Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 225 (2007) 1994–2015
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations

Li Wang *, Dimitri J. Mavriplis

Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071-3295, United States

Received 13 February 2006; received in revised form 20 October 2006; accepted 1 March 2007
Available online 14 March 2007

Abstract

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 second-order Crank–Nicholson (CN2) and a fourth-order implicit Runge–Kutta (IRK4) scheme 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 Crank–Nicholson methodology, although not L-stable, demonstrates superior performance compared to the BDF2 scheme for the problems chosen in this work. However, the fourth-order accurate implicit Runge–Kutta scheme is found to be the most efficient in terms of computational cost for a given accuracy level as compared to the lower-order schemes, in spite of the added cost per time step, and the benefits of this scheme increase for tighter error tolerances.

Keywords: Discontinuous Galerkin methods; High-order methods; Time-dependent; Implicit time-integration; *p*-multigrid methods; Time accuracy; Compressible flow

1. Introduction

In recent years there has been a growing interest in the use of discontinuous Galerkin (DG) methods for obtaining high-accuracy solutions of systems of conservation laws. From a simple linear advection problem to the more complicated non-linear compressible Euler and Navier–Stokes equations [1–9], 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 Riemann solver techniques, which are now well understood and have been extensively developed in the context of finite volume methods.

* Corresponding author. Tel.: +1 307 399 6550; fax: +1 307 766 2695.
E-mail addresses: wangli@uwyo.edu (L. Wang), mavripl@uwyo.edu (D.J. Mavriplis).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.03.002

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 [5,8,10]. 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 for steady-state problems.

The CFL-stability limit of explicit schemes applied to high-order spatial discretizations becomes more restrictive as the order of the spatial discretization increases. In general, the stability limit depends on the specific basis functions employed, but can be as severe as CFL $< \frac{1}{p^2}$ for spectral basis functions [11], where p represents the polynomial degree of the basis functions. As a consequence, implicit time-integration strategies, which are unconditionally stable and allow the selection of the time step based purely on temporal accuracy considerations, [12–14] can be expected to provide a more effective approach for problems with disparate length and time scales. Recently, many efforts have been made especially on the use of high-order implicit time-integration methods [13,15,16] for solution procedures. However, 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 on implicit time-integration approaches of various orders, namely first and second-order implicit backwards differencing schemes (BDF1 and BDF2), the second-order Crank–Nicholson scheme (CN2), and an implicit fourth-order Runge–Kutta scheme (IRK4), for solving the Euler equations with high-order spatial DG discretizations. 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 manner. This is achieved using a spectral multigrid (p-MG) method [17,6,18,9]. The p-multigrid 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 nearly h and p independent convergence rates [9].

The objectives of this work are twofold. On the one hand, we extend the p-multigrid solution strategy to time-implicit problems, and examine the performance of this solution strategy with regards mesh size and time-step size. The second objective is to examine the accuracy and efficiency of higher-order temporal discretizations, such as implicit Runge–Kutta discretizations, compared to lower-order temporal discretizations such as BDF2 and Crank–Nicolson schemes, with particular emphasis on the suitability of these schemes in terms of accuracy and efficiency for spatial discretizations of various orders.

The organization of this paper is as follows. The governing equations to be solved are described in Section 2. Section 3 describes the spatial and temporal discretizations used in this work. Section 4 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 5, two-dimensional numerical results are shown for an isentropic convecting vortex case, and for a periodic vortex-shedding problem using an unstructured triangular mesh. Finally, Section 6 summarizes the conclusions of this paper.

2. Governing equations

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

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

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

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{Bmatrix}, \quad \mathbf{f} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{Bmatrix}, \quad \mathbf{g} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \end{Bmatrix} \tag{2}$$

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 as

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

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 Eqs. (4) and (5), respectively, where $\partial\Omega$ represents the boundary of Ω:

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

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

3. Discretizations

3.1. 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 [19]. Thus, in this approach, the computational domain Ω is partitioned into an ensemble of non-overlapping elements, i.e. $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_{n\text{Elem}}$, where nElem denotes the number of elements in the domain. Assuming that Ω_m is a typical triangulation element of Ω, the weak formulation is obtained by multiplying by a test function, ζ_j, and integrating over space as

$$\int_{\Omega_m} \zeta_j \left[\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u}(\mathbf{x}, t))}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u}(\mathbf{x}, t))}{\partial y}\right] \mathrm{d}V = 0 \tag{6}$$

and then, integrating by parts:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_m} \zeta_j \mathbf{u}_h \,\mathrm{d}V - \int_{\Omega_m} \left[\frac{\partial \zeta_j}{\partial x}\mathbf{f}(\mathbf{u}_h) + \frac{\partial \zeta_j}{\partial y}\mathbf{g}(\mathbf{u}_h)\right] \mathrm{d}V + \int_{\partial\Omega_m} \zeta_j [\mathbf{f}(\mathbf{u}_h) n_x + \mathbf{g}(\mathbf{u}_h) n_y] \mathrm{d}S = 0 \tag{7}$$

where the unit normal vector $\mathbf{n} = (n_x, n_y)$ is outward to the boundary, and points into the computational domain. $\boldsymbol{u}_h$ represents the finite-element solution approximation which is expanded as

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

In the above equation, ζ_k represents the basis functions which are taken as the same set as the test functions in Eq. (7) in this Galerkin formulation. The current implementation in this work uses a set of hierarchical basis functions which are rotationally invariant within the standard isoparametric triangle with $\{0 < \xi, \eta < 1\}$ [9,20]. To briefly summarize, since the first-order Lagrange polynomials are given as

$$L_1 = 1 - \xi - \eta, \quad L_2 = \xi, \quad L_3 = \eta \tag{9}$$

then, the hierarchical basis set, $\{\zeta_k\}$, is fully described by *vertex*:

$$\zeta_1^{\mathrm{v}} = L_1, \quad \zeta_2^{\mathrm{v}} = L_2, \quad \zeta_3^{\mathrm{v}} = L_3 \tag{10}$$

edge:

$$\begin{aligned} \zeta_n^{\mathrm{e1}} &= L_1 L_2 \psi_{n-2}(L_2 - L_1) \\ \zeta_n^{\mathrm{e2}} &= L_2 L_3 \psi_{n-2}(L_3 - L_2) \\ \zeta_n^{\mathrm{e3}} &= L_3 L_1 \psi_{n-2}(L_1 - L_3) \end{aligned} \tag{11}$$

and *bubble*:

$$\zeta_{n1,n2}^{\mathrm{b}} = L_1 L_2 L_3 \psi_{n1-1}(L_2 - L_1)\psi_{n2-1}(L_1 - L_3) \tag{12}$$

shape functions, where $2 \leqslant n \leqslant p^{\mathrm{e}}$, $n_1 + n_2 = p^{\mathrm{b}} - 1$ and $n_1, n_2 \geqslant 1$. The kernel functions $\psi(z)$ are given as,

$$\psi_{n-2}(z) = \frac{-2\sigma}{n-1} P^{1,1}_{n-2}(z) \tag{13}$$

where $P^{\alpha,\beta}_n$ represents the Jacobi polynomial of order, n, with weights α and β. In our discretization, the edge order, p^{e}, and the bubble order, p^{b}, are set to be the discretization order within the element ($p^{\mathrm{e}} = p^{\mathrm{b}} = p$), where p is the discretization order within the element. For $p \geqslant 2$ the basis functions within the standard triangle, $\{\phi_i, i = 4 \ldots M\}$, are normalized Lobatto (*i.e.* $\phi_{n\geqslant 2} = \sigma \int_{-1}^{x} P^{0,0}_{n-1}(z)\mathrm{d}z$) functions [20], which take zero values at the end of their definition interval. The normalization factor, σ, can be used to condition the mass or convection matrices. Presently, this set of shape functions enables the finite-element solution approximations ranging from first order ($p = 0$), up to fifth order ($p = 4$) spatial accuracy.

Substituting the solution expansion into Eq. (7), we then obtain a set of N equations given as

$$\frac{\mathrm{d}\mathbf{u}_k(t)}{\mathrm{d}t} \int_{\Omega_m} \zeta_j \zeta_k \,\mathrm{d}V - \int_{\Omega_m} \left[\frac{\partial \zeta_j}{\partial x} \mathbf{f}(\mathbf{u}_h) + \frac{\partial \zeta_j}{\partial y} \mathbf{g}(\mathbf{u}_h) \right] \mathrm{d}V + \int_{\partial\Omega_m} \zeta_j [\mathbf{f}(\mathbf{u}_h) n_x + \mathbf{g}(\mathbf{u}_h) n_y] \mathrm{d}S = 0 \quad j = 1, 2, \ldots, N \tag{14}$$

where the repeated index k in Eq. (14) denotes the summation convention ranging from 1 to N, and $\mathbf{u}_k(t)$ 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 an approximate Riemann flux function $\mathrm{Flux}(\mathbf{u}^L_h, \mathbf{u}^R_h, \mathbf{n})$, where $\mathbf{u}^L_h$ and $\mathbf{u}^R_h$ represent the internal element interface solution variables and neighboring element interface solution variables, respectively. In this work, the HLLC approximate Riemann solver [21,22] is employed due to its efficiency and robustness. All the integrals including the domain and boundary integrals in Eq. (14) are evaluated by use of Gauss quadrature formulae [23,24] 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. (14) are those satisfied with integrals exactly on polynomials of order $2p$, $2p$ and $2p + 1$. Because the fluxes in the last two integrals are nonlinear functions of the state variables (which are the variables expressed in term of the basis functions), an additional order of accuracy in the quadrature (e.g. $2p + 1$ vs. $2p$ in the last integral) is used as recommended in Ref. [25]. Although this integration is not exact, it has been shown to be sufficient for retaining the design order of accuracy of the discretization in previous work [9].

By grouping together the time-dependent and spatial contributions, Eq. (14) can be written as the following ODE form,

$$M \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{R}_{\mathbf{p}}(\mathbf{u}) = 0 \tag{15}$$

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

$$M_{\mathrm{d}} = \begin{pmatrix} \int_{\Omega_m} \zeta_1\zeta_1 \,\mathrm{d}V & \int_{\Omega_m} \zeta_1\zeta_2 \,\mathrm{d}V & \cdots & \int_{\Omega_m} \zeta_1\zeta_N \,\mathrm{d}V \\ \int_{\Omega_m} \zeta_2\zeta_1 \,\mathrm{d}V & \int_{\Omega_m} \zeta_2\zeta_2 \,\mathrm{d}V & \cdots & \int_{\Omega_m} \zeta_2\zeta_N \,\mathrm{d}V \\ \cdots & \cdots & \cdots & \cdots \\ \int_{\Omega_m} \zeta_N\zeta_1 \,\mathrm{d}V & \int_{\Omega_m} \zeta_N\zeta_2 \,\mathrm{d}V & \cdots & \int_{\Omega_m} \zeta_N\zeta_N \,\mathrm{d}V \end{pmatrix} \tag{16}$$

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

3.2. 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 discretiza-

tions, 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 thus more suitable for stiff problems. 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), the second-order accurate Crank–Nicolson or trapeziodal scheme, and a fourth-order accurate implicit multistage Runge–Kutta scheme (IRK4).

