

INVESTIGATION OF HIGH-ORDER TEMPORAL SCHEMES FOR THE DISCONTINUOUS GALERKIN SOLUTION OF THE NAVIER-STOKES EQUATIONS

F. BASSI¹, A. COLOMBO¹, C. DE BARTOLO², N. FRANCHINA¹, A. GHIDONI^{3,*} and A. NIGRO²

¹ Department of Industrial Engineering, University of Bergamo, viale Marconi, 5 - 24044 Dalmine (BG), Italy. Email: francesco.bassi@unibg.it, nicoletta.franchina@unibg.it

² Department of Mechanical, Energetic and Management, University of Calabria, via P. Bucci, 44/C - 87036 Rende (CS), Italy. Email: carmine.debartolo@unical.it, alessandra.nigro@unical.it

³ Department of Mechanical and Industrial Engineering, University of Brescia, Via Branze, 38 - 25123 Brescia, Italy. Email: antonio.ghidoni@ing.unibs.it

Key words: Discontinuous Galerkin methods, high-order temporal schemes, Runge-Kutta methods, multistep methods.

Abstract. In this work different high-order temporal schemes, used to advance in time the DG space discretized equations, are investigated: the Explicit Singly Diagonally Implicit Runge Kutta (ESDIRK), the Modified Extended BDF (MEBDF), the Two Implicit Advanced Step-point (TIAS) and a Rosenbrock method. The proposed schemes are evaluated in terms of accuracy and efficiency for two unsteady test-cases: (i) the convection of an inviscid isentropic vortex and (ii) the laminar flow around a cylinder.

1 INTRODUCTION

In recent years Discontinuous Galerkin (DG) methods have received increasing attention in Computational Fluid Dynamics (CFD) [1, 2] due to many attractive features, such as the geometrical flexibility, the use of elements with different solution polynomial approximation in the same grid and the compactness of the scheme. The DG space discretized equations can be advanced in time using different time integration schemes. Explicit Runge-Kutta methods are very popular for the solution of unsteady problems because they can match in time the high accuracy of the DG discretization. However these schemes can be very inefficient due to the time-step restriction. Implicit time integration schemes can be adopted to overcome this limitation, such as the multistep Backward Differentiation Formulae (BDF) schemes. However, BDF are only *A*-stable up to the second-order and their low accuracy is not well suited to match the spatial accuracy of DG methods. In this work different high-order temporal schemes were investigated in

terms of accuracy and efficiency. In particular the following schemes were considered: the Explicit Singly Diagonally Implicit Runge Kutta (ESDIRK) [3], the Modified Extended BDF (MEBDF) [4], the Two Implicit Advanced Step-point (TIAS) [5] and a Rosenbrock method [12]. These schemes have been evaluated for two unsteady test-cases: (i) the convection of an inviscid isentropic vortex and (ii) the laminar flow around a cylinder.

1.1 DG SPACE DISCRETIZATION

The Navier-Stokes (NS) equations can be written in compact form as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) + \nabla \cdot \mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \quad (1)$$

where \mathbf{u} is the vectors of the m conservative variables, and $\mathbf{F}_c, \mathbf{F}_v \in \mathbb{R}^m \otimes \mathbb{R}^d$ are defined as the arrays of the inviscid and viscous flux vectors. A weak formulation of the NS equations is obtained multiplying each scalar law in Eq. (1) by an arbitrary smooth test function $v_j \in \mathbf{v}$, $1 \leq j \leq m$, and integrating by parts, that is

$$\int_{\Omega} v_j \frac{\partial u_j}{\partial t} \, d\mathbf{x} - \int_{\Omega} \nabla v_j \cdot \mathbf{F}_j(\mathbf{u}, \nabla \mathbf{u}) \, d\mathbf{x} + \int_{\partial\Omega} v_j \mathbf{F}_j(\mathbf{u}, \nabla \mathbf{u}) \cdot \mathbf{n} \, d\sigma = 0, \quad (2)$$

where \mathbf{F}_j is the sum of the inviscid and viscous flux vectors of the j -th equation.

