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha \Delta \mathbf{w}^{k+1}$ *update* $\mathbf{R_e}(\mathbf{w}^{k+1})$	3.$\mathbf{u}_h^{n+1} = \mathbf{w}^{\mathcal{M}}$ *when* $\mathbf{R_e}(\mathbf{w}^{\mathcal{M}}) = 0$
LEJ	1. $\mathbf{w}^1 = \mathbf{u}_h^n$ *get* $\mathbf{R_e}(\mathbf{w}^1)$ *get* $D_T^n(\mathbf{u}_h^n)$ *get* $O_T^n(\mathbf{u}_h^n)$	2. *for* $k = 1, \; 2, \cdots, \mathcal{M}$ $\Delta \mathbf{w}^{k+1} = [D_T^n]^{-1} \left[\mathbf{S} - \mathbf{R_e}(\mathbf{w}^k) - O_T^n \Delta \mathbf{w}^k\right]$ $\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha \Delta \mathbf{w}^{k+1}$ *update* $\mathbf{R_e}(\mathbf{w}^{k+1})$	3.$\mathbf{u}_h^{n+1} = \mathbf{w}^{\mathcal{M}}$ *when* $\mathbf{R_e}(\mathbf{w}^{\mathcal{M}}) = 0$

In table 2, the source term $\mathbf{S}$, which represents the residual restriction term from the finer multigrid levels, is included in the formulae on all levels for consistency, although this term vanishes on the finest level (highest-order approximation level) in the multigrid formulation. Moveover, note that the diagonal block matrix D_T, the off-diagonal block O_T and the unsteady-state residual $\mathbf{R_e}$ may vary with the different time-integration schemes. To make this point clear, we also list these terms for the corresponding BDF1, BDF2 schemes, and for intermediate stage functions of the IRK4 scheme, respectively, as given in table 3, where D^k and O^k denote the diagonal and off-diagonal blocks of the Jacobian matrix for the corresponding steady-state solver, $\left[\partial \mathbf{R_p}(\mathbf{w}^k)/\partial \mathbf{w}\right]$, in a similar manner.

B. Multigrid V-Cycle

In this work, a V-cycle spectral multigrid methodology has been employed for accelerating the solution at each implicit time step. As shown in Fig. 1(b), for the case of a fifth-order accurate discretization scheme, the

Table 3. Diagonal, off-diagonal blocks and unsteady-state residuals for BDF1, BDF2 and IRK4 schemes

	$D_T(\mathbf{w}^k)$	$O_T(\mathbf{w}^k)$	$\mathbf{R_e}(\mathbf{w}^k)$
BDF1	$D_T = \frac{M}{\Delta t} + D^k$	O^k	$\frac{M}{\Delta t}\mathbf{w}^k + \mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}\mathbf{u}_h^n$
BDF2	$D_T = \frac{3}{2}\frac{M}{\Delta t} + D^k$	O^k	$\frac{3}{2}\frac{M}{\Delta t}\mathbf{w}^k + \mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}(2\mathbf{u}_h^n - \frac{1}{2}\mathbf{u}_h^{n-1})$
IRK4 $(s = 1, \cdots, 6)$	$D_T = \frac{M}{\Delta t} + a_{ss}D^k$	$a_{ss}O^k$	$\frac{M}{\Delta t}\mathbf{w}^k + a_{ss}\mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}\mathbf{u}_h^n - \sum_{j=1}^{s-1} a_{sj}\mathbf{R_p}(\mathbf{u}^{(j)})$

finest multigrid level consists of the original $p = 4$ (fifth-order accurate) discretization, while the intermediate levels consist of lower $p = 3$, $p = 2$, and $p = 1$ discretizations, while a $p = 0$ first-order accurate discretization is employed at the lowest level.

The p-multigrid approach fits naturally with DG discretizations because the use of a hierarchical basis set makes the formulation of interpolation operators between high-order and low-order approximation levels very simple. The restriction operator (I_p^{p-1}) from high-order level to low-order level, is obtained by simply deleting the corresponding higher order modal coefficients and transferring lower order modes exactly. The prolongation operator (I_{p-1}^{p}) from lower-order to high-order level is obtained by injecting lower order modal coefficients exactly. The main reason for this simple projection is due to the fact that the lower order interpolant functions are a subset of the higher order ones. In this work, we make use of the Full Approximation Storage (FAS)[22,23] multigrid scheme. The procedure of a two level V-cycle FAS p-MG approach is as follows, note that we use notation $\mathbf{w}_p^m$ to denote intermediate solution in the time interval $[n\Delta t,(n+1)\Delta t]$, where the superscript represents the number of p-MG cycles and subscript represents the p-MG level.

- Perform τ_1 subiterations on the high-order approximation level (p), using any Element Jacobi solver mentioned previously: $\mathbf{R_e}_p(\mathbf{w}_p^m) = \mathbf{S}_p$; Get residual: $r_p^m = \mathbf{S}_p - \mathbf{R_e}_p(\mathbf{w}_p^m)$
- Restrict both residual and solution to the low-order approximation level ($p-1$): $\mathbf{S}_{p-1} = I_p^{p-1} r_p^m$; $\mathbf{w}_{p-1}^m = I_p^{p-1}\mathbf{w}_p^m$
- Solve the low-order approximation level problem by using the same Element Jacobi solver with τ_2 subiterations: $\mathbf{R_e}_{p-1}(\mathbf{w}_{p-1}^m) = \mathbf{S}_{p-1}$; Obtain the low-order level error: $e_{p-1}^m = w_{p-1}^m - I_p^{p-1}\mathbf{w}_p^m$
- Prolongate this low-order level error to correct the high-order approximation level: $\mathbf{w}_p^{m+1} = \mathbf{w}_p^m + I_{p-1}^p \mathbf{e}_{p-1}^m$
- Continue the same steps to the next V-cycle of p-MG method until $\mathbf{R_e}_p(\mathbf{w}_p^{m+1}) = 0$, then the solution for the $n+1$ time step is obtained: $\mathbf{u}_h^{n+1} = \mathbf{w}_p^{m+1}$

For the p-mltigrid method, the same restriction operator is used for both residual and solution restriction, while this is not necessarily the case for geometric multigrid methods.

V. Numerical Results

In this section, two test cases are used to illustrate the performance of the proposed solution techniques for the time-dependent compressible Euler equations. Firstly, the performance of various implicit time-integration schemes for simulating an isentropic convecting vortex flow is demonstrated. The comparisons of temporal accuracy and efficiency are discussed for this test case, and the performance of the p-MG methodology is compared with that of the single level solver. The second test problem consists of a periodic vortex-shedding flow over a triangular wedge on an unstructured triangular mesh, and is used to illustrate the effectiveness of increasing the spatial discretization on overall accuracy. Both test cases are concerned with problems of low reduced frequency, which are naturally suitable for implicit time-stepping strategies.

A. Convection of an isentropic vortex

The convection of an 2-D inviscid isentropic vortex[10,17,29] is simulated to examine the performance of the proposed implicit time-stepping schemes. The exact solution for this test case at any time t is the initial solution at $t_0 = 0$ translated over a distance $u_\infty t$ for a horizontally convecting vortex, which provides a valuable reference for measuring the accuracy of the computed solution.