The BDF1 and BDF2 schemes are both unconditionally stable, while the CN2 scheme is A-stable, but not L-stable [26]. For these reasons, CN2 has often been shunned in favor of BDF2 in many computational fluid dynamics problems. However, the lack of L-stability may be acceptable particularly for problems with smooth solutions, and the scheme is therefore included in the present study. Because higher-order multistep backwards difference schemes beyond second-order are not A-stable, we choose to investigate the use of implicit Runge–Kutta schemes for achieving higher temporal accuracy. A six-stage diagonally implicit IRK scheme is chosen which is fourth-order accurate in time. While this scheme may not necessarily represent the optimal fourth-order temporal scheme for all problems, it has been designed with stiff stability and accuracy considerations in mind [27], and has been used successfully on lower-order finite volume schemes by various authors [13,28]. One of the drawbacks of IRK methods is their expense, since these require the solution of multiple implicit problems at each time step (one per stage), as opposed to BDF and CN schemes which only require the solution of a single implicit problem per time step. Therefore, one of the objectives of this work is to determine if IRK schemes can be competitive or superior to lower-order schemes when used in conjunction with efficient solvers, particularly when high accuracy is required.

Starting from the set of ordinary differential equations given by Eq. (15), the formulations for BDF1, BDF2 and CN2 schemes are given respectively as

$$\frac{M}{\Delta t}(\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) + \mathbf{R_p}(\mathbf{u}_h^{n+1}) = 0 \tag{17}$$

$$\frac{M}{\Delta t}\left(\frac{3}{2}\mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \frac{1}{2}\mathbf{u}_h^{n-1}\right) + \mathbf{R_p}(\mathbf{u}_h^{n+1}) = 0 \tag{18}$$

$$\frac{M}{\Delta t}(\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) + \frac{1}{2}\mathbf{R_p}(\mathbf{u}_h^{n+1}) + \frac{1}{2}\mathbf{R_p}(\mathbf{u}_h^n) = 0 \tag{19}$$

where Δt represents the integration time step, and $\mathbf{u}_h^n$ and $\mathbf{u}_h^{n+1}$ are numerical solutions for the current and the next (unknown) time step, respectively. By defining a non-linear unsteady residual, $\mathbf{R_e}$, for the corresponding BDF1, BDF2 and CN2 schemes as

$$\text{BDF1:}\quad \mathbf{R_e}(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t}\mathbf{u}_h^{n+1} + \mathbf{R_p}(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t}\mathbf{u}_h^n = 0 \tag{20}$$

$$\text{BDF2:}\quad \mathbf{R_e}(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t}\left(\frac{3}{2}\mathbf{u}_h^{n+1}\right) + \mathbf{R_p}(\mathbf{u}_h^{n+1}) - \frac{M}{\Delta t}(2\mathbf{u}_h^n - \frac{1}{2}\mathbf{u}_h^{n-1}) = 0 \tag{21}$$

$$\text{CN2:}\quad \mathbf{R_e}(\mathbf{u}_h^{n+1}) = \frac{M}{\Delta t}\mathbf{u}_h^{n+1} + \frac{1}{2}\mathbf{R_p}(\mathbf{u}_h^{n+1}) - \left(\frac{M}{\Delta t}\mathbf{u}_h^n - \frac{1}{2}\mathbf{R_p}(\mathbf{u}_h^n)\right) = 0 \tag{22}$$

the solution of these schemes can be achieved by solving the non-linear problems $\mathbf{R_e}(\mathbf{u}_h^{n+1}) = 0$ at each time step.

These schemes are relatively efficient because they solve only one implicit set of equations per time step. In the case of multistage IRK schemes, however, multiple implicit problems are required per time step (each stage works on one implicit system solution), 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. In this paper, we utilize the ESDIRK class of RK schemes, which corresponds to Explicit first stage, Single Diagonal coefficient, diagonally Implicit Runge–Kutta. The formula for a $\mathscr{S}$-stage ESDIRK scheme, in the case of the Euler equations, can be written as

Table 1
Butcher tableau for ESDIRK class of six-stage RK schemes

$c_1 = 0$	0	0	0	0	0	0
c_2	a_{21}	$a_{22} = a_{66}$	0	0	0	0
c_3	a_{31}	a_{32}	$a_{33} = a_{66}$	0	0	0
c_4	a_{41}	a_{42}	a_{43}	$a_{44} = a_{66}$	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	$a_{55} = a_{66}$	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

$$\begin{aligned} &\text{(i)}\ \ \mathbf{u}^{(0)} = \mathbf{u}_h^n \\ &\text{(ii)}\ \text{For } s = 1, \ldots, \mathscr{S} \\ &\qquad \mathbf{u}^{(s)} = \mathbf{u}^n - \Delta t \sum_{j=1}^{s} a_{sj} M^{-1} \mathbf{R}_\mathbf{p}(\mathbf{u}^{(j)}) \\ &\text{(iii)}\ \mathbf{u}_h^{n+1} = \mathbf{u}^{(\mathscr{S})} \end{aligned} \tag{23}$$

where a_{sj} are the Butcher coefficients of the scheme. The Butcher table for the six-stage ESDIRK scheme ($\mathscr{S} = 6$, fourth-order accurate) employed presently is shown in Table 1 and the values are given in Appendix A (from Ref. [13,28]). The set of coefficients, a_{sj}, defines the implicit RK schemes (as shown in Eq. (23)). 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

$$c_k = \sum_{j=1}^{k} a_{kj} \quad (k = 1, 2, \ldots, 6) \tag{24}$$

Similarly, the unsteady residual corresponding to the non-linear implicit system at each stage of the IRK4 scheme can be written as

$$\mathbf{R}_\mathbf{e}(\mathbf{u}^{(s)}) = \frac{M}{\Delta t}\mathbf{u}^{(s)} + a_{ss}\mathbf{R}_\mathbf{p}(\mathbf{u}^{(s)}) - \left[\frac{M}{\Delta t}\mathbf{u}^n - \sum_{j=1}^{s-1} a_{sj}\mathbf{R}_\mathbf{p}(\mathbf{u}^{(j)})\right] = 0 \quad (s = 1, \ldots, 6) \tag{25}$$

4. *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 [6,8,9,18], in conjunction with an 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 [29], but instead of using physically fewer elements on coarser levels, lower-order approximations serve as "coarse" levels while the same spatial grid elements are used on all levels, as shown in (Fig. 1). Therefore, with this methodology, no additional grid information needs to be stored. In previous work, we have examined the combination of *p* and *h*-multigrid methods for obtaining an optimal solution strategy which delivers h and p independent convergence rates [9]. In the current work, we restrict ourselves to the use of *p*-multigrid alone, mainly for simplicity, with the understanding that our results should extend naturally to combined *h*-*p*-multigrid methods. Furthermore, the addition of *h*-multigrid can be expected to be less important for time-dependent problems, since the resulting implicit systems are more diagonally dominant and are somewhat more local in nature than the corresponding steady-state problem.

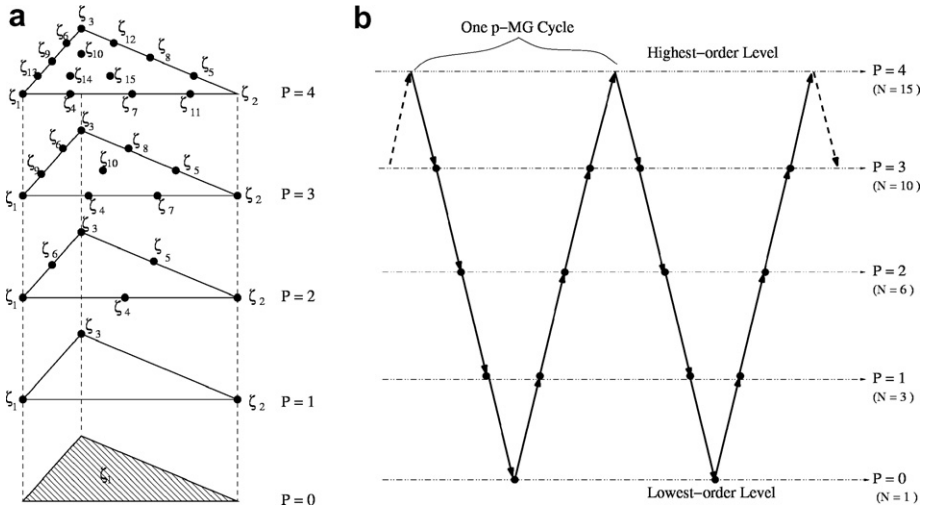

Fig. 1. Illustration of a five level p-multigrid algorithm.

4.1. Element Jacobi smoother

At each p-multigrid level, an element Jacobi scheme is used as a smoother. Returning to Eq. (20), for example, in order to solve this implicit problem for the $n+1$ time step, we consider the use of a Newton scheme given by

$$\begin{aligned}
&\text{(i)} \quad \mathbf{w}^1 = \mathbf{u}_h^n \\
&\text{(ii)} \quad \left[\frac{\partial \mathbf{R_e}}{\partial \mathbf{w}}\right]^k \Delta \mathbf{w}^{k+1} = -\mathbf{R_e}(\mathbf{w}^k) \\
&\qquad \mathbf{w}^{k+1} = \mathbf{w}^k + \alpha \Delta \mathbf{w}^{k+1}, \quad k = 1, 2, \ldots, \mathscr{M} \\
&\text{(iii)} \quad \mathbf{u}_h^{n+1} = \mathbf{w}^{\mathscr{M}}, \quad \text{when } \mathbf{R_e}(\mathbf{w}^{\mathscr{M}}) = 0
\end{aligned} \tag{26}$$

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 for clarity, and $\mathbf{w}$ represents intermediate solutions in the Newton solver process for the solution of $\mathbf{u}_h^{n+1}$. Additionally, α is a relaxation parameter designed to keep $\|\alpha \Delta \mathbf{w}^{k+1}\|_{L_\infty} / \|\mathbf{w}^{k+1}\|_{L_\infty} \leqslant 10\%$.