Let Ω_h be an approximation of the domain $\Omega \in \mathbb{R}^d$, $\mathcal{T}_h = \{K\}$ a mesh of Ω_h , i.e. a collection of “finite elements” K , $\mathcal{F}_h = \{F\}$ the mesh faces, and let \mathbf{V}_h denotes a discontinuous finite element space spanned by polynomial functions continuous only inside each element K , i.e.

$$\mathbf{V}_h \stackrel{\text{def}}{=} [\mathbb{P}_d^l(\mathcal{T}_h)]^m, \quad (3)$$

where

$$\mathbb{P}_d^l \stackrel{\text{def}}{=} \{v_h \in L^2(\Omega_h) : v_h|_K \in \mathbb{P}_d^l, \forall K \in \mathcal{T}_h\} \quad (4)$$

is the space of polynomials of degree at most l on the element K . Hierarchical and orthogonal shape functions are adopted, and are obtained using a modified Gram-Schmidt procedure, assuming as a starting point a set of monomial functions [9].

In Eq. (2) the solution \mathbf{u} and the test function \mathbf{v} are replaced with a finite element approximation \mathbf{u}_h and a discrete test function \mathbf{v}_h respectively, where \mathbf{u}_h and \mathbf{v}_h belong to the space $\mathbf{V}_h \stackrel{\text{def}}{=} [\mathbb{P}_d^l(\mathcal{T}_h)]^m$. The discontinuous approximation of the numerical solution requires to introduce a specific treatment of the inviscid and viscous interface fluxes. In order to ensure conservation and correctly account for the waves propagation the former is based on the Godunov flux computed with an exact Riemann solver. For the latter the BR2 scheme, proposed in [6] and theoretically analyzed in [8, 7], is employed.

Accounting for these aspects, the DG formulation of the compressible NS equations consists in seeking $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \int_K v_{h,j} \frac{\partial u_{h,j}}{\partial t} \, d\mathbf{x} - \sum_{K \in \mathcal{T}_h} \int_K \nabla_h v_{h,j} \cdot \mathbf{F}_j(\mathbf{u}_h, \nabla_h \mathbf{u}_h + \mathbf{r}(\llbracket \mathbf{u}_h \rrbracket)) \, d\mathbf{x} \\ + \sum_{F \in \mathcal{F}_h} \int_F \llbracket v_{h,j} \rrbracket \cdot \hat{\mathbf{F}}_j(\mathbf{u}_h^\pm, (\nabla_h \mathbf{u}_h + \eta_F \mathbf{r}_F(\llbracket \mathbf{u}_h \rrbracket))^\pm) \, d\sigma = 0 \end{aligned} \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \quad (5)$$

where $\llbracket \cdot \rrbracket$ is the jump operator, \mathbf{r} and \mathbf{r}_F the global and local lifting operators, and η_F is a penalty parameter prescribed accordingly to [7].

2 TIME INTEGRATION

The DG space discretization of Eq. (5) results in the following system of (nonlinear) ODEs in time:

$$\mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}, \quad (6)$$

where \mathbf{U} is the global vector of unknown degrees of freedom, \mathbf{M} is a global block diagonal matrix and $\mathbf{R}(\mathbf{U})$ is the vector of residuals. The matrix \mathbf{M} reduces to the identity matrix due to the use of orthonormal basis functions.

In this work the system (6) is advanced in time in an implicit sense by using the following schemes: the Explicit Singly Diagonally Implicit Runge Kutta (ESDIRK), the Modified Extended BDF (MEBDF), the Two Implicit Advanced Step-point (TIAS) and a Rosenbrock method.

2.1 MEBDF schemes

MEBDF schemes [4] are A -stable up to fourth order, and a k -step MEBDF scheme is of order $k + 1$. Standard MEBDF schemes have some limitations; they are not self starting from the third order and the time step must be constant during the simulation.