1. Numerical solutions

The mean flow density, ρ_∞, velocity, u_∞ and v_∞, pressure, p_∞ and temperature T_∞ are taken as freestream values. In this case we set these flow parameters as $(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 1, 0, 1, 1)$ with periodic boundary conditions in the horizontal direction and free boundary conditions in the vertical direction. At $t_0 = 0$, the flow is perturbed by an isentropic vortex $(\delta u, \delta v, \delta T)$ centered as (x_0, y_0) with the form:

$$\delta u = -\frac{\alpha}{2\pi}(y - y_0)e^{\phi(1-r^2)} \tag{19}$$

$$\delta v = \frac{\alpha}{2\pi}(x - x_0)e^{\phi(1-r^2)} \tag{20}$$

$$\delta T = -\frac{\alpha^2(\gamma - 1)}{16\phi\gamma\pi^2}e^{2\phi(1-r^2)} \tag{21}$$

where, ϕ is a parameter determining the gradient of the solution and α determines the strength of the vortex, $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the distance to the vortex center, and $\gamma = 1.4$ is the ratio of specific heats of the fluid. In this study, we set ϕ as unity and α as 4.0. Given the perturbation functions shown in Eq. (19), (20) and (21), we can determine other resulting conservative variables, assuming isentropic flow throughout the domain (i.e. $p/\rho^\gamma = 1$ and $T = p/\rho$ for a perfect gas):

$$\rho = T^{1/(\gamma-1)} = (T_\infty + \delta T)^{1/(\gamma-1)} = \left[1 - \frac{\alpha^2(\gamma - 1)}{16\phi\gamma\pi^2}e^{2\phi(1-r^2)}\right]^{1/(\gamma-1)} \tag{22}$$

$$u = u_\infty + \delta u = 1 - \frac{\alpha}{2\pi}(y - y_0)e^{\phi(1-r^2)} \tag{23}$$

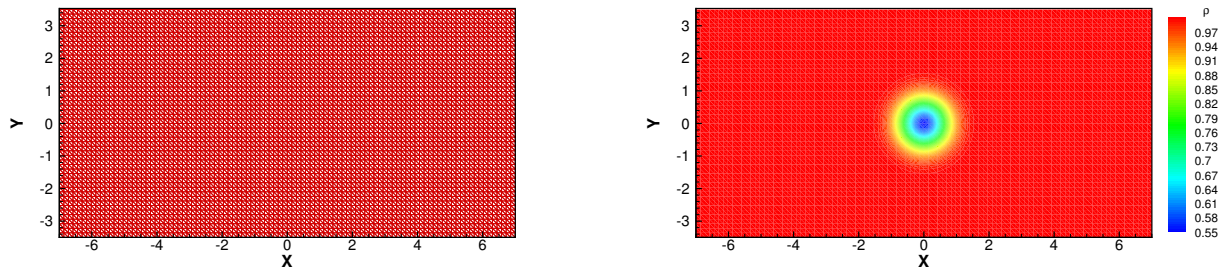
$$v = v_\infty + \delta v = 1 + \frac{\alpha}{2\pi}(x - x_0)e^{\phi(1-r^2)} \tag{24}$$

This test case employs a uniform Cartesian triangle grid. The initial vortex is placed at $(x_0, y_0) = (0, 0)$ on a domain of $-7 \le x \le 7$ and $-3.5 \le y \le 3.5$ with 10000 elements, as shown in Fig. 2. $p = 4$ (fifth-order accurate) is used for the spatial discretization in all cases. The time-step (Δt) is set equal to 0.2. Since the local CFL number is defined as,

$$CFL_i = \frac{\Delta t}{vol_i}\sum_{j=1}^{3edges}(|\mathbf{u}\cdot\mathbf{n}| + c) \qquad 1 \le i \le nElem \tag{25}$$

where *vol* denotes the area of the element in 2D case and c is the local speed of sound, the time-step $\Delta t = 0.2$, corresponds to a maximum CFL number of 11. Fig. 2 illustrates the computational mesh and the initial density contours in the domain. The length of the domain is 14, and the horizontal velocity is $u = 0.5$, thus the vortex requires $T = 28$ to complete one revolution around the periodic grid in the x-direction. Computed density solutions at times $t = 4$, $t = 10$, $t = 20$ and $t = 50$ obtained by BDF1, BDF2 and IRK4 schemes are shown in Figures 3 and 4. The first three of these solutions are within the first horizontal period of the vortex motion, and the last $t = 50$ is close to the end of the second period (since it requires $t = 56$ to complete the second period). Fig. 3 illustrates how the vortex is diffused using the BDF1 scheme even at these small time increments. Conversely, the BDF2 scheme displays a substantially better shape retaining property for the vortex, except at later times such as $t = 50$ where oscillations and some dispersion are evident around the brink of the vortex. On the other hand, the IRK4 scheme shows the best accuracy with the final shape of the vortex at $t = 50$ being almost indistinguishable from the initial shape.

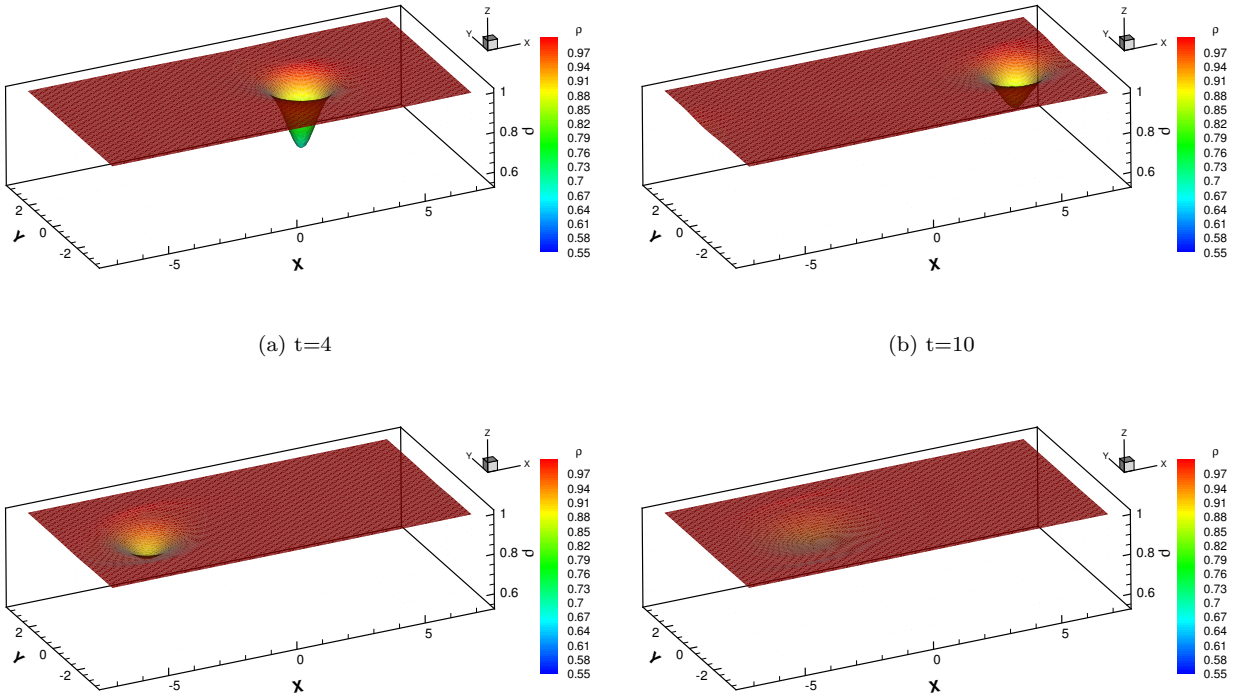
In order to further compare vortex solutions for various time-integration schemes, the density profiles along the horizontal centerline are compared with the exact solution, obtained by translating the initial