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 the non-linear element Jacobi solver (NEJ). The quasi-non-linear element Jacobi (qNEJ) approach represents a variant of the NEJ solver where the diagonal block Jacobians are frozen over a certain number of subiterations 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, i.e. $[\partial \mathbf{R_e}(\mathbf{w}^k)/\partial \mathbf{w}] = [D_T^k] + [O_T^k]$. 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 explicitly. This corresponds to the full linearization of the NEJ scheme. In general, five subiterations are used in both the (qNEJ) and (LEJ) schemes between non-linear Jacobian (and residual for the LEJ method) updates.

Note that the diagonal block matrix D_T, the off-diagonal blocks O_T and the unsteady residual $\mathbf{R_e}$ will have different forms for the various time-integration schemes. To make this point clear, we also list these terms for the corresponding BDF1, BDF2 and CN2 schemes, and for intermediate stage functions of the IRK4 scheme, respectively, as given in Table 2, where D^k and O^k denote the diagonal and off-diagonal blocks of the Jacobian matrix for the corresponding steady-state solver, $[\partial \mathbf{R_p}(\mathbf{w}^k)/\partial \mathbf{w}]$, in a similar manner.

Table 2
Diagonal blocks, off-diagonal blocks and unsteady residuals for BDF1, BDF2, CN2 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	$\frac{3}{2}\frac{M}{\Delta t}\mathbf{w}^k + \mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}(2\mathbf{u}_h^n - \frac{1}{2}\mathbf{u}_h^{n-1})$
CN2	$D_T = \frac{M}{\Delta t} + \frac{1}{2}D^k$	$\frac{1}{2}O^k$	$\frac{M}{\Delta t}\mathbf{w}^k + \frac{1}{2}\mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}\mathbf{u}_h^n - \frac{1}{2}\mathbf{R_p}(\mathbf{u}^n)$
IRK4 ($s = 1,\ldots,6$)	$D_T = \frac{M}{\Delta t} + a_{ss}D^k$	$a_{ss}O^k$	$\frac{M}{\Delta t}\mathbf{w}^k + a_{ss}\mathbf{R_p}(\mathbf{w}^k) - \mathbf{F}$ $\mathbf{F} = \frac{M}{\Delta t}\mathbf{u}_h^n - \sum_{j=1}^{s-1} a_{sj}\mathbf{R_p}(\mathbf{u}^{(j)})$

4.2. 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), for the case of a fifth-order accurate discretization scheme, the finest multigrid level consists of the original $p = 4$ (fifth-order accurate) spatial 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 levels to low-order levels 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 levels 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 previous work we have investigated the use of linear and non-linear multigrid strategies [9]. In this work, we make use exclusively of the non-linear approach, using the full approximation storage (FAS) [9,30] multigrid scheme to solve the non-linear problem at each multigrid level. The FAS formulation for p-multigrid is identical to the original FAS formulation derived for h-multigrid problems [31]. The procedure of a two-level V-cycle FAS p-MG approach is described below, noting that we use the notation $\mathbf{w}_p^m$ to denote the intermediate solution in the time interval $[n\Delta t, (n+1)\Delta t]$, where the superscript represents the number of p-MG cycles and the subscript represents the p-MG level.

- **Step 1:** Perform τ_1 subiterations on the high-order approximation level (p) to solve the problem: $\mathbf{R}_{\mathbf{e}_p}(\mathbf{w}_p^m) = \mathbf{S}_p$, using any Element Jacobi solver mentioned previously; get residual: $\mathbf{r}_p^m = \mathbf{S}_p - \mathbf{R}_{\mathbf{e}_p}(\mathbf{w}_p^m)$
- **Step 2:** Restrict both residual and solution to the low-order approximation level $(p-1)$: $\mathbf{S}_{p-1} = I_p^{p-1}\mathbf{r}_p^m$; $\mathbf{w}_{p-1}^m = I_p^{p-1}\mathbf{w}_p^m$
- **Step 3:** Solve the low-order approximation level problem by using the same Element Jacobi solver with τ_2 subiterations: $\mathbf{R}_{\mathbf{e}_{p-1}}(\mathbf{w}_{p-1}^m) = \mathbf{S}_{p-1}$; Obtain the low-order level error: $\mathbf{e}_{p-1}^m = \mathbf{w}_{p-1}^m - I_p^{p-1}\mathbf{w}_p^m$
- **Step 4:** Prolongate this low-order level error to correct the high-order approximation level: $\mathbf{w}_p^{m+1} = \mathbf{w}_p^m + I_{p-1}^p\mathbf{e}_{p-1}^m$
- **Step 5:** Repeat Steps 1–4 for each V-cycle of the p-MG method until $\mathbf{R}_{\mathbf{e}_p}(\mathbf{w}_p^{m+1}) = 0$ (machine zero or a suitably determined tolerance), then the solution for the $n+1$ time step is obtained as: $\mathbf{u}_h^{n+1} = \mathbf{w}_p^{m+1}$

Note that for the p-multigrid method, the same restriction operator is used for both residual and solution restriction. The source term $\boldsymbol{S}_p$, which represents the residual restriction term from the finer multigrid levels, vanishes on the finest level (highest-order approximation level) in the multigrid formulation, but is retained on all levels in the description for generality. The two-level multigrid scheme described above is used recursively to solve the coarse level problem, resulting in the full multilevel algorithm. Furthermore, in this paper, a constant number of five subiterations is employed on all multigrid levels for the element Jacobi smoothers. For the

qNEJ (and LEJ) smoothers, this requires the only one non-linear Jacobian (and non-linear residual for the LEJ scheme) evaluation for each visit to a given mesh level.

5. 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. The first test case consists of an isentropic convecting vortex on a uniform mesh, while the second test case consists of the time-dependent vortex shedding from a triangular wedge on a highly graded unstructured mesh.

For the first test case, we concentrate on assessing the accuracy and efficiency of the various time-integration schemes in the presence of high spatial accuracy ($p = 4$, fifth-order spatial accuracy). Accuracy is assessed on the one hand by comparing solutions using the same time step for the various schemes with each other and with an exact solution, in order to demonstrate that we are within (or close to) the asymptotic region of convergence for the higher-order time-integration methods for such problems. A more quantitative time step refinement study is then performed to demonstrate the asymptotic error reduction properties of the respective temporal schemes. In order to compare the efficiency of the various schemes, we first evaluate the performance of the *p*-multigrid solution strategy, including the dependencies on grid resolution and time step size, which are ultimately the key determining factors for any efficiency comparison of time-implicit schemes. We then examine the delivered accuracy of the various schemes as a function of cpu time.

The second test case represents a more realistic problem, for which an exact solution is not available. In this case, we investigate the effect of low and high-order accurate time-integration schemes combined with low and high-order spatial discretizations on highly graded meshes on overall solution accuracy. We begin by establishing the asymptotic convergence of the time-integration schemes on this more demanding problem, through a time-step refinement study, and then examine the overall solution error for various combinations of spatial and temporal accuracy. Finally, a qualitative comparison is made with a simple explicit scheme, to illustrate the advantages of time-implicit approaches for low-reduced frequency problems of this type.

5.1. Convection of an isentropic vortex

The convection of a 2D inviscid isentropic vortex [32–34] 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.

5.1.1. Numerical solutions

The mean flow density, ρ_∞, velocity, u_∞ and v_∞, pressure, p_∞ and temperature T_∞ are taken as freestream values, which are set as $(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 0.5, 0, 1, 1)$ in this test case. Freestream boundary conditions are imposed on the top and bottom boundaries, while periodic boundary conditions are applied between the inlet and outlet of the domain. These boundary conditions are applied on all levels of the multigrid sequence. At $t_0 = 0$, the flow is perturbed by an isentropic vortex $(\delta u, \delta v, \delta T)$ centered at (x_0, y_0) with the form:

$$\delta u = -\frac{\alpha}{2\pi}(y - y_0)\mathrm{e}^{\phi(1-r^2)} \tag{27}$$

$$\delta v = \frac{\alpha}{2\pi}(x - x_0)\mathrm{e}^{\phi(1-r^2)} \tag{28}$$

$$\delta T = -\frac{\alpha^2(\gamma - 1)}{16\phi\gamma\pi^2}\mathrm{e}^{2\phi(1-r^2)} \tag{29}$$

where ϕ and α are parameters which determine 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 air. In this study, we set ϕ as unity and α as 4.0. Given the perturbation functions shown in Eqs. (27)–(29), we can determine the other resulting conservative variables, assuming isentropic flow throughout the domain (i.e. $p/\rho^\gamma = 1$ and $T = p/\rho$ for a perfect gas):

$$\rho = T^{1/(\gamma-1)} = (T_\infty + \delta T)^{1/(\gamma-1)} = \left[1 - \frac{\alpha^2(\gamma-1)}{16\phi\gamma\pi^2}\mathrm{e}^{2\phi(1-r^2)}\right]^{1/(\gamma-1)} \tag{30}$$

$$u = u_\infty + \delta u = 0.5 - \frac{\alpha}{2\pi}(y - y_0)\mathrm{e}^{\phi(1-r^2)} \tag{31}$$

$$v = v_\infty + \delta v = 0 + \frac{\alpha}{2\pi}(x - x_0)\mathrm{e}^{\phi(1-r^2)} \tag{32}$$

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

$$\mathrm{CFL}_i = \frac{\Delta t}{\mathrm{vol}_i}\sum_{j=1}^{3\,\mathrm{edges}}(|\mathbf{u}\cdot\mathbf{n}| + c)_j \quad 1 \leqslant i \leqslant n\mathrm{Elem} \tag{33}$$

where vol denotes the area of the element in the 2D case and c is the local speed of sound, then the fixed time-step $\Delta t = 0.2$ corresponds to a maximum CFL number of 11. Note that for the considered spatial discretization ($p = 4$, fifth-order accurate), the explicit stability limit could be as much as 176 times smaller (i.e. $\sim \frac{1}{p^2}$) than our chosen time step.

5.1.2. Temporal accuracy

We begin by examining the error of the respective time-integration schemes at a fixed time-step size, in order to determine if these schemes are within or close to their asymptotic regions of convergence. This is important since the error properties of these schemes are asymptotic in nature, based on the presumption of smooth solutions, and there is no guarantee that higher-order methods will deliver smaller errors than lower-order methods, even at equivalent time steps, when these assumptions do not hold.