The k -step MEBDF involves three stages to advance the solution to the next time step. The first two stages are k order BDF formulas, while the last stage combines the two previous BDF evaluations, resulting in a $k + 1$ order solution. The three stages are as follows:

1. Stage 1. Compute the first predictor $\bar{\mathbf{U}}^{n+k}$ as the solution of the k -step BDF:

$$\frac{\mathbf{M}}{\Delta t} \left(\bar{\mathbf{U}}^{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j \mathbf{U}^{n+j} \right) + \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}^{n+k}) = \mathbf{0}.$$

2. Stage 2. Compute the second predictor $\bar{\mathbf{U}}^{n+k+1}$ as the solution of the k -step BDF:

$$\frac{\mathbf{M}}{\Delta t} \left(\bar{\mathbf{U}}^{n+k+1} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}^{n+k} + \sum_{j=0}^{k-2} \hat{\alpha}_j \mathbf{U}^{n+j+1} \right) + \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}^{n+k+1}) = 0.$$

3. Stage 3. Compute the final solution \mathbf{U}^{n+1} as

$$\frac{\mathbf{M}}{\Delta t} \left(\mathbf{U}^{n+k} + \sum_{j=0}^{k-1} \bar{\alpha}_j \mathbf{U}^{n+j} \right) + \hat{\beta}_k \mathbf{R}(\mathbf{U}^{n+k}) + (\bar{\beta}_k - \hat{\beta}_k) \mathbf{R}(\bar{\mathbf{U}}^{n+k}) + \bar{\beta}_{k+1} \mathbf{R}(\bar{\mathbf{U}}^{n+k+1}) = 0.$$

$\hat{\alpha}_j$ and $\hat{\beta}_k$ are the coefficients of the BDF, while $\bar{\alpha}_j$, $\bar{\beta}_k$ and $\bar{\beta}_{k+1}$ are the coefficients of the MEBDF.

The three non-linear systems are solved by means of an inexact Newton approach. In particular the Jacobian matrix is computed at the stage 1 of the first time step and then it is recomputed only if the convergence rate of the Newton method between iteration $j+1$ and j is above a given tolerance or after 10 time steps since the last evaluation. In particular the Jacobian matrix is recomputed if the following relation is not satisfied:

$$\frac{\|\Delta \mathbf{U}_{j+1}\|_2}{\|\Delta \mathbf{U}_j\|_2} > tol_J \quad \text{or} \quad n_{\Delta t} > 10, \quad (7)$$

where tol_J is a tolerance, which is set to 0.25, $n_{\Delta t}$ is the number of time steps and $\|\Delta \mathbf{U}_j\|_2$ the L^2 norm of the solution variation at the j^{th} Newton iteration.

The linear system arising at each Newton step is solved using the restarted GMRES algorithm preconditioned with the block Jacobi method as available in the PETSc library [10].

2.2 TIAS schemes

TIAS schemes [5] are A -stable up to the sixth order, and a k -step TIAS scheme is of order $k+1$. Standard TIAS schemes have same limitations as MEBDF.

The k -step TIAS scheme involves four stages to advance the solution to the next time step. The first three stages are k order BDF formulas, while the last stage combines the three previous BDF evaluations, resulting in a $k+1$ order solution. The four stages are as follows:

- Stage 1. Compute the first predictor $\bar{\mathbf{U}}^{n+k}$ as the solution of the k -step BDF::

$$\mathbf{M} \left(\bar{\mathbf{U}}^{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j \mathbf{U}^{n+j} \right) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}^{n+k}) = 0.$$

- Stage 2. Compute the second predictor $\bar{\mathbf{U}}^{n+k+1}$ as the solution of the k -step BDF::

$$\mathbf{M} \left(\bar{\mathbf{U}}^{n+k+1} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}^{n+k} + \sum_{j=0}^{k-2} \hat{\alpha}_j \mathbf{U}^{n+j+1} \right) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}^{n+k+1}) = 0.$$