(a) grid

(b) initial density contour (2D)

Figure 2. Grid and initial density contour



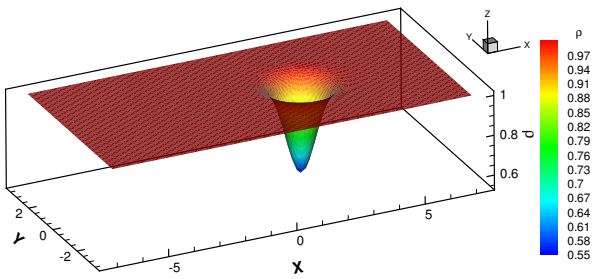
(a) t=4

(b) t=10

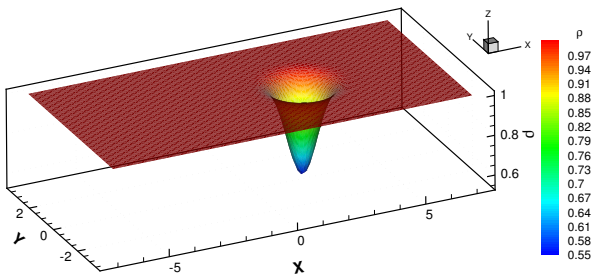
(c) t=20

(d) t=50

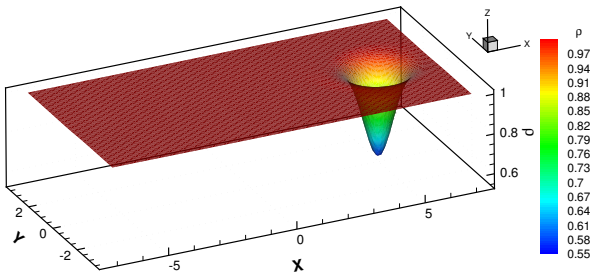
Figure 3. Density countours (3D) of the BDF1 scheme at various times, t=4, 10, 20 and 50, using a time-step of Δt=0.2



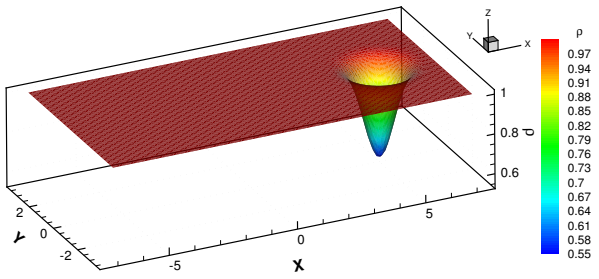
(a) t=4, BDF2



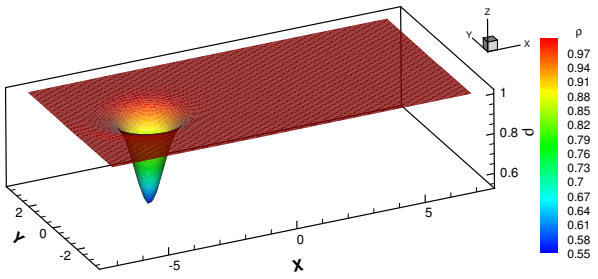
(b) t=4, IRK4



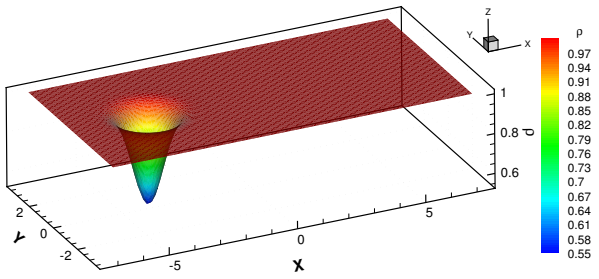
(c) t=10, BDF2



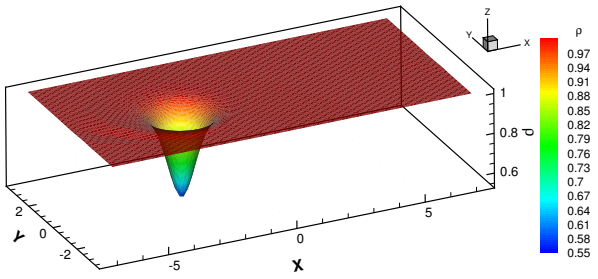
(d) t=10, IRK4



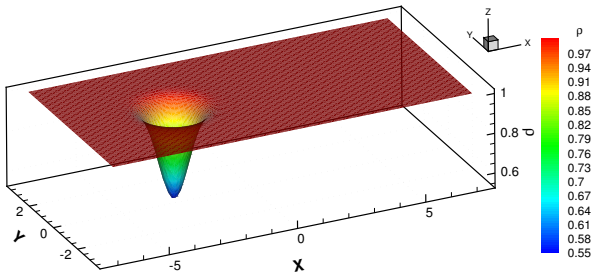
(e) t=20, BDF2



(f) t=20, IRK4

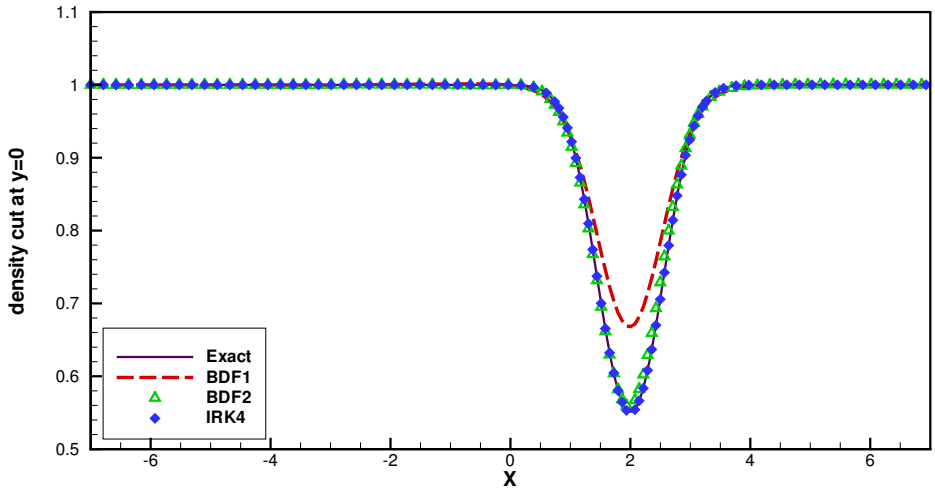


(g) t=50, BDF2

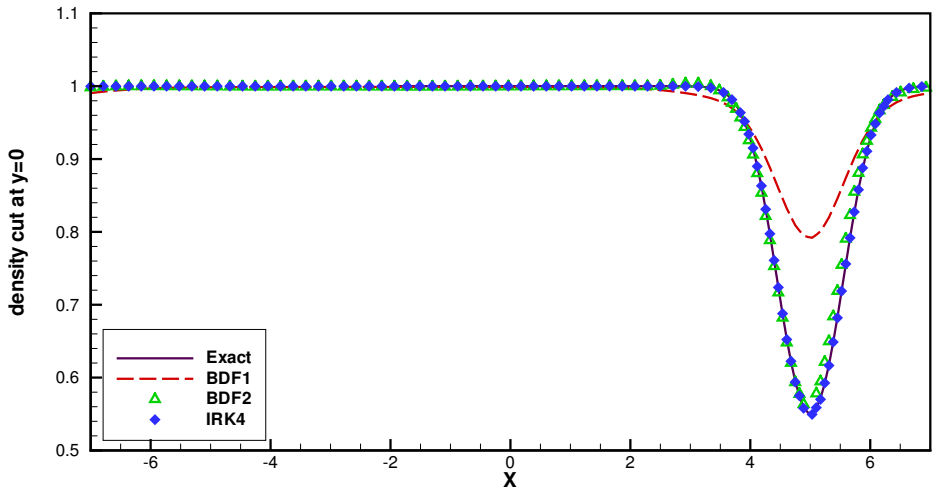


(h) t=50, IRK4

Figure 4. Density countours (3D) of the BDF2 and IRK4 schemes at various times, t=4, 10, 20 and 50, using a time-step of Δt=0.2