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 $u_\infty = 0.5$, 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 the BDF1, BDF2, CN2 and IRK4 schemes are shown in Figs. 3 and 4. The first three non-dimensional times 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). The results of the BDF1 scheme (Figs. 3a–d) illustrate how the vortex is diffused using this first-order accurate scheme, even at small time increments. Conversely, as shown in Figs. 3e–h and 4a–d, the BDF2 and CN2 schemes display a substantially better shape-retaining property for the vortex, except at later times such as $t = 50$ where some dispersion and oscillations are evident around the brink of the vortex. On the other hand, the IRK4 scheme (Figs. 4e–h) provides the best accuracy with the final shape of the vortex at $t = 50$ being almost indistinguishable from the initial shape.

A more quantitative comparison is given in Fig. 5, where the density profiles along the horizontal centerline ($y = 0$) for the BDF1, BDF2, CN2 and IRK4 time-integration schemes at various times, $t = 4$, $t = 10$, $t = 20$, and $t = 50$, are compared with the exact solution, obtained by translating the initial centerline density profile to the appropriate spatial location for each given time. From the figure, it can be seen that the IRK4 scheme

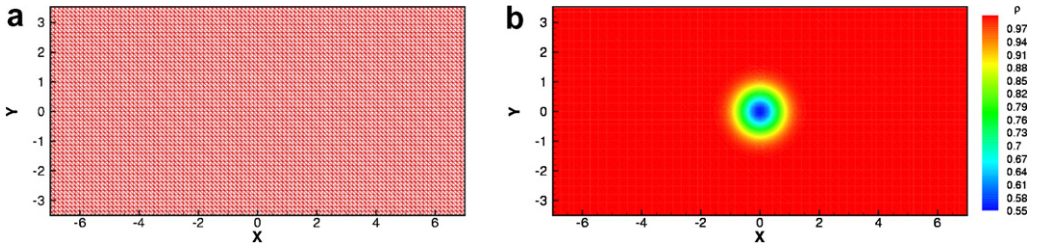

Fig. 2. Grid and initial density contours for isentropic vortex convection problem: (a) grid; (b) initial density contour (2D).

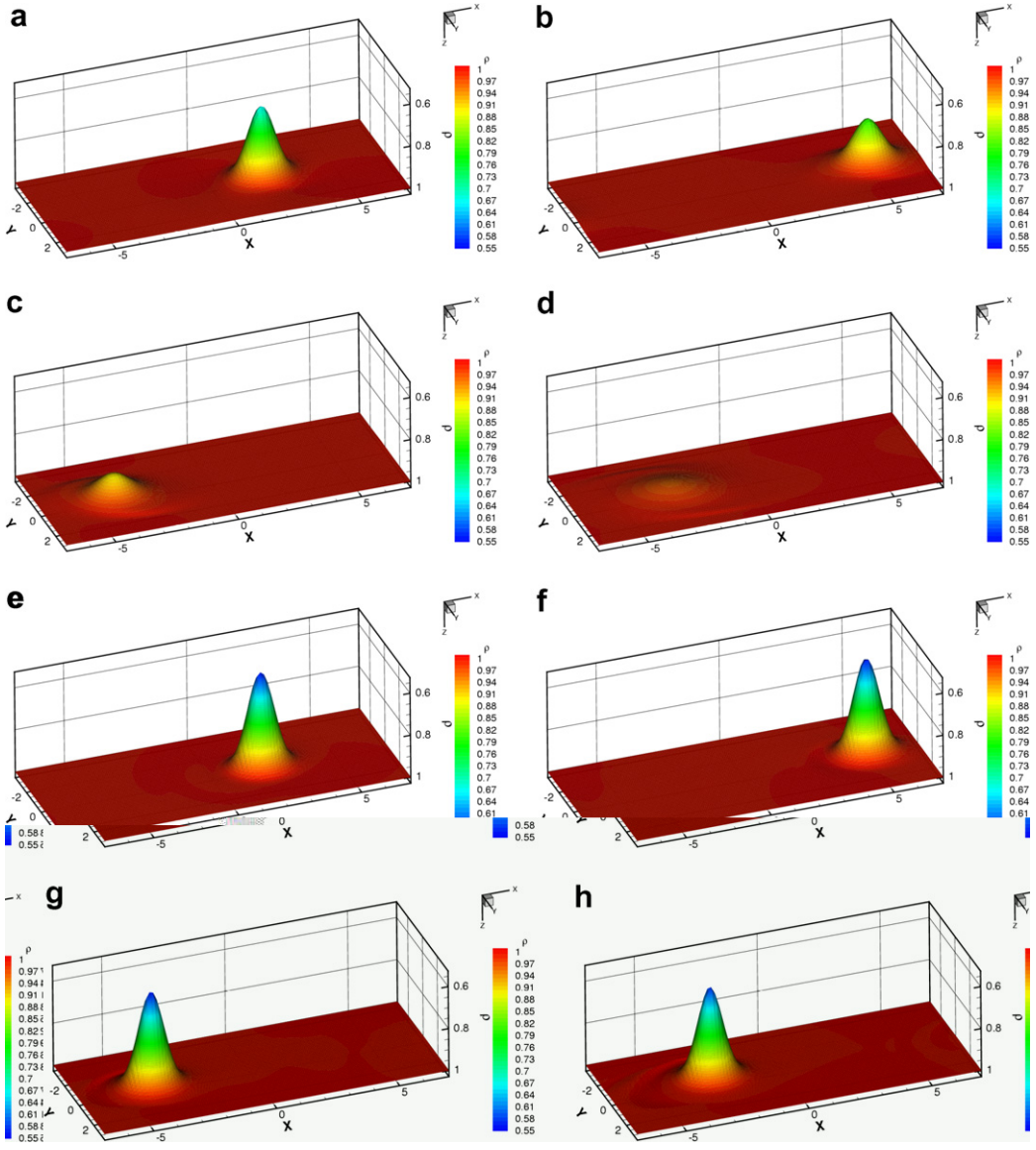

Fig. 3. Density contours (3D) of the BDF1 (a–d) and BDF2 (e–h) schemes at various times, $t = 4$ (a,e), 10 (b,f), 20 (c,g) and 50 (d,h), using a time-step of $\Delta t = 0.2$ and $p = 4$ spatial discretization.

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 rapid dissipation of the vortex core, as mentioned previously, while the BDF2 and CN2 schemes provide substantially better resolution and higher accuracy than the BDF1 scheme. Before $t = 20$, the computed results obtained by the BDF2 and CN2 schemes fall almost on top of the exact solution, but at later times, these schemes show increased deviations from the exact profile, although the CN2 scheme remains substantially more accurate than the BDF2 scheme. However, this result must be balanced by the fact that the CN2 scheme is not L-stable [26], and may perform poorly in other cases.

In order to assess the asymptotic behavior of the time-integration schemes, a temporal refinement study is carried out using the same fixed spatial discretization ($p = 4$). Because the overall error is due to both spatial and temporal error, the "exact" numerical solution for each temporal scheme is obtained using a small time-

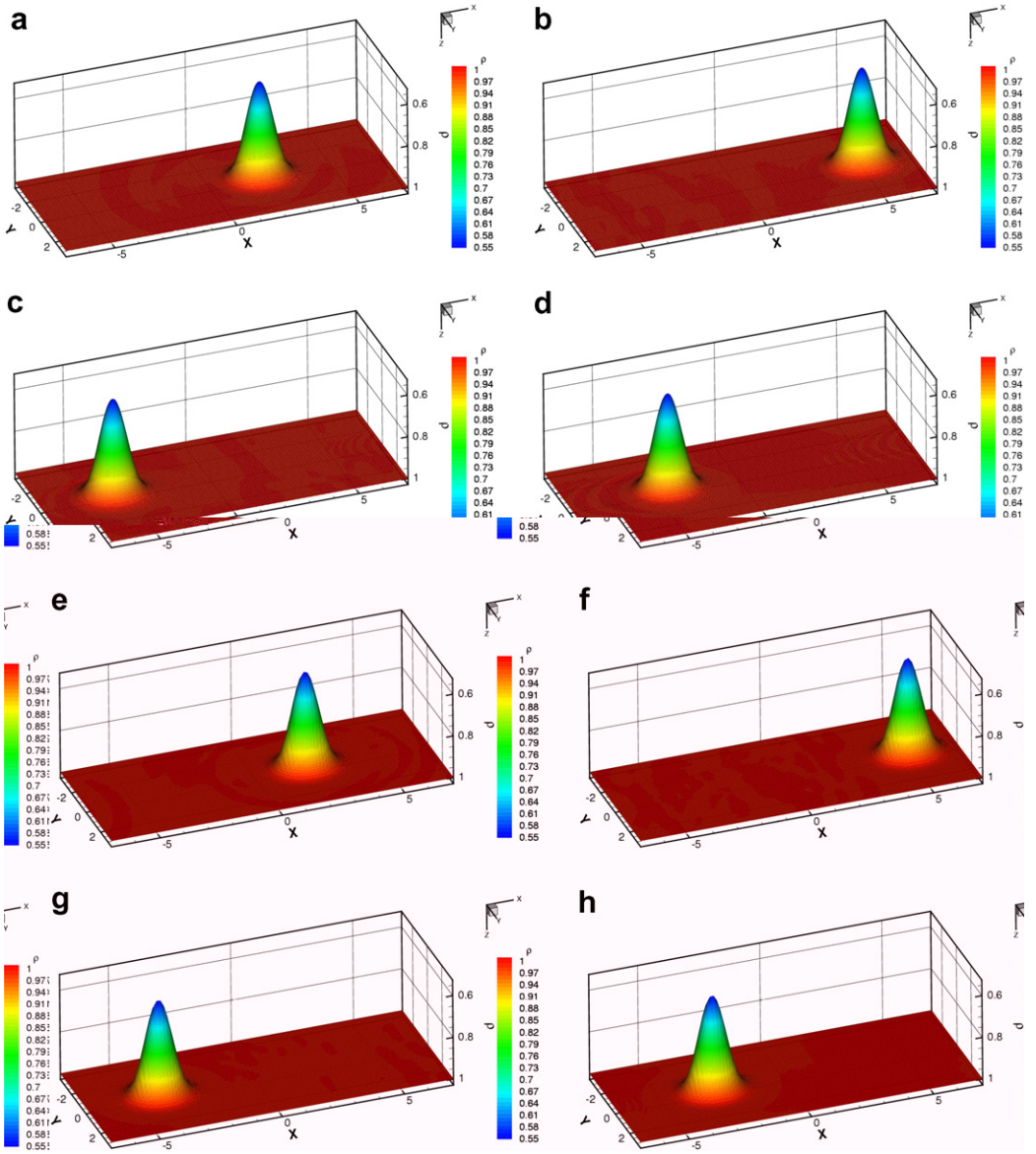

Fig. 4. Density contours (3D) of the CN2 (a–d) and IRK4 (e–h) schemes at various times, $t = 4$ (a,e), 10 (b,f), 20 (c,g) and 50 (d,h), using a time-step of $\Delta t = 0.2$ and $p = 4$ spatial discretization.

step reference solution, 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 the temporal schemes. Additionally, two smaller time-steps, $\Delta t = 0.125$ and 0.0625 are employed for the BDF1, BDF2 and CN2 schemes to extend their range of comparison. A regular mesh with a grid spacing of $\Delta x = \Delta y = 0.25$ and a total of 3136 elements is employed for this study. 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.