- Stage 3. Compute the third predictor $\bar{\mathbf{U}}^{n+k+2}$ as the solution of the k -step BDF::

$$\mathbf{M} \left(\bar{\mathbf{U}}^{n+k+2} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}^{n+k+1} + \hat{\alpha}_{k-2} \bar{\mathbf{U}}^{n+k} + \sum_{j=0}^{k-3} \hat{\alpha}_j \mathbf{U}_{n+j+2} \right) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}^{n+k+2}) = 0.$$

- Stage 4. Compute the final solution \mathbf{U}^{n+1} as:

$$\mathbf{M} \left(\mathbf{U}^{n+k} + \sum_{j=0}^{k-1} \tilde{\alpha}_j \mathbf{U}^{n+j} \right) + \Delta t \left[\tilde{\beta}_{k+2} \mathbf{R}(\bar{\mathbf{U}}^{n+k+2}) + \tilde{\beta}_{k+1} \mathbf{R}(\bar{\mathbf{U}}^{n+k+1}) + \beta_k \mathbf{R}(\bar{\mathbf{U}}^{n+k}) + (\tilde{\beta}_k - \beta_k) \mathbf{R}(\mathbf{U}^{n+k}) \right] = 0.$$

$\hat{\alpha}_j$ and $\hat{\beta}_k$ are the BDF coefficients, while $\tilde{\alpha}_j$, $\tilde{\beta}_{k+2}$, $\tilde{\beta}_{k+1}$, $\tilde{\beta}_k$ and β_k are the TIAS coefficients (Tab. 1). In particular, $\tilde{\beta}_{k+2}$ and β_k are free coefficients which determine the stability properties of the scheme, and all the other coefficients are expressed in terms of $\tilde{\beta}_{k+2}$. The non linear system arising at each stage is solved by using the same approach described for the MEBDF.

2.3 ESDIRK schemes

This class of Runge-Kutta schemes [3] can be constructed to be A- and L-stable for any temporal order of accuracy. A s -stage ESDIRK scheme applied to the system (6) can be written as

$$\begin{aligned} \mathbf{U}^0 &= \mathbf{U}^n \\ \mathbf{U}^s &= \mathbf{U}^n - \Delta t \sum_{j=1}^i a_{ij} \mathbf{M}^{-1} \mathbf{R}(\mathbf{U}^j), \quad i = 1, \dots, s \\ \mathbf{U}^{n+1} &= \mathbf{U}_h^s \end{aligned}$$

where a_{ij} are the Butcher coefficients of the scheme. The first stage is explicit due to $a_{11} = 0$ and the last stage defines the solution for the subsequent step. The s non linear problems arising at each time step are solved by using the same strategy proposed for the MEBDF schemes.

k	$\tilde{\alpha}_0$	$\tilde{\alpha}_1$	$\tilde{\alpha}_2$	$\tilde{\alpha}_3$	$\tilde{\alpha}_4$
1	-1				
2	$\frac{5}{23} - \frac{12}{23} \cdot \tilde{\beta}_{k+2}$	$-\frac{28}{23} + \frac{12}{23} \cdot \tilde{\beta}_{k+2}$			
3	$-\frac{17}{197} + \frac{165}{197} \cdot \tilde{\beta}_{k+2}$	$\frac{99}{197} - \frac{648}{197} \cdot \tilde{\beta}_{k+2}$	$-\frac{279}{197} + \frac{483}{197} \cdot \tilde{\beta}_{k+2}$		
4	$\frac{111}{2501} - \frac{8018}{7503} \cdot \tilde{\beta}_{k+2}$	$-\frac{728}{2501} + \frac{14427}{2501} \cdot \tilde{\beta}_{k+2}$	$\frac{2124}{2501} - \frac{29106}{2501} \cdot \tilde{\beta}_{k+2}$	$-\frac{4008}{2501} + \frac{52055}{7503} \cdot \tilde{\beta}_{k+2}$	
5	$-\frac{394}{14919} + \frac{74711}{59676} \cdot \tilde{\beta}_{k+2}$	$\frac{2925}{14919} - \frac{123215}{14919} \cdot \tilde{\beta}_{k+2}$	$-\frac{3200}{4973} + \frac{11762}{4973} \cdot \tilde{\beta}_{k+2}$	$\frac{18700}{14919} - \frac{459473}{14919} \cdot \tilde{\beta}_{k+2}$	$-\frac{26550}{14919} + \frac{914897}{59676} \cdot \tilde{\beta}_{k+2}$