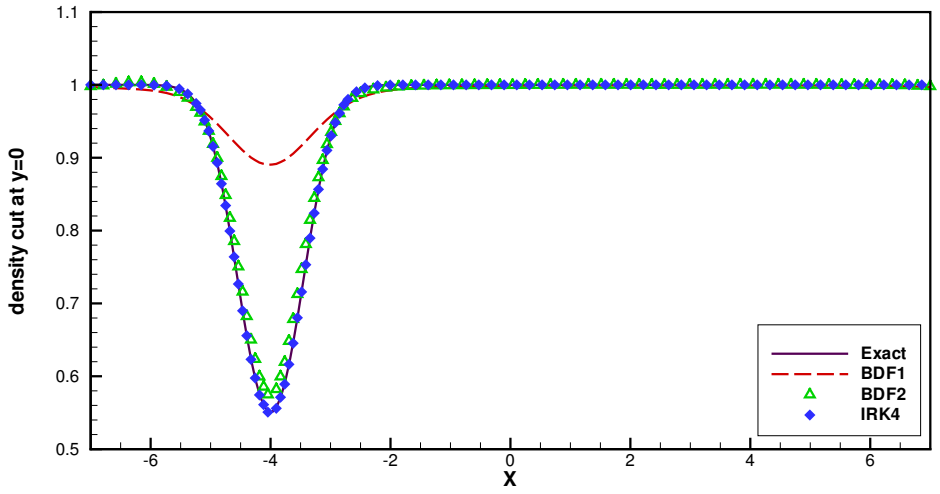


(a) t=4

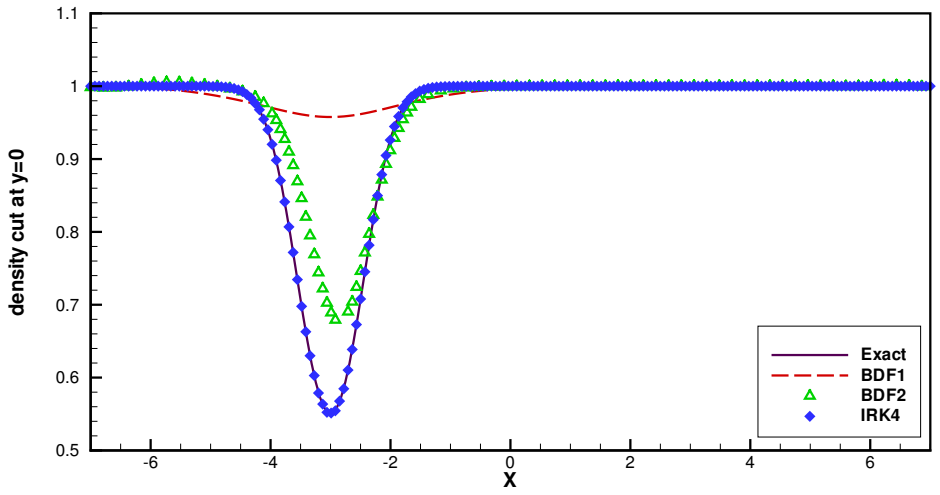


(b) t=10

Figure 5. Comparison of Density Profiles for BDF1, BDF2 and IRK4 schemes at t=4 and t=10, using a time-step of Δt=0.2



(a) t=20



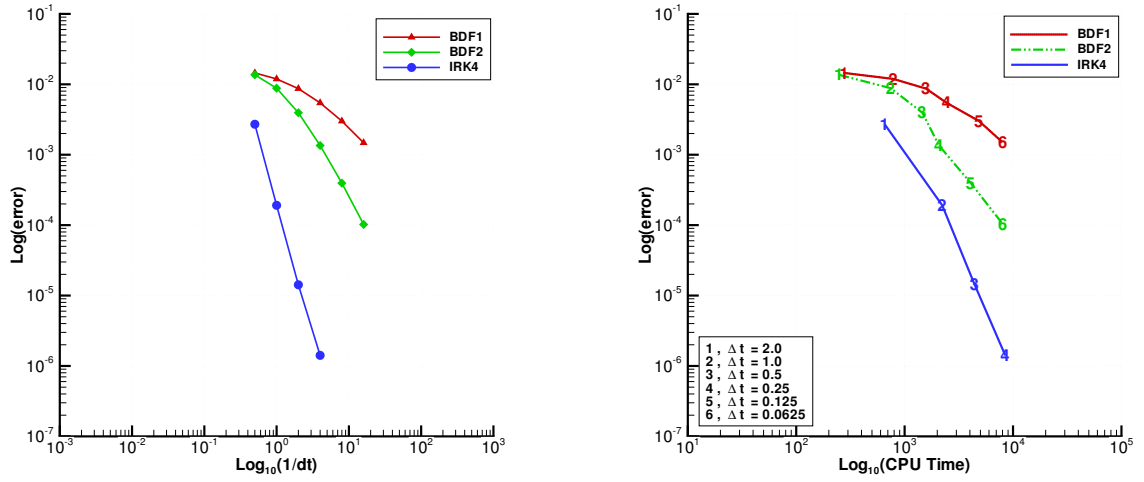
(b) t=50

Figure 6. Comparison of Density Profiles for BDF1, BDF2 and IRK4 schemes at t=20 and t=50, using a time-step of Δt=0.2

centerline density profile. Specifically, Fig. 5 and Fig. 6 display the comparisons of the density profile for the BDF1, BDF2 and IRK4 time-integration schemes, with the exact solution, at various times, $t = 4$, $t = 10$ and $t = 20$, $t = 50$, respectively. As shown in these two figures, it is evident that the IRK4 scheme exhibits the best resolution and provides very good agreement with the exact solution, since there is no visual deviation between the computed results and exact results and the vortex core is well conserved. On the other hand, the BDF1 scheme which is only first-order accurate in time, produces rapi dissipation of the vortex core, as mentioned previously, while the BDF2 scheme provides substantially better resolution and higher accuracy than the BDF1 scheme. Before $t = 20$, the computed results obtained by the BDF2 scheme fall almost on top of the exact solution, although at later times, the BDF2 results show substantial deviation from the exact profile.