The temporal accuracy results for the BDF1, BDF2, CN2 and IRK4 schemes at $t = 4$ are illustrated in Fig. 6, where the computed temporal error is plotted as a function of the time-step on a log–log plot. The first-order backwards differencing scheme displays a slope of 1.0. The second-order backwards differencing

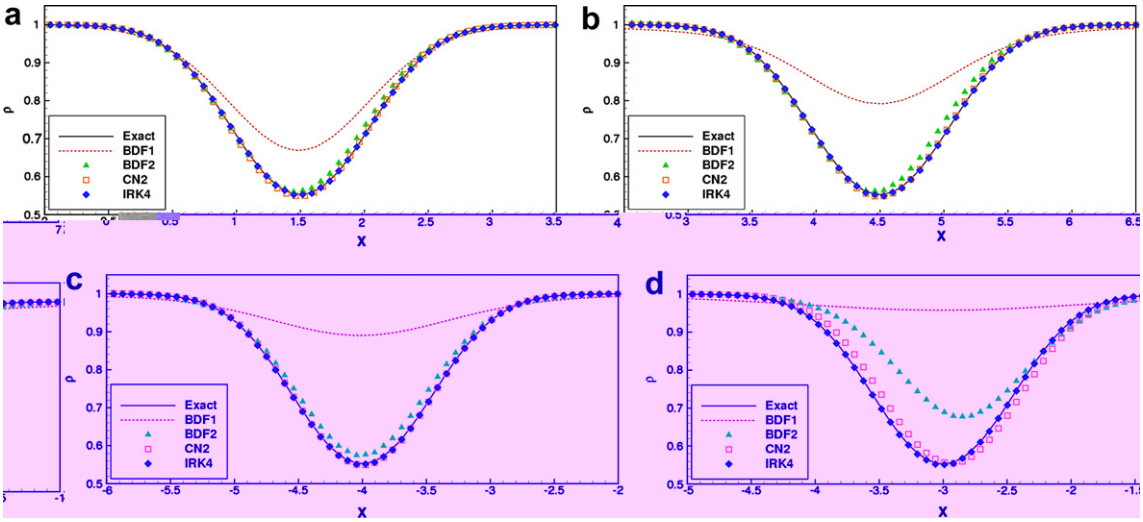

Fig. 5. Comparison of the various temporal schemes with the exact solution, illustrated by density profiles at the centerline $y = 0$, at $t = 4$ (a), 10 (b), 20 (c) and 50 (d), using a time-step of $\Delta t = 0.2$.

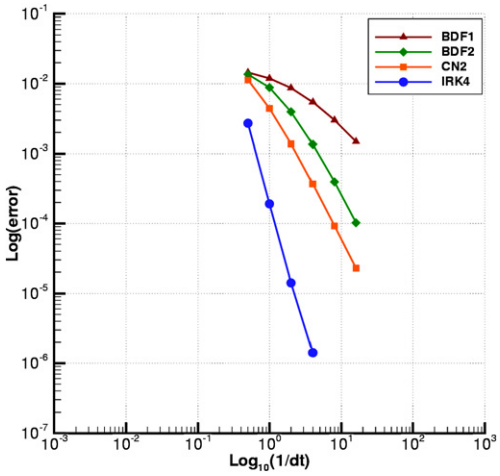

Fig. 6. Comparisons of temporal accuracy for various implicit temporal schemes as a function of time-step size at $t = 4$.

scheme and the Crank-Nicholson scheme demonstrate similar slopes of 1.9 and 2.0, respectively, although the CN2 scheme is consistently more accurate in absolute terms than the BDF2 scheme. The fourth-order Runge–Kutta scheme exhibits a slope of 3.82, which is close to the design value of 4. For any given time-step size, the IRK4 scheme achieves higher accuracy than the BDF1, BDF2 and CN2 schemes, while the BDF2 and CN2 schemes provide better accuracy than BDF1. For example, using a time-step size of $\Delta t = 0.25$, the IRK4 scheme attains a temporal error of approximately 10^{-6} while all the other schemes incur temporal errors larger than 3×10^{-4}.

These results demonstrate that the chosen temporal discretization schemes achieve their design order of accuracy within the range of time steps of interest and for high-order spatial discretizations. On the other hand, since the higher-order temporal schemes (particularly the IRK scheme) are more expensive than the lower-order schemes, the more practical consideration of computational efficiency of these schemes for a prescribed error tolerance must be addressed. However, since the efficiency of these schemes is inherently related to the performance of the implicit solver used at each time step, the performance of the *p*-multigrid solver must first be examined and quantified.

5.1.3. Comparisons of p-multigrid smoothing strategies

As discussed in Section 4.1, several p-multigrid solver variants consisting of the non-linear element Jacobi (NEJ), the quasi-non-linear element Jacobi (qNEJ) and the linearized element Jacobi (LEJ) can be used as smoothers at each implicit time step. In this section, the performance of these three variants in one implicit time step is demonstrated using the IRK4 scheme and the $p = 4$ spatial discretization for the vortex convection test case. Within the multigrid scheme, 5 smoothing passes are utilized on each mesh level. Furthermore, the qNEJ and LEJ smoothers effectively freeze the non-linear Jacobians (and residuals for the LEJ scheme) over five iterations, resulting in a single non-linear update on each grid level within a multigrid cycle.

Fig. 7a depicts the convergence profiles of the LEJ, qNEJ and NEJ solvers in terms of p-multigrid cycles for a single implicit time step solution. Since the solution of five unsteady residuals $(\mathbf{R}_e(\mathbf{u}^{(s)}), s = 2, \ldots, 6)$ is required to be solved in a single implicit time step of the IRK4 scheme, the results shown in this figure are 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 to those obtained in the context of the equivalent steady-state solver [9]: the linearized element Jacobi, quasi-non-linear element Jacobi and non-linear element Jacobi converge with similar rates in terms of the number of p-multigrid cycles. However, when compared in terms of cpu time, as shown in Fig. 7b, the LEJ and qNEJ solvers are seen to be substantially more efficient than the NEJ solver due to the omission of the expensive block diagonal Jacobian recomputation required in the latter scheme at each smoothing pass. In this test case, the LEJ and qNEJ smoother require only a single Jacobian evaluation and LU factorization on each grid level within a multigrid cycle, whereas this procedure must be repeated five times for the NEJ scheme. The reduction of the number of non-linear residual evaluations in the LEJ scheme results in only slightly improved cpu time, while the qNEJ solver has been found to be more robust. In light of these findings, the qNEJ solver is exclusively utilized in the remainder of this work.

5.1.4. Efficiency of p-multigrid strategy

Having settled on the use of the qNEJ smoother, we must now assess the efficiency of the overall p-multigrid solver for time-implicit problems, in order to later examine and understand the efficiency comparisons of the various time-integration schemes. The use of efficient non-linear solvers is particularly important for the performance of high-order temporal schemes, such as IRK, which entail the solution of multiple non-linear problems within a given time step. These schemes are substantially more expensive per time step than their lower-order counterparts, but enable the use of much larger time steps for equivalent accuracy, due to their superior asymptotic properties. Thus, the performance of the multigrid scheme as a function of time step size must be considered in addition to the more traditional performance metric as a function of grid size.

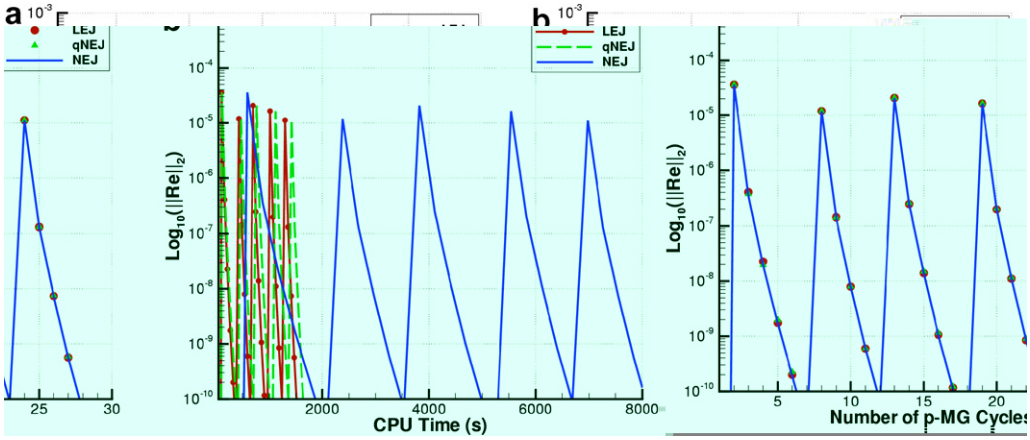

Fig. 7. Comparisons of convergence for solver variants (LEJ, qNEJ and NEJ) in one time step of IRK4 scheme on a mesh of 7056 elements and order $p = 4$, using fixed time-step size of $\Delta t = 0.5$. (a) Convergence history in terms of p-multigrid cycles. (b) Convergence history in terms of cpu time.

In this section, we examine the convergence of the FAS p-multigrid solver (using the qNEJ smoother) compared to that of the corresponding single level solver as a function of grid size and time-step size in one implicit time-integration. Specifically, the grid sizes vary from 3136, to 7056 to 14400 elements on a domain of $-7 \leqslant x \leqslant 7$ and $-3.5 \leqslant y \leqslant 3.5$, and time-step sizes range from $\Delta t = 0.5$, $\Delta t = 1.0$ to $\Delta t = 5.0$. The spatial discretization ($p = 4$), as well as initial and boundary conditions, are kept unchanged and we choose BDF2 as the time-integration scheme, with the understanding that the performance of the multigrid scheme will be similar for implicit systems arising from each BDF or CN2 time step, or each IRK stage.