k	$\tilde{\beta}_k$	$\tilde{\beta}_{k+1}$
1	$\frac{3}{2} + \tilde{\beta}_{k+2}$	$-\frac{1}{2} - 2 \cdot \tilde{\beta}_{k+2}$
2	$\frac{22}{23} + \frac{53}{23} \cdot \tilde{\beta}_{k+2}$	$\frac{4}{23} - \frac{64}{23} \cdot \tilde{\beta}_{k+2}$
3	$\frac{150}{197} + \frac{827}{197} \cdot \tilde{\beta}_{k+2}$	$-\frac{18}{197} - \frac{706}{197} \cdot \tilde{\beta}_{k+2}$
4	$\frac{1644}{2501} + \frac{16767}{2501} \cdot \tilde{\beta}_{k+2}$	$-\frac{144}{2501} - \frac{10998}{2501} \cdot \tilde{\beta}_{k+2}$
5	$\frac{2940}{4973} + \frac{48933}{4973} \cdot \tilde{\beta}_{k+2}$	$-\frac{200}{4973} - \frac{25961}{4973} \cdot \tilde{\beta}_{k+2}$

Table 1: Coefficients of the k -steps TIAS corrector stage.

2.4 Linearly implicit Rosenbrock-type Runge–Kutta schemes

Unlike the methods reported above, the linearly implicit (Rosenbrock-type) Runge–Kutta schemes involve the solution of a sequence of linear instead of non-linear systems. In this paper we consider the RODASP, fourth-order six stages, scheme of Steinbach [12]. Besides the systems of Ordinary Differential Equations (ODEs) here considered, these schemes can also accurately be applied to systems of (non-linear) Differential Algebraic Equations (DAEs) that can arise from the discretization of incompressible flows, see [13]. This class of methods, applied to the system 6, can be compactly written as

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \sum_{j=1}^s b_j \mathbf{K}_j, \quad (8)$$

$$\left(\frac{\mathbf{M}}{\Delta t} + \gamma \mathbf{J} \right) \mathbf{K}_i = -\mathbf{R} \left(\mathbf{U}^n + \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{K}_j \right) - \mathbf{J} \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_j \quad i = 1, \dots, s, \quad (9)$$

where b_i , α_{ij} , γ_{ij} are real coefficients and \mathbf{J} is the Jacobian matrix of the DG space discretization that is computed analytically to fully account for the dependence of the residuals on the unknown vector and its derivatives including the implicit treatment of the lifting operator and of boundary conditions. A direct implementation of Eq. 9 requires at each stage the matrix-vector multiplication $\mathbf{J} \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_j$. In practice this can be avoided by resorting to the following equivalent formulation

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \sum_{j=1}^s m_j \mathbf{Y}_j, \quad (10)$$

$$\left(\frac{\mathbf{M}}{\gamma \Delta t} + \mathbf{J} \right) \mathbf{Y}_i = -\mathbf{R} \left(\mathbf{U}^n + \sum_{j=1}^{i-1} a_{ij} \mathbf{Y}_j \right) - \frac{\mathbf{M}}{\gamma \Delta t} \sum_{j=1}^{i-1} c_{ij} \mathbf{Y}_j \quad i = 1, \dots, s, \quad (11)$$

where for $i = 1, \dots, s$

$$\mathbf{K}_i = \frac{1}{\gamma} \mathbf{Y}_i - \sum_{j=1}^{i-1} c_{ij} \mathbf{Y}_j, \quad (12)$$

and m_j , a_{ij} and c_{ij} are the transformed coefficients of the scheme, see [14].