2. Temporal accuracy and temporal efficiency

A temporal refinement study is presented to assess the accuracy among various time-integration schemes, BDF1, BDF2, and IRK4 given the same initial condition, periodic boundary condition and $p = 4$ spatial discretization, as described in the previous section. Because the overall error is due to both spatial and temporal error, the "numerical exact" solution for one temporal scheme is obtained using a small time-step on its own, in order to eliminate the effect of spatial error and to isolate the temporal error. The time-step to obtain the "exact" solution is $\Delta t = 0.01$ for all time-integration schemes. Various time-steps, consisting of $\Delta t = 2.0$, 1.0, 0.5 and 0.25, which correspond to a maximum CFL number of 110, 55, 28 and 14, respectively, have been used for all of three temporal schemes. Additionally, another two time-steps, $\Delta t = 0.125$ and 0.0625 are employed for the BDF1 and BDF2 schemes to extend their range of comparison. A regular mesh with a grid spacing of $\Delta x = \Delta y = 0.25$ and a total 3136 elements is employed in the accuracy and efficiency tests. The temporal error is obtained by computing the RMS difference of all conserved variables at all grid points between the computed solution and the reference exact solution.



(a) Temporal accuracy for the BDF1, BDF2 and IRK4 schemes, as a function of time-step size

(b) Temporal accuracy for the BDF1, BDF2 and IRK4 schemes, as a function of CPU time, using p-MG solver

Figure 7. Comparsions of temporal accuracy and efficiency for various implicit schemes at t = 4

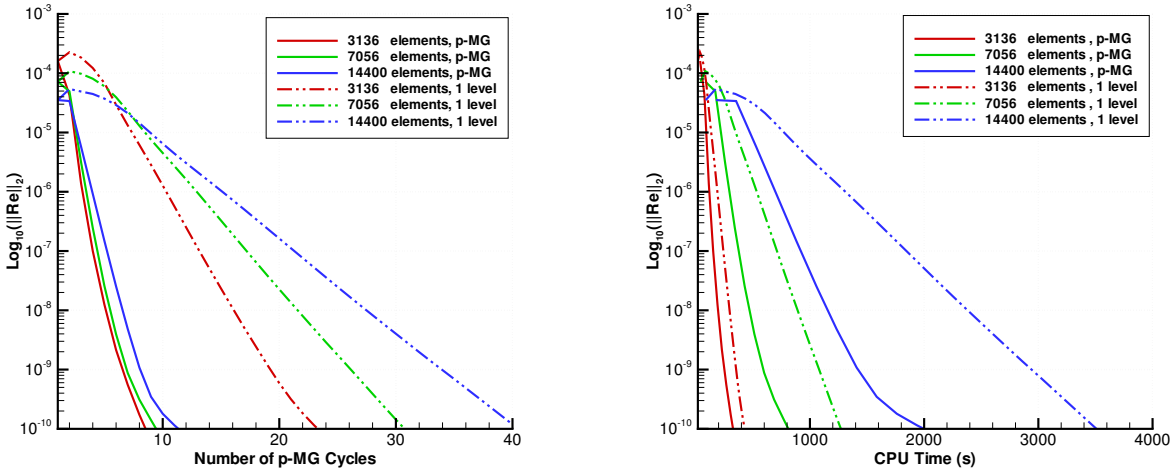
The temporal accuracy results for the BDF1, BDF2 and IRK4 schemes at $t = 4$ are given in Fig. 7(a), where the computed temporal error is plotted as a function of the time-step on a log-log plot. The fourth-order Runge-Kutta scheme exhibits a slope of 3.82, which is near to the design value of 4. The first and second-order backwards differencing schemes display a slope of 1.0 and 1.9, respectively. For any given time-step size, IRK4 obtains a drastically higher accuracy level than the BDF1 and BDF2 schemes, while

the BDF2 scheme provides better accuracy than BDF1. For example, using a time-step size of $\Delta t = 0.25$, the IRK4 scheme attains a temporal error of near 10^{-7} but the other two backwards Euler schemes incur temporal errors larger than 10^{-3}. Even employing a very small time-step size, $\Delta t = 0.0625$, which is just six times larger than the "small time-step" used to obtain the "exact" solution, the BDF2 scheme incurs a temporal error of the order of 10^{-4}, while the BDF1 scheme incurs a temporal error of the order of 10^{-3}.

Fig. 7(b) presents the computed temporal error for the BDF1, BDF2 and IRK4 schemes as a function of the required CPU time, which is obtained using the V-cycle p-MG algorithm (qNEJ) described in Section III, the performance of which is discussed further in the following section. These results illustrate that, in order to reach a specified error level, the IRK4 scheme requires the least CPU time, while the BDF1 scheme is clearly not practical in terms of efficiency, due to its low convergence rate. For example, to achieve an accuracy level of 1×10^{-4}, the IRK4 scheme requires 2615 seconds of CPU time, while the BDF2 scheme requires 8042 seconds, which is over 3 times longer. Furthermore, for higher accuracy levels, the advantage of the IRK4 scheme increases, due to the asymptotic properties of these schemes.

3. Efficiency of p-MG strategy

The p-MG approach is capable of obtaining rapid convergence rates, minimizing CPU time especially for large meshes and large time-steps. The use of an efficient implicit solver is particularly critical with regards to the implementation of implicit Runge-Kutta schemes, which require the solution of multiple implicit systems at each time step, as opposed to BDF schemes which only require the solution of a single implicit system at each time step. The higher overall efficiency of the IRK4 scheme demonstrated previously relies on the use of an efficient implicit solver, namely the p-MG solver, and may not be as competitive in cases where weaker solvers such as a single level solver are emploed. Therefore, in this section, we examine the convergence rates of the FAS p-MG solver (qNEJ) and the corresponding single level solver for a given implicit time step, as a function of grid size and time-step size. Specifically, the grid sizes vary from 3136, 7056 to 14400 elements on a domain of $-7 \leq x \leq 7$ and $-3.5 \leq y \leq 3.5$, and time-step sizes range from $\Delta t = 0.5$, $\Delta t = 1.0$ to $\Delta t = 5.0$. The initial-boundary conditions, spatial discretization ($p = 4$) are kept unchanged and we choose BDF2 as the time-integration scheme.



(a) convergence rate vs. p-MG cycles

(b) convergence rate vs. CPU time

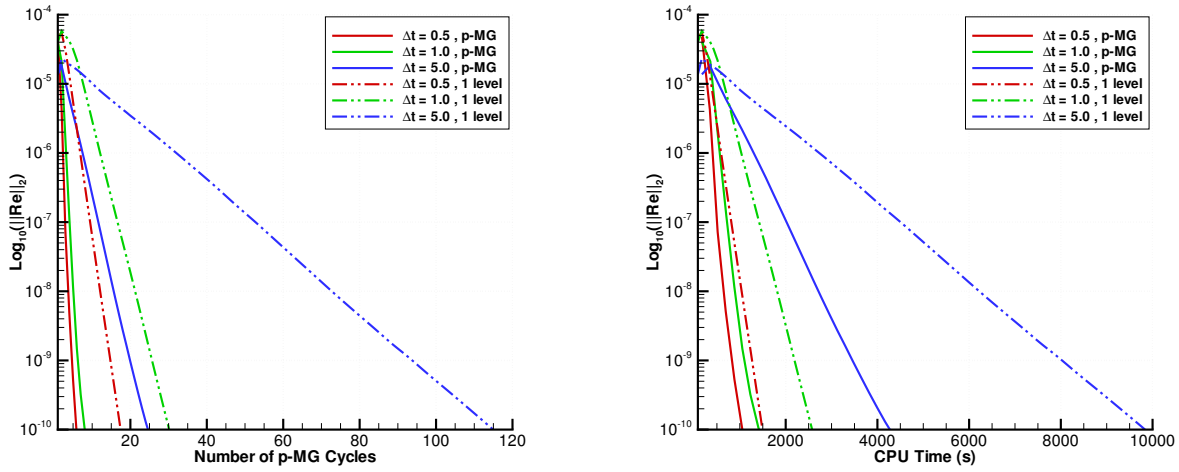
Figure 8. Comparisons of the convergence rates between p-MG and single level solvers, using variant grid sizes, N=3136, N=7056 and N=14400 with a fixed time-step of Δt = 1.0 of BDF2 scheme