Fig. 8a depicts the p-multigrid convergence 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 to the mesh size, which illustrates the h-independent property of the p-multigrid solver. On the other hand, the convergence rate of the single level solver decreases progressively 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 (as depicted in Fig. 8b), 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 and computational time, we also discuss the characteristics of the p-multigrid approach and corresponding single level solver with respect to time-step sizes. The obtained convergence profiles 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 problems [9]. 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 $\Delta t = 5$.

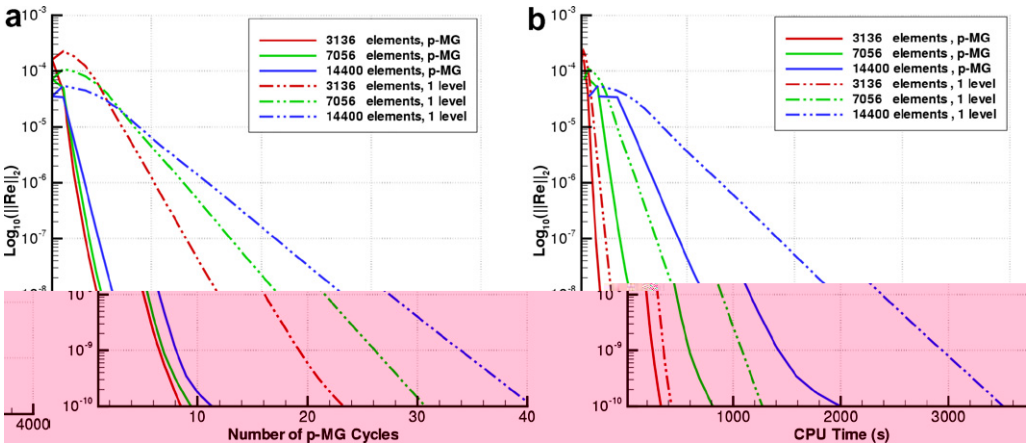

Fig. 8. Comparisons of the convergence between p-MG and single level solvers, using various grid sizes, $N = 3136$, $N = 7056$ and $N = 14400$ with a fixed time-step of $\Delta t = 1.0$ for the BDF2 scheme. (a) Convergence history vs. p-MG cycles. (b) Convergence history vs. cpu time.

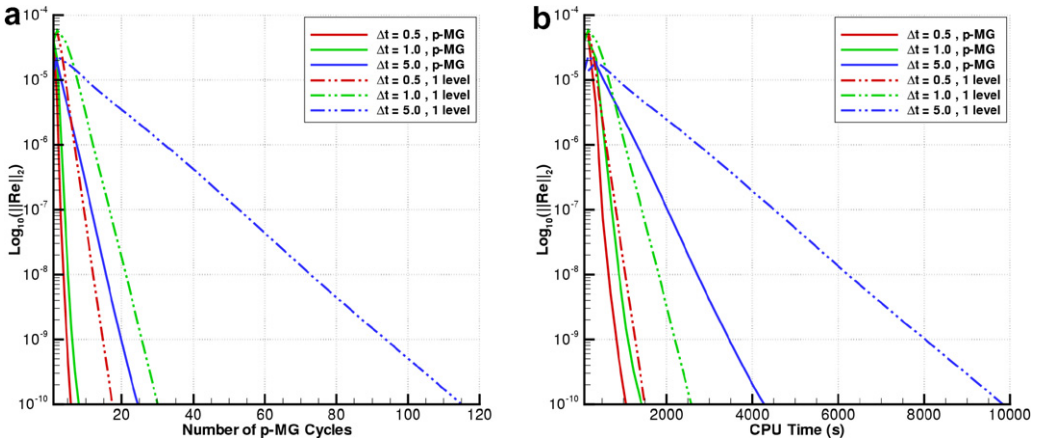

Fig. 9. Comparisons of the convergence between p-MG and single level solvers, using various time-step sizes, $\Delta t = 0.5$, $\Delta t = 1.0$ and $\Delta t = 5.0$, with a fixed mesh size of $N = 14\,400$ for the BDF2 scheme. (a) Convergence history vs. p-MG cycles. (b) Convergence history vs. cpu time.

5.2. Comparison of efficiency of temporal schemes

The results of the time step refinement study discussed in Section 5.1.2 and depicted in Fig. 6 are revisited in Fig. 10, where the temporal error is plotted as a function of the required cpu time, which is obtained using the V-cycle p-MG algorithm (with qNEJ smoother) described in Section 4.1, the performance of which was examined in the previous section. In all cases, the non-linear residuals at each time step are converged to machine precision. Although this may not be necessary for preserving the final accuracy of the computed solutions, the determination of suitable levels for implicit system convergence remains an active research area. In any case, for a given accuracy level, one may expect the required convergence levels to be the same for all schemes, so that the relative comparisons between cpu time required for the various temporal schemes remain valid. The results of Fig. 10 indicate 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 s of cpu time, while the BDF2 scheme requires 8042 s, which is over three times longer. On the other hand, the CN2 scheme in this case requires 3975 s to achieve this accuracy level, which only is 1.52 times longer than the IRK4 scheme and twice as fast as the BDF2 scheme. While the CN2 scheme has the same asymptotic behavior as the BDF2 scheme, it consistently outperforms the BDF2 scheme and comes closest to matching the efficiency of the IRK4 scheme. Nevertheless, the lack of L-stability for the CN2 scheme remains a disadvantage which may result in poor performance of this scheme in other cases. For higher accuracy levels, the advantage of the IRK4 scheme increases, due to the asymptotic properties of these schemes. In general, the choice of the most efficient scheme will depend not only on the efficiency of the non-linear solver, but also on the desired level of accuracy, with higher accuracy levels favoring higher-order schemes.

In another comparison of the efficiency of the various schemes, the delivered accuracy of these schemes at equivalent cpu cost is compared in Fig. 11. This numerical experiment is similar in nature to that displayed in Fig. 5, except that the comparison of the solutions is made at equivalent cpu time as opposed to the use of a constant time step. In order to achieve equivalent cpu time for the various schemes, the time step of each scheme must be adjusted accordingly to achieve a target total run time. The chosen time steps and resulting cpu times for each scheme are given in Table 3, where the use of larger time steps for the more expensive schemes is evident. While the chosen time steps result in approximately similar total cpu costs, the final solutions do not always lie at exactly the same locations in time, due to the use of integer numbers of time steps of varying sizes. Therefore, the final solutions are compared by translating the vortex density profile back to the origin, in order to compare solution accuracy at the approximate points in time given by $t \sim 4$, $t \sim 10$, $t \sim 20$ and $t \sim 50$.

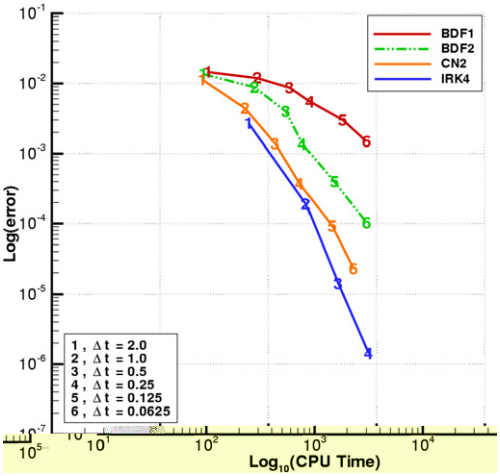

Fig. 10. Comparisons of temporal efficiency for various implicit temporal schemes at $t = 4$ for vortex convection test case.

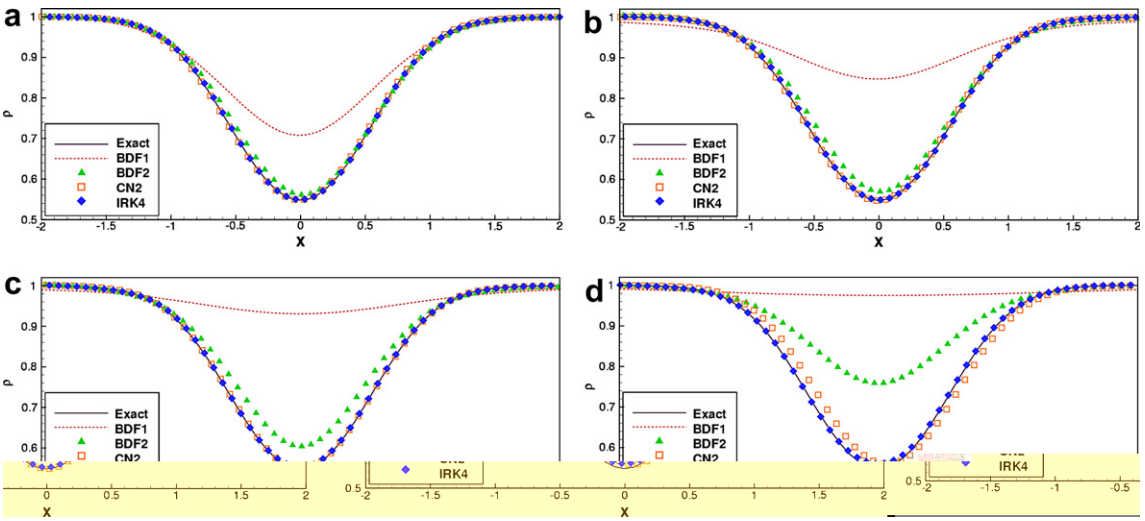

Fig. 11. Comparison of the various temporal schemes with the exact solution for equal work performance, illustrated by density profiles at the centerline $y = 0$, at $t \sim 4$ (a), 10 (b), 20 (c) and 50 (d) using the various required time-step sizes indicated in Table 3.

Table 3
Time-step sizes and cpu costs for BDF1, BDF2, CN2 and IRK4 schemes used for accuracy study at equivalent cpu cost for vortex convection test case

	BDF1	BDF2	CN2	IRK4
Δt	0.31	0.24	0.2	1.1
$t \simeq 4$	2080	1965	2180	1939
$t \simeq 10$	5056	5013	4951	4895
$t \simeq 20$	9810	10122	9895	9818
$t \simeq 50$	23908	25215	24736	24575

The results illustrate what can be surmised from Fig. 10, namely that the fourth-order implicit Runge–Kutta scheme remains the most competitive approach, while the errors in the other schemes accumulate more rapidly as time proceeds, with the CN2 scheme providing the next best accuracy for a given cost. Note that the IRK scheme achieves superior accuracy in this case while using a time step which is approximately five times larger than that used for the lower-order schemes.