Each stage of Eq. 11 requires to solve a linear system of the form $\mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{0}$. However, since the matrix \mathbf{A} is the same at each stage, the Jacobian matrix needs to be evaluated only once. The linear system can be solved using the GMRES (Generalized Minimal RESidual) algorithm, preconditioned with the block Jacobi method.

3 RESULTS

In this section the performance and potential of the temporal schemes described in the previous section are investigated for two test-cases: (i) the convection of an inviscid isentropic vortex and (ii) the laminar flow around a cylinder. The notations BDF $_x$, MEBDF $_x$, TIAS $_x$, ESDIRK $_x$ and RODASP $_x$ indicate the x -th order schemes. The computing time is reported in the figures as a normalized value with respect to the TauBenchmark [11] value, $t_{\text{TauBenchmark}}$, obtained on a full node of the cluster used for the CFD simulations¹. The normalized computing time is measured as work units (WU) and is defined as $\text{WU} = (t_{\text{CPU}} * n_{\text{cores}}) / t_{\text{TauBenchmark}}$, where t_{CPU} is the wall clock time and n_{cores} the numbers of cores used for the simulation.

3.1 Convection of an isentropic vortex

The free stream values of the vortex problem were the flow density ρ_∞ , the velocity vector components u_∞ and v_∞ , the pressure p_∞ and the temperature T_∞ , which were set to $(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 1, 1, 1, 1)$. At $t_0 = 0$ the free stream flow was perturbed by an isentropic vortex $(\delta u, \delta v, \delta T)$ centered at (x_0, y_0) and defined as:

$$\delta u = -\frac{\alpha}{2\pi} (y - y_0) e^{\phi(1-r^2)}, \quad (13)$$

$$\delta v = \frac{\alpha}{2\pi} (x - x_0) e^{\phi(1-r^2)}, \quad (14)$$

$$\delta T = -\frac{\alpha^2 (\gamma - 1)}{16\phi\gamma\pi^2} e^{2\phi(1-r^2)}, \quad (15)$$

¹-n 250000 -s 10 define the reference TauBench workload for the hardware benchmark

where $\gamma = 1.4$ is the ratio of specific heats, $\phi = \frac{1}{2}$ and $\alpha = 5$ are parameters which determine the strength of the vortex, $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the distance of a generic point (x, y) from the vortex center (x_0, y_0) , which is set to $(5, 5)$ in a periodic domain $[0, 10] \times [0, 10]$. The computations were performed on a uniform cartesian grid of 50×50 quadrilaterals with a simulation time equal to the vortex revolution period T and a \mathbb{P}^6 solution approximation. In Fig. 1 (left) the initial flowfield is depicted. Figure 1 (right) compares the exact and the computed pressure profiles along the square diagonal using BDF2, MEBDF4, TIAS4 and TIAS6 for the timestep $T/40$. Notice that only using sixth order TIAS scheme the minimum value of the pressure is not smoothed out, while the solution obtained with BDF2 is completely wrong,