Fig. 8(a) depicts the p-multigrid convergence rate for various mesh sizes, compared with the single level solver for a fixed time step of the BDF2 scheme. The solid lines represent computed results using the p-multigrid strategy and the dashed lines represent results using the corresponding single level solver, i.e. no

multigrid involved. Since the time-step size has been fixed for all runs, this figure illustrates the effect of grid resolution or mesh size on convergence rate. An important aspect of these results is the relative insensitivity of the p-multigrid convergence rate to the mesh size, which illustrates the h-independent property of the p-MG solver. On the other hand, the convergence rate of the single level solver decreases drastically for finer mesh sizes. While a p-multigrid cycle requires more computational time than a single-grid cycle, the p-multigrid approach is still seen to be more efficient overall in terms of cpu time than the single level solver, and the benefit of the p-multigrid solver can be expected to increase for larger mesh sizes.

Since time-step size is another factor which affects convergence rate and computational time, we then discuss the characteristics of the p-multigrid approach and corresponding single level solver with respect to time-step sizes. The obtained convergence rates for various time-step sizes are illustrated in Fig. 9, for the largest mesh size of N=14400. As previously, the p-multigrid approach delivers faster convergence and requires lower overall cpu time than the corresponding single-level solver, for a given time-step.

As the time-step size is increased, the convergence of both solvers deteriorates, as expected, since small time-step sizes correspond to more diagonally dominant systems. In particular, at relatively small time-step values, the use of the p-multigrid solver may not be necessary, and a few passes of the single grid solver (element Jacobi) may be sufficient to adequately solve the implicit system. However, for large time steps, the convergence of the p-multigrid solver deteriorates less severely than that of the single level solver, and will eventually asymptote to the convergence observed for steady state-state problems.[24] Thus, for moderate time step sizes, the p-multigrid solver provides the most efficient mechanism for integrating the implicit equations in time, and the benefit of this method can be expected to increase for larger time step sizes and for finer meshes. For example, using a time step of $\Delta t = 1$ on the mesh size of N=14400, the p-multigrid approach achieves a speedup of 1.8 over the single grid solver, while a speedup of 2.35 is obtained for a time step of $\Delta t = 5$.



(a) convergence rate vs. p-MG cycles

(b) convergence rate vs. CPU time

Figure 9. Comparisons of the convergence rates between p-MG and single level solvers, using variant time-step sizes, $\Delta t = 0.5, \Delta t = 1.0$ $\Delta t = 5.0$, with a fixed mesh size of N=14400 of BDF2 scheme

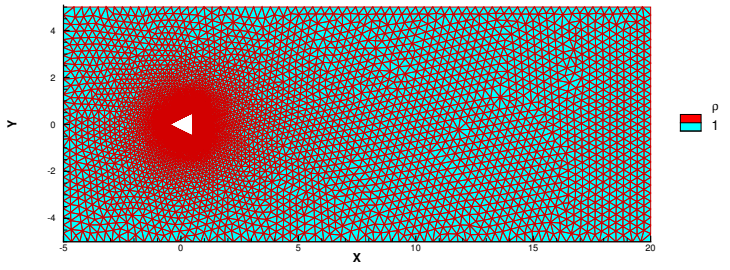
B. Shedding flow over a triangular wedge

1. Numerical solutions

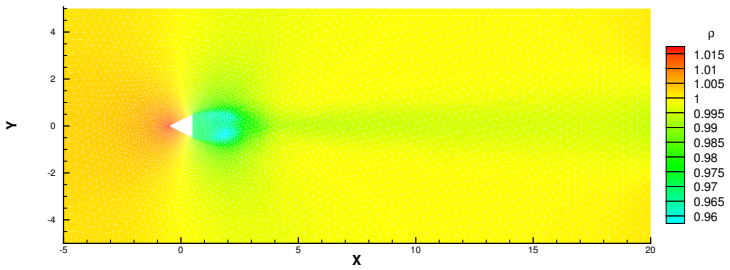
This test problem, which was originally used in the context of viscous flows,[30] is used to illustrate the numerical accuracy and performance when different orders of spatial discretization are employed. In this case, first-order accurate ($p = 0$) up to fourth-order accurate ($p = 3$) spatial discretizations are used. A uniform free stream with a Mach number of $M = 0.2$ flows over a triangular wedge placed on the centerline

$y = 0$ of the domain. The spatial computational domain contains 10836 unstructured triangular elements. Both the BDF2 scheme and the IRK4 scheme are used with a fixed time-step size of $\Delta t = 0.05$, corresponding to a maximum CFL number of 85 which occurs at the smallest cells in the grid, is employed for all runs. Free boundary conditions are applied to all cases. Here note that if we use an explicit second-order forwards time-difference scheme, the maximum permissible time-step size required for stability is $\Delta t_{max} = 5 \times 10^{-5}$.

As the flow passes the wedge, the flow separates after some time due to the "artificial viscosity"[7] and vortices are originated around the two sharp corners and then convected downstream with shedding. While what we focus on here is the ability of these schemes for retaining the shape of the vortices as these are convected downstream of the body. Based on this purpose, we first employ a $p = 0$ spatial scheme and BDF2 temporal scheme, by using a uniform flow as the initial condition: $v(\mathbf{x}, t = 0) = v_\infty = 0$, $\rho(\mathbf{x}, t = 0) = \rho_\infty = 1.0$ (as shown in Fig. 10(a)), and obtain an intermediate solution (shown in Fig. 10(b)), in which the formed vortices have not separated yet. This intermediate solution is then used as the initial condition for all other high-order $p = 1$, 2, 3 spatial-discretization schemes and the BDF2, IRK4 time-integration schemes.



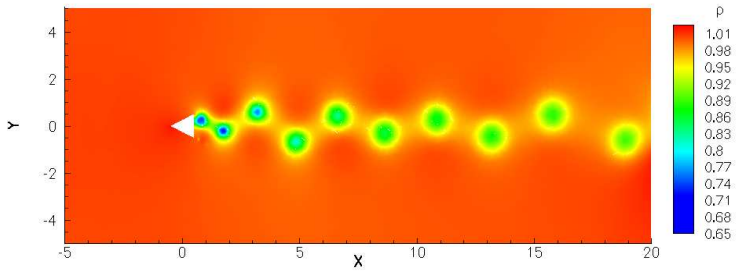
(a) initial condition



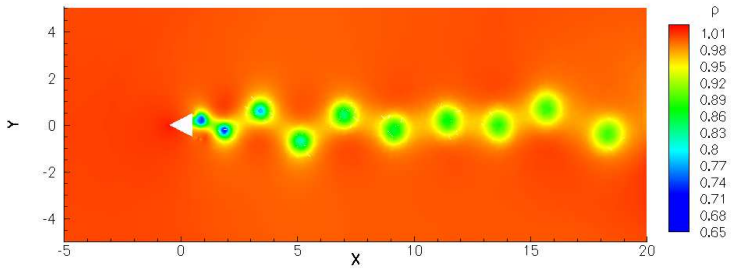
(b) intermediate condition

Figure 10. density contours of initial and intermediate solutions on $p = 0$ scheme