5.3. Shedding flow over a triangular wedge

5.3.1. Numerical solutions

The next test case involves the problem of vortex shedding over a triangular wedge. Because of the inviscid nature of the flow simulation, a triangular geometry is chosen in order to ensure that vortices are produced due to separation at sharp corners. In addition to representing a more realistic test case, involving a highly graded unstructured mesh, this case is also used to study the performance of the various temporal schemes in combination with low and higher-order spatial discretizations, thus focusing on the interplay between spatial and temporal errors. The geometry consists of a triangular wedge placed on the centerline $y = 0$ of the computational domain, which contains 10 836 unstructured triangular elements, with the ratio of the smallest to largest cell area being 1:1425 (which corresponds to an explicit CFL ratio of 38:1). The flow is inviscid, and a uniform freestream Mach number of 0.2 is applied as the initial condition. Since our principal focus in these calculations is the ability of these schemes in retaining the shape of the vortices as these are convected downstream from the wedge, and in order to reduce the occurrence of initial discontinuities near the surface of the wedge which would be produced with a uniform freestream initial condition and would be detrimental for high-order spatial discretizations, we first employ a $p = 0$ spatial discretization and BDF2 temporal scheme, utilizing a uniform flow initial condition: $u(\mathbf{x}, t = 0) = u_\infty = 0.5$, $v(\mathbf{x}, t = 0) = v_\infty = 0$, $\rho(\mathbf{x}, t = 0) = \rho_\infty = 1.0$ (as shown in Fig. 12a), to obtain an intermediate solution (shown in Fig. 12b), in which the formed vortices have not yet separated. Then, this intermediate solution is applied as the initial condition to all other high-order $p = 1$ and $p = 3$ spatial discretization schemes, using either the BDF2 or IRK4 time-integration schemes.

Starting with the computed intermediate solution as the initial condition, Figs. 13 and 14 depict the numerical results at $t = 100$ for $p = 1$ and $p = 3$ spatial discretizations using the BDF2 and IRK4 time-integration schemes with the same fixed time-step size of $\Delta t = 0.05$, 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 the best shape-retaining convection capability: the vortices produced around the corners of the triangle wedge keep their shapes far downstream. On the other hand, the $p = 1$ scheme is relatively dissipative as seen by the diffusion of the core of the vortices as they are convected downstream. While the improvement in the solution due to the increase in spatial discretization order is evident, any solution changes due to the use of higher-order temporal schemes are less evident in these qualitative illustrations.

In order to assess the accuracy of our temporal discretizations, we perform a temporal refinement study, similar to that described for the previous test case. The temporal error is isolated by producing a reference "exact" numerical solution, using a small time-step size of $\Delta t = 2 \times 10^{-4}$ for each temporal and spatial discretization. Using the same initial and boundary conditions as described above, computations are performed using time-step sizes from 0.05 to 5×10^{-4} and the error is measured at $t = 0.1$. The results shown in Fig. 15 indicate that the design accuracy of the BDF2 and IRK4 schemes is approached in both cases, yielding curve slopes of 1.92 and 3.96 in the case of the second-order accurate ($p = 1$) spatial discretization, and 1.72 and 3.96 in the case of the fourth-order accurate ($p = 3$) spatial discretization, for the BDF2 and IRK4 temporal schemes, respectively. However, it is notable that the absolute temporal errors of both temporal schemes are consistently smaller for the second-order accurate ($p = 1$) spatial discretization as compared to the higher-order spatial discretization case.

Returning to the results of Figs. 13 and 14, we take a more quantitative look at the incurred errors, by plotting the the density distributions along cuts taken through the centerlines of last three downstream vortices in

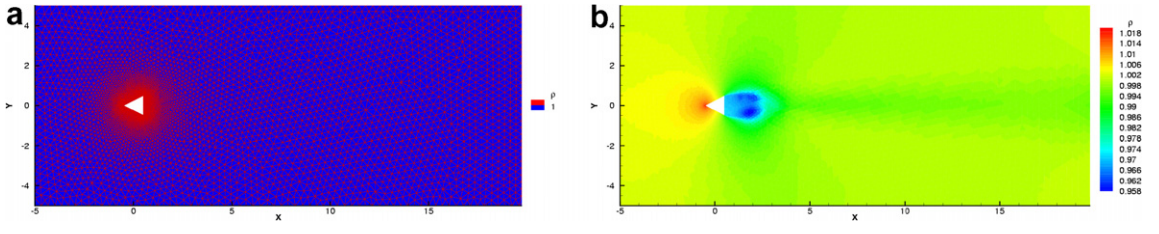

Fig. 12. Mesh and density contours for uniform flow (a) and intermediate solution (b) obtained using $p = 0$ scheme.

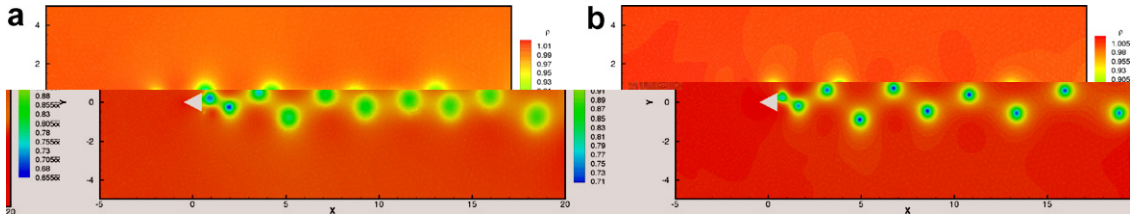

Fig. 13. Density contours of $p = 1$ (a) and $p = 3$ (b) spatial discretization schemes, at $t = 100$, using BDF2 scheme and $\Delta t = 0.05$.

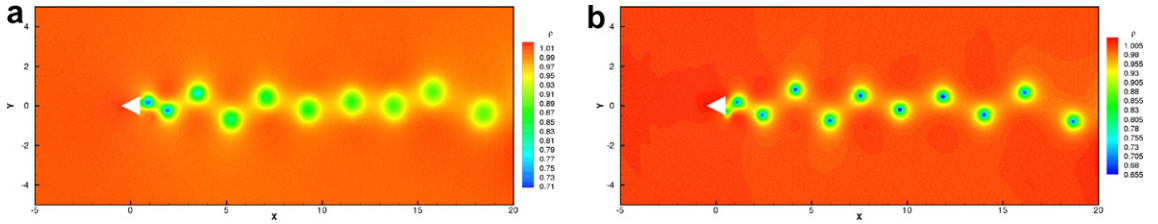

Fig. 14. Density contours of $p = 1$ (a) and $p = 3$ (b) spatial discretization schemes, at $t = 100$, using IRK4 scheme and $\Delta t = 0.05$.

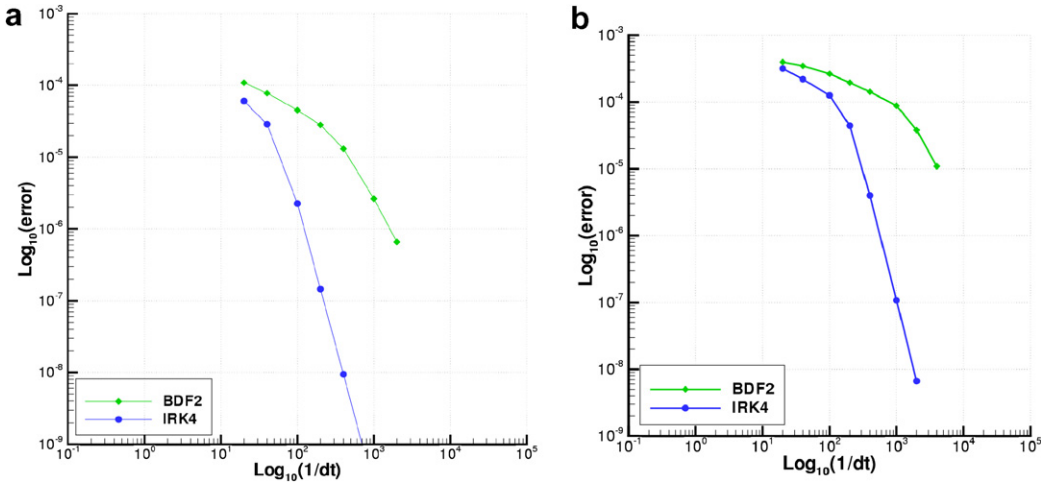

Fig. 15. Comparison of the temporal accuracy as a function of time-step size for the BDF2 and IRK4 schemes at $t = 0.1$ (a) for $p = 1$ spatial discretization, (b) for $p = 3$ spatial discretization.

Fig. 16. For lower-order spatial discretization of $p = 1$ (Fig. 16a), there is no evident improvement in overall solution accuracy using the IRK4 temporal discretization scheme over the BDF2 scheme, since the spatial error dominates the temporal error in this case. On the other hand, for the higher-order spatial discretization of $p = 3$ (Fig. 16b), the IRK4 scheme is seen to be more accurate than the BDF2 scheme, producing notable variations in both diffusion and dispersion rates. Therefore, one may conclude that the use of higher temporal accuracy is not always beneficial, and can even be wasteful in cases where the dominant error is due to other sources. Generally, the most efficient overall solution procedures for a given error tolerance will be strategies which attempt to balance temporal and spatial error sources. This may not be possible in the case of explicit schemes, where the maximum permissible time step is determined by stability considerations rather than accuracy considerations.

We next perform a qualitative comparison between a second-order accurate explicit scheme and the second-order BDF implicit scheme for the vortex shedding problem described above using the fourth-order accurate

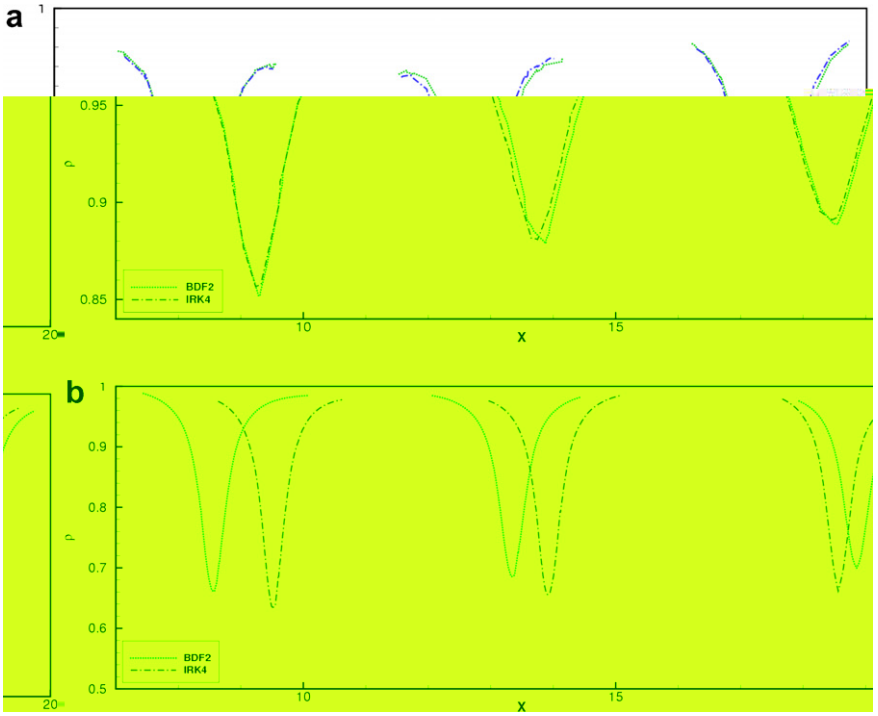

Fig. 16. Density distributions of $p = 1$ and $p = 3$ spatial discretization schemes, at $t = 100$, using BDF2 and IRK4 schemes and $\Delta t = 0.05$. Note the different scales used for the $p = 1$ (a) and $p = 3$ (b) spatial discretizations.