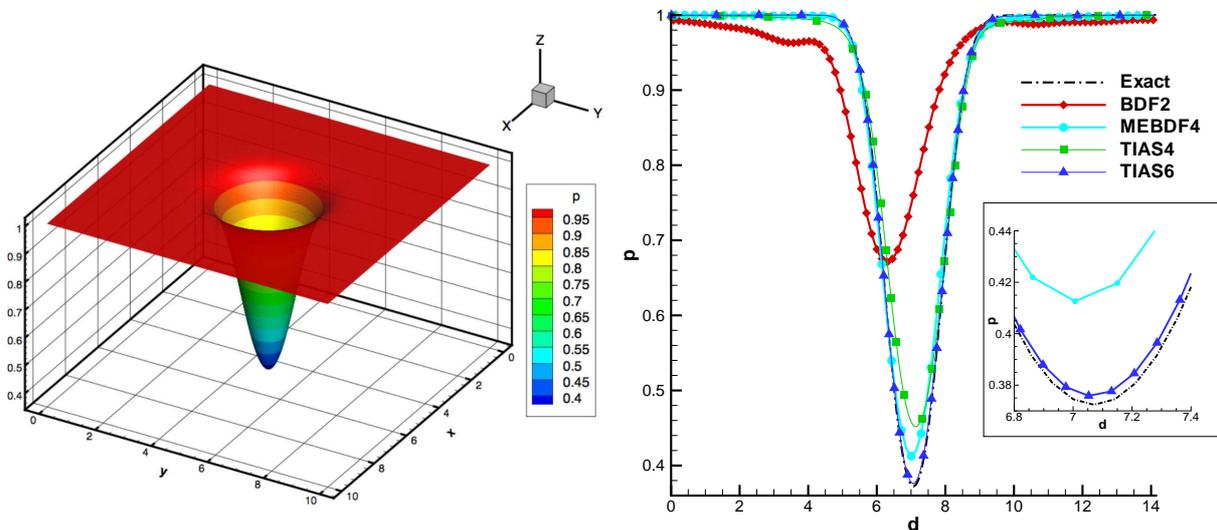


Figure 1: Vortex: pressure initial flow field (left) and comparison between computed and exact pressure profiles along the domain diagonal for BDF2, MEBDF4, TIAS4 and TIAS6, \mathbb{P}^6 solution

The analysis of the design-order convergence was performed for progressively smaller time steps, by keeping a high-order solution approximation (\mathbb{P}^6) to minimize the spatial discretization error. In Fig. 2 (left) the L^2 norm of the pressure error as a function of the time step is depicted, showing that all schemes verify their design-order convergence.

All temporal schemes and BDF2 were compared to assess their computational efficiency and to understand when high-order temporal schemes could be advantageous. Figure 2 (right) shows the L^2 norm of the pressure error as a function of the work units. Some high-order temporal schemes (MEBDF4, TIAS6, RODASP4) can always achieve greater accuracy than BDF2 with a comparable CPU time. RODASP4 is the more efficient scheme for large time steps, while for small time steps TIAS6 performs better. RODASP4 guarantees also the lowest error level.

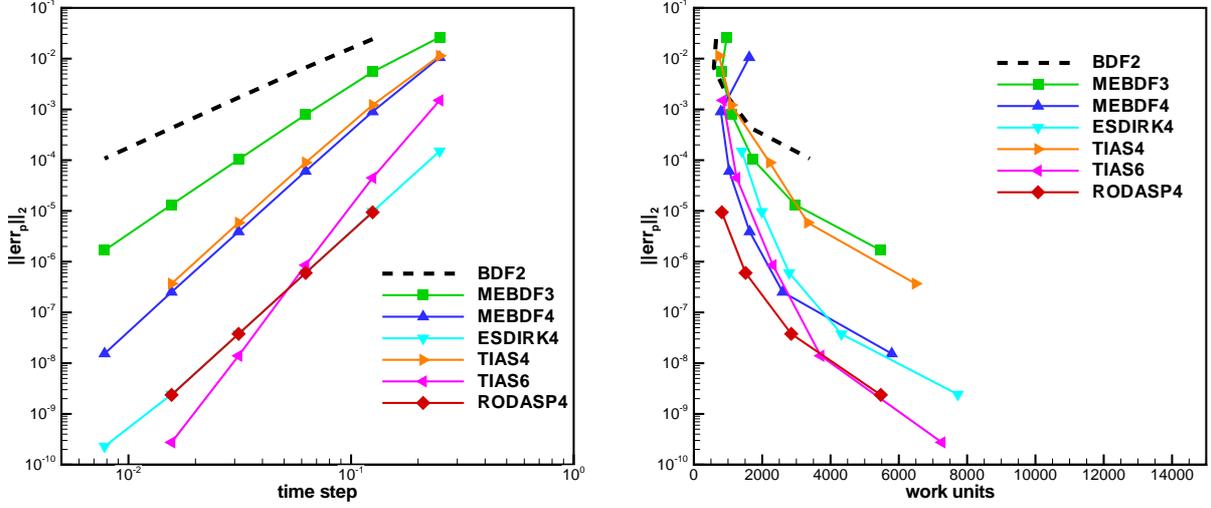


Figure 2: Vortex: L^2 norm of the pressure error as a function of the time step (left) and of the normalized CPU time (right), \mathbb{P}^6 solution