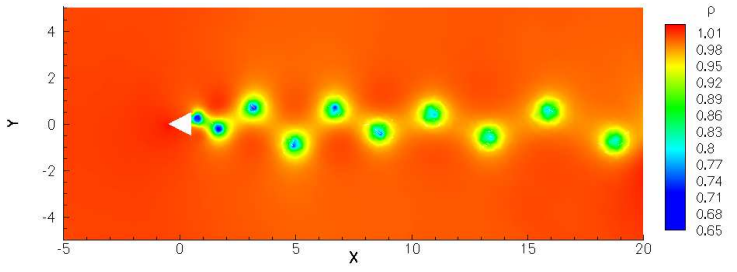
Using the computed initial condition, Figures 11 and 12 plot the numerical results at $t = 100$ for $p = 1$, $p = 2$ and $p = 3$ spatial discretizations using the BDF2 and IRK4 time-integration schemes, respectively. It can be observed that for a fixed temporal scheme (either the BDF2 or IRK4 scheme), the higher-order accurate $p = 3$ spatial scheme, provides best shape-retaining convection capability: the vortices produced around the corners of the triangle wedge keep their shapes far downstream. On the other hand, $p = 1$ scheme has evident dissipation as seen in the diffusion of the core of the vortices as they are convected downstream. While if we compare the computed results of the various time-integration schemes based on a same fixed spatial-discretization scheme, we can notice the IRK4 scheme is seen to be more suitable than the BDF2 scheme for the p=3 case (Fig. 12(b)) due to the fact that it effectly balances the spatial and temporal accuracy. In the case of p=1 solutions (Fiures 11(a) and 11(b)), since the spatial error is dominated in the solution of the IRK4 scheme, there is no evident impovements even the IRK4 scheme is utilized. From this point of view, it is concluded that high-order spatial discretizations demonstrate high accuracy when



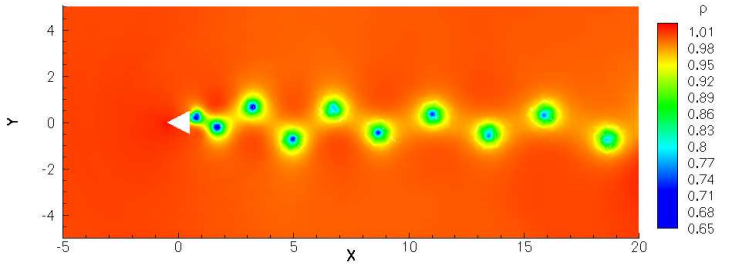
(a) $p = 1$,BDF2



(b) $p = 1$,IRK4

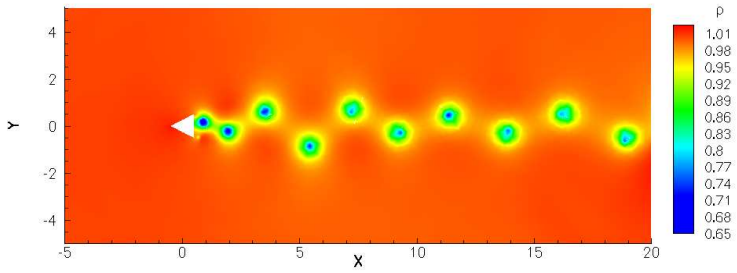


(c) $p = 2$,BDF2

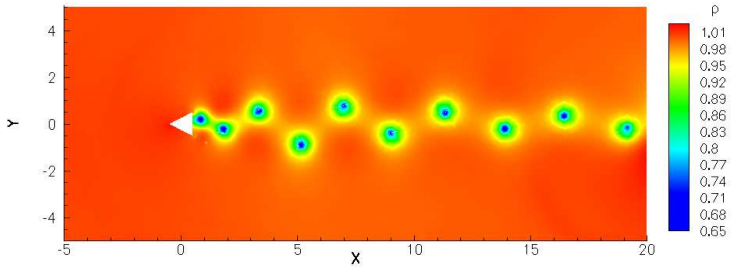


(d) $p = 2$,IRK4

Figure 11. density contours of p=1 and p=2 spatial-discretization schemes, at t=100, using BDF2 and IRK4 schemes



(a) $p = 3$,BDF2



(b) $p = 3$,IRK4

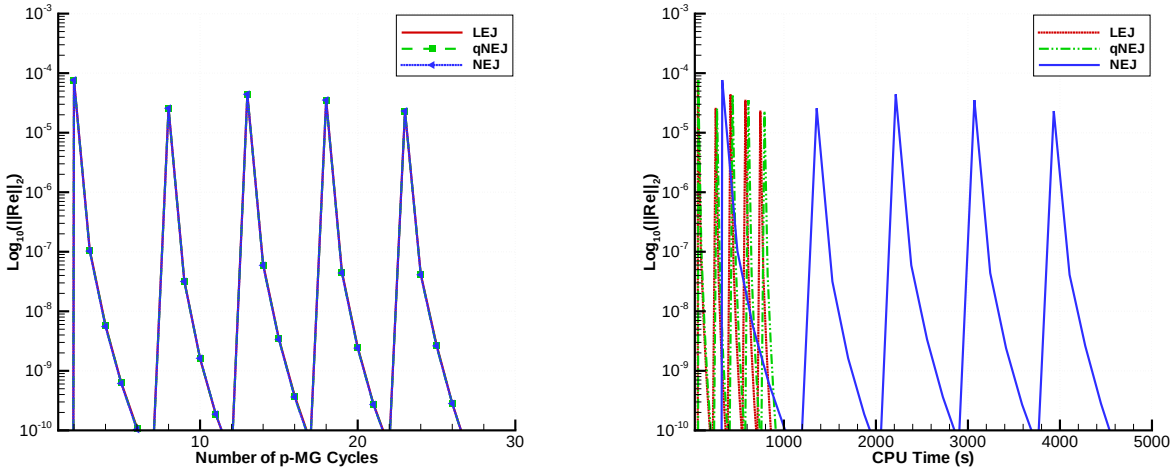
Figure 12. density contours of p=3 spatial-discretization scheme, at t=100, using BDF2 and IRK4 schemes

same temporal scheme is used, and high-order accurate time-integration schemes are exclusively fitful for high-order DG discretizations to achieve an overal high accuracy.

2. Comparisons of p-multigrid solvers

As we discussed in Section IV. A, several p-multigrid solver variants consisting of non-linear element Jacobi (NEJ), quasi-nonlinear element (qNEJ) and linearized element Jacobi (LEJ) can be used as a smoother in each implicit time step. In this section, the performance of these solver variants in one implicit time step is demonstrated using the IRK4 scheme and p=3 spatial discretization in the test case of shedding vortices.