Table 4
Explicit-implicit comparison for the shedding flow case

For solution at $t = 2.5$	Time-step size	Time steps	Convergence limit	Cpu time (s)
Implicit (BDF2)	$\Delta t = 0.05$	50	1×10^{-7}	5160
Explicit (FD2)	$\Delta t = 5 \times 10^{-5}$	50 000	–	22 920

($p = 3$) spatial discretization. Using a second-order accurate forward Euler explicit scheme (FD2), the maximum permissible time step for stability, which is determined by the smallest cells in the mesh, is $\Delta t_{\max} = 5 \times 10^{-5}$. This is three orders of magnitude smaller than the time-step of 0.05 used for the BDF2 scheme. Table 4 compares the characteristics of the explicit and implicit approaches to this problem, using the p-multigrid solver to drive the implicit scheme, and converging each implicit system to a relatively low tolerance of 1×10^{-7}.

While the implicit scheme requires relatively few time steps compared to the explicit scheme, the cpu time required for a single explicit time step is much lower than that required for an implicit time step, since only a residual evaluation is required in the explicit case, as opposed to the solution of a large non-linear system of equations in the implicit case. Nevertheless, the results in Table 4 show that the implicit time-integration scheme requires 4.4 times less cpu time to integrate the solution to the non-dimensional time $t = 2.5$ compared to the explicit approach in this case. However, this comparison is only qualitative in nature, since the temporal accuracy of the two approaches is not matched, and the cpu time required by the implicit approach is strongly dependent on the level of convergence required of the inner iterations for the implicit solver. However, the performance of the explicit scheme is determined by stability considerations, which will be detrimental when the resulting temporal accuracy imposed by the stability limit is not closely related to the desired temporal accuracy.

6. Conclusions

The accurate and efficient solution of time-dependent problems requires a careful balance between both spatial and temporal errors. For problems with disparate length and time scales, implicit time-integration approaches offer the most promising approach for realizing efficient and accurate solutions. In the interest of balancing spatial and temporal errors, this paper has investigated the use of high-order implicit Runge–Kutta schemes in addition to the more traditional second-order accurate backwards differencing (BDF2) and trapeziodal (CN2) time-integration schemes. For high-accuracy requirements, the IRK4 scheme consistently outperforms the lower-order schemes, in spite of the increased cost incurred per time step. The second-order accurate Crank–Nicolson scheme (CN2) performs better than the BDF2 scheme, although the lack of L-stability for this scheme indicates that it should be used with caution.

The key to competitive implicit time-integration schemes lies in the use of an efficient non-linear solver at each time step. In this regards, the *p*-multigrid strategy which has been extended from steady-state problems to time-implicit problems in this work, has been shown to provide an effective mechanism for solving time-implicit systems.

Further work will focus on extending this approach to the unsteady Navier–Stokes equations and including the presence of dynamically deforming meshes.

Acknowledgment

This work was supported by a grant from the Office of Naval Research ONR Grant number N00014-04-1-0602.

Appendix A

The Butcher variables a_{sj} and c_s for the ESDIRK4 scheme are (with $a_{6j} = b_j$)

$$\begin{array}{cccc} a_{21} & a_{22} & a_{31} & a_{32} \\ a_{33} & a_{41} & a_{42} & a_{43} \\ a_{44} & a_{51} & a_{52} & a_{53} \\ a_{54} & a_{55} & a_{61} & a_{62} \\ a_{63} & a_{64} & a_{65} & a_{66} \\ c_1 & c_2 & c_3 & c_4 \\ c_5 & c_6 & & \end{array} \tag{34}$$

$$\begin{array}{cccc} \frac{1}{4} & \frac{1}{4} & \frac{8611}{62500} & \frac{-1743}{31250} \\ \frac{1}{4} & \frac{5012029}{34652500} & \frac{-654441}{2922500} & \frac{174375}{388108} \\ \frac{1}{4} & \frac{15267082809}{155376265600} & \frac{-71443401}{120774400} & \frac{730878875}{902184768} \\ \frac{2285395}{8070912} & \frac{1}{4} & \frac{82889}{524892} & 0 \\ \frac{15625}{83664} & \frac{69875}{102672} & \frac{-2260}{8211} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{83}{250} & \frac{31}{50} \\ \frac{17}{20} & 1 & & \end{array} \tag{35}$$

References

[1] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of 2D Euler equation, J. Comput. Phys. 138 (1997) 251–285.

[2] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method of the numerical solution of the compressible Navier–Stokes equation, J. Comput. Phys. 131 (1997) 267–279.

[3] B. Berde, M. Borrel, Numerical experiments on the accuracy of a discontinuous Galerkin method for the Euler equations, Aerosp. Sci. Technol. 5 (1998) 279–288.
[4] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Appl. Mech. Eng. 35 (1998) 2440–2463.
[5] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, SIAM J. Sci. Comput. 16 (2001) 173–261.
[6] K.J. Fidkowski, T.A. Oliver, J. Lu, D. Darmofal, *p*-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 207 (1) (2005) 92–113.
[7] J.E. Flaherty, L. Krivodonova, J. Remacle, M.S. Shephard, Aspects of discontinuous Galerkin methods for hyperbolic conservation laws, Finite Elements Anal. Design 38 (2002) 889–908.
[8] H. Luo, J.D. Baum, R. Lohner, A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, J. Comput. Phys. 211 (2006) 767–783.
[9] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an *hp*-multigrid approach, J. Comput. Phys. 213 (1) (2006) 330–357.
[10] B. Cockburn, Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws, J. Comput. Phys. 128 (2001) 187–204.
[11] I. Lomtev, C.B. Quillen, G.E. Karniadakis, Spectral/hp methods for viscous compressible flows on unstructured 2d meshes, J. Comput. Phys. 144 (1998) 325–327.
[12] V. Dolejsi, M. Feistauer, A semi-implicit discontinuous Galerkin finite element method for the numerical solution of invisicid compressible flow, J. Comput. Phys. 198 (2004) 727–746.
[13] H. Bijl, M.H. Carpenter, V.N. Vatsa, C.A. Kennedy, Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow, J. Comput. Phys. 179 (2002) 313–329.
[14] P. Rasetarinera, M.Y. Hussaini, An efficient implicit discontinuous spectral Galerkin method, J. Comput. Phys. 172 (2001) 718–738.
[15] H. Bijl, Iterative methods for unsteady flow computations using implicit Runge–Kutta integration schemes, AIAA Paper 2006-1278 (January 2006).
[16] G. Arambatzis, P. Vavilis, I. Toulopoulos, J.A. Ekaterinaris, Implicit high-order time marching schemes for the linearized Euler equations, AIAA Paper 2006-3548 (June 2006).
[17] K.J. Fidkowski, D.L. Darmofal, Development of a higher-order solver for aerodynamic applications, AIAA Paper 2004–0436 (January 2004).
[18] B. Helenbrook, D.J. Mavriplis, Atkins, Analysis of "*p*"-multigrid for continuous and discontinuous finite element discretizations, AIAA Paper 2003-3989 (June 2003).
[19] T.C. Warburton, I. Lomtev, Y. Du, S. Sherwin, G. Karniadakis, Galerkin and discontinuous Galerkin spectral/*hp* methods, Comput. Methods Appl. Mech. Eng. 175 (1999) 343–359.
[20] P. Solin, P. Segeth, I. Zel, High-Order Finite Element Methods, Studies in Advanced Mathematics, Chapman and Hall, 2003.
[21] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (2) (1997) 1553–1570.
[22] E.F. Toro, M. spruce, W. Spears, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1994) 25–34.
[23] D.A. Dunavant, Higher degree efficient symmetrical Gaussian guadrature rules for the triangle, Int. J. Numer. Methods Eng. 21 (1985) 1129–1148.
[24] D.A. Dunavant, Economical symmetrical quadrature rules for complete polynomials over a square domain, Int. J. Numer. Methods Eng. 21 (1985) 1777–1784.
[25] B. Cockburn, C.-W. Shu, The Runge–Kutta dinscontinuous galerkin method for conservation laws v: multidimensional systems, J. Comput. Phys. 141 (1997) 199–224.
[26] E. Hairer, G. Warner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer-Verlag, Berlin, 1991.
[27] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection diffusion reaction equations, Appl. Numer. Math. 44 (1) (2003) 139–181.
[28] G. Jothiprasad, D.J. Mavriplis, D.A. Caughey, Higher order time integration schemes for the unsteady Navier–Stokes equations on unstructured meshes, J. Comput. Phys. 191 (2) (2003) 542–566.
[29] J.C. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Taylor and Francis, 1977.
[30] D.J. Mavriplis, An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers, J. Comput. Phys. 175 (2002) 302–325.
[31] J.W. Ruge, K. Stüben, Algebraic multigrid, in: S.F. McCormick (Ed.), Multigrid Methods, SIAM Frontiers in Applied Mathematics, SIAM, Philadelphia, 1987, pp. 73–131.
[32] F. Davoudzadeh, H. Mcdonald, B.E. Thompson, Accuarcy evaluation of unsteady CFD numerical schemes by vortex preservation, Computers and Fluids 24 (1995) 883–895.
[33] E. Garnier, P. Sagaut, M. Deville, A class of explicit ENO filters with application to unsteady flows, J. Comput. Phys. 170 (2000) 184–204.
[34] H.C. Yee, N.D. Sandham, M.J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters, J. Comput. Phys 150 (1999) 199–238.