Fig. 13(a) depicts the convergence rates of the LEJ, qNEJ and NEJ solvers in terms of p-multigrid cycles in the process of the next time step solution. Since five-stage unsteady-state residuals ($\mathbf{R_e}(\mathbf{u}^{(s)})$, $s = 2, \cdots, 6$) require to get convergence in one implicit time step of the IRK4 scheme, the results shown in this figure is distinct from that of the BDF schemes in which only one implicit solver is utilized in each implicit time step. As expected, the results are similar with the ones when tested in the steady-state solver,[24] the linearized element Jacobi, quasi-nonlinear element Jacobi and non-linear element Jacobi converge with similar rates in terms of the number of p-multigrid cycles. While if compared with the CPU time, as shown in Fig. 13(b), the LEJ and qNEJ solvers are seen to be substantially more efficient than the NEJ solver due to its expensive computation on the Jacobian evolutions. Since in this test case, 5 subiterations have been employed for all the three solver variants, it then turns out that LEJ and qNEJ have 5 times fewer computations spent on the evolutions of Jacobian matrix (i.e. $[D_T]$), thus the result illustrates the NEJ solver costs near 5 times longer in CPU time than the other two solvers. Moreover, the qNEJ solver has been found to be a specially stable and efficient solver when dealing with strong non-linearity and thus it is exclusively utilized in this work.



(a) convergence rate in terms of p-multigrid cycles

(b) convergence rate in terms of cpu time

Figure 13. Comparisons of convergence rates for solver variants (LEJ, qNEJ, NEJ) in one time step of IRK4 scheme on a unstructed mesh size of 10836 elements and order p=3

VI. Conclusions

In this paper, Discontinuous Galerkin methods, which make use of element basis functions that are discontinuous across element interfaces, have been employed in the spatial discretization of the two-dimensional unsteady Euler equations to reduce the gridding requirements and to achieve high spatial resolution. Several implicit schemes including BDF1, BDF2 and IRK4 schemes are employed in the time-integration of the unsteady Euler equations. The temporal accuracy and efficiency of these implicit schemes have been

investigated, focusing on the efficiency of fourth-order implicit Runge-Kutta schemes in comparison with the popular Backward Differencing formulations. It is concluded that the IRK4 scheme, which represents an attempt to balance the spatial and temporal orders of accuracy, can outperform the backward methods (BDF1 and BDF2) to achieve a desired error level in an efficient way. Meanwhile, The p-multigrid strategy, which uses element Jacobi as a smoother to solve non-linear problems on each multigrid level, can produce h-independent convergence rates, with only slight dependence on the time-step size, for implicit time-stepping problems. Thus it provides an efficient approach for solving implicit problems particularly on large meshes and with large time-steps. Further work will focus on extending this approach to the unsteady Navier-Stokes equations and including the presence of dynamically deforming meshes.

References

[1]F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method of the numerical solution of the compressible navier-stokes equation. *J. Comput. Phys.*, 131:267–279, 1997.

[2]F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of 2d euler equation. *J. Comput. Phys.*, 138:251–285, 1997.

[3]P. Batten, N. Clarke, C. Lambert, and D. M. Causon. On the choice of wavespeeds for the hllc riemann solver. *SIAM J. Sci. Comput.*, 18 (2):15531570, 1997.

[4]B.Berde and M. Borrel. Numerical experiments on the accuracy of a discontinuous galerkin method for the euler equations. *Aerosp. Sci. Technol.*, 5:279–288, 1998.

[5]H. Bijl, M. H. Carpenter, and V. N. Vatsa. Time integration schemes for the unsteady navier-stokes equations. In *the 15th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2001-2612, 2001.

[6]H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy. Implicit time integration schemes for the unsteady compressible navier stokes equations: Laminar flow. *J. Comput. Phys.*, 179:313–329, 2002.

[7]B. Cockburn. Devising discontinuous galerkin methods for non-linear hyperbolic conservation laws. *J. Comput. Phys.*, 128:187–204, 2001.

[8]B. Cockburn and C.-W. Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Appl. Mech. Engrg.*, 35:2440–2463, 1998.

[9]B. Cockburn and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *SIAM J. Sci. Comput.*, 16:173–261, 2001.

[10]F. Davoudzadeh, H. Mcdonald, and B. E. Thompson. Accuarcy evaluation of unsteady cfd numerical schemes by vortex preservation. *Computers and Fluids*, 24:883–895, 1995.

[11]V. Dolejsi and M. Feistauer. A semi-implicit discontinuous galerkin finite element method for the numerical solution of invisicid compressible flow. *J. Comput. Phys.*, 198:727–746, 2004.

[12]D. A. Dunavant. Economical symmetrical quadrature rules for complete polynomials over a square domain. *Int. J. Numer. Meth. Engng.*, 21:1777–1784, 1985.

[13]D. A. Dunavant. Higher degree efficient symmetrical gaussian guadrature rules for the triangle. *Int. J. Numer. Meth. Engng.*, 21:1129–1148, 1985.

[14]K. J. Fidkowski and D. L. Darmofal. Development of a higher-order solver for aerodynamic applications. In *Proceedings of the 42nd Aerospace Sciences Meeting and Exihibit, Reno NV*, 2004. AIAA Paper 2004–0436.

[15]K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-multigrid solution of high-order discontinuous galerkin discretizations of the compressible navier-stokes equations. *submitted to J. Comput. Phys.*, 2004.

[16]J. E. Flaherty, L. Krivodonova, J. Remacle, and M. S. Shephard. Aspects of discontinuous galerkin methods for hyperbolic conservation laws. *Finite Elements Anal. Design*, 38:889–908, 2002.

[17]E. Garnier, P. Sagaut, and M. Deville. A class of explicit eno filters with application to unsteady flows. *J. Comput. Phys.*, 170:184–204, 2000.

[18]F. Graham, J. Carey, and T. Oden. *Finite Elements A Second Course*, volume 2. 89-95, 1983.

[19]B. Helenbrook, D. J. Mavriplis, and Atkins. Analysis of "p"-multigrid for continuous and discontinuous finite element discretizations. In *Proceedings of the 16th AIAA Computational Fluid Dymamics Conference*. AIAA 2003-3989, 2003.

[20]G. Jothiprasad, D. J. Mavriplis, and D. A. Caughey. Higher order time integration schemes for the unsteady navier-stokes equations on unstructured meshes. *J. Comput. Phys.*, 191(2):542–566, Nov. 2003.

[21]H. Luo, J. D. Baum, and R. Lohner. A p-multigrid discontinuous galerkin method for the euler equations on unstructured grids. *J. Comput. Phys.*, 211:767–783, 2006.

[22]D. J. Mavriplis. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *J. Comput. Phys.*, 175:302–325, 2002.

[23]C. R. Nastase and D. J. Mavriplis. High-order discontinuous galerkin methods using an hp-multigrid approach. *J. Comput. Phys.*, (In Press).

[24]C. R. Nastase and D. J. Mavriplis. High-order discontinuous galerkin methods using a spectral multigrid approach. In *the 43th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2005-1268, 2005.

[25]P. Rasetarinera and M. Y. Hussaini. An efficient implicit discontinuous spectral galerkin method. *J. Comput. Phys.*, (172):718–738, 2001.

[26]J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. 1977.

[27]E. F. Toro, M. spruce, and W. Spears. Restoration of the contact surface in the hll-riemann solver. *Shock Waves*, 4:25–34, 1994.

[28]T. C. Warburton, I. Lomtev, Y. Du, S. Sherwin, and G. Karniadakis. Galerkin and discontinuous galerkin spectral/hp methods. *Comput. Methods Appl. Mech. Engrg.*, (175):343–359, 1999.

[29]H. C. Yee, N. D. Sandham, and M. J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *J. Comput. Phys.*, 150:199–238, 1999.

[30]Z. Zhang, S. T. J. Yu, and S. Chang. A space-time conservation element and solution element method for solving the two- and three-dimensional unsteady euler equations using quadrilateral and hexahedral meshes. *J. Comput. Phys.*, 175, 2002.