3.2 Unsteady vortex shedding behind a circular cylinder

The second test-case was the laminar flow around a two-dimensional circular cylinder computed for a Mach number $M_\infty = 0.2$ and a Reynolds number $Re = 100$. The computational grid was a quadratic mesh containing 3690 quadrilateral. Figure 3 shows the potential of the TIAS6 scheme in comparison with the BDF2. The density and velocity contours are depicted after 30 vortex shedding periods, highlighting the greater accuracy provided by the TIAS6 scheme.

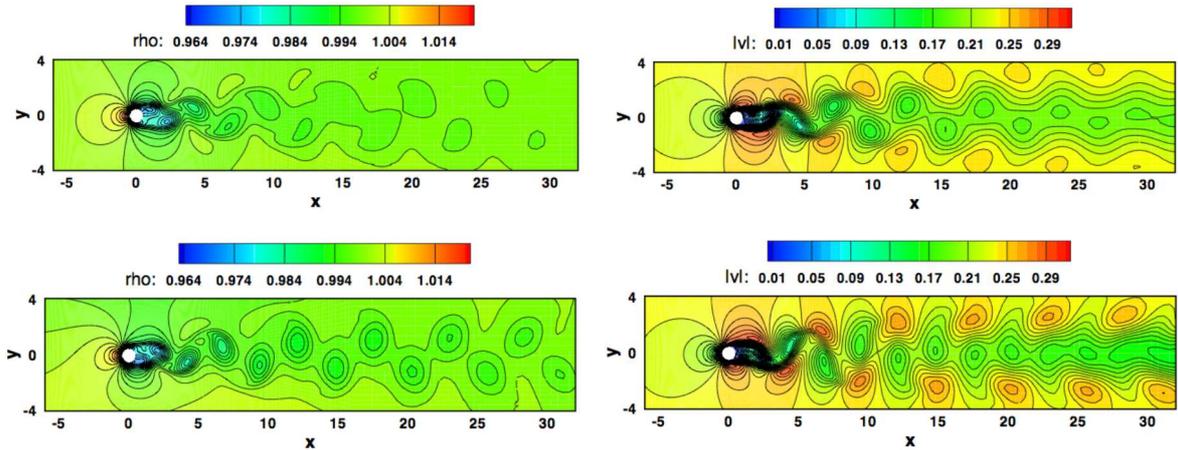


Figure 3: Cylinder: density (left column) and velocity magnitude (right column) fields after 30 vortex shedding periods. Top row: BDF2. Bottom row: TIAS6. \mathbb{P}^5 solution

The analysis of the design-order convergence was performed for progressively smaller

time steps and for a simulation time equal to $1.5T$, where T was the vortex shedding period. A high-order solution approximation, *i.e.* \mathbb{P}^5 , was used to minimize the spatial discretization error. For this test-case an exact solution was not available and, hence, the TIAS6 scheme was used to compute a reference solution with a very small time step $\Delta t = T/5120$. Fig. 4 (left) shows the lift coefficient error (the error is defined with respect to the reference C_L computed with TIAS6) as a function of the time step. All schemes verified their design-order convergence.

All temporal schemes and BDF2 were compared to assess their computational efficiency and to understand when high-order temporal schemes could be advantageous. Figure 4 (right) shows that also for this testcase the high-order temporal schemes can achieve greater accuracy than BDF2 with a comparable CPU time. However, for large time steps and high error levels (10^{-2}) BDF2 scheme seems to be more efficient. RODASP4 and TIAS6 have the same behaviour shown in the previous test case. RODASP4 performs slightly better for large time steps, while TIAS6 is better for small time steps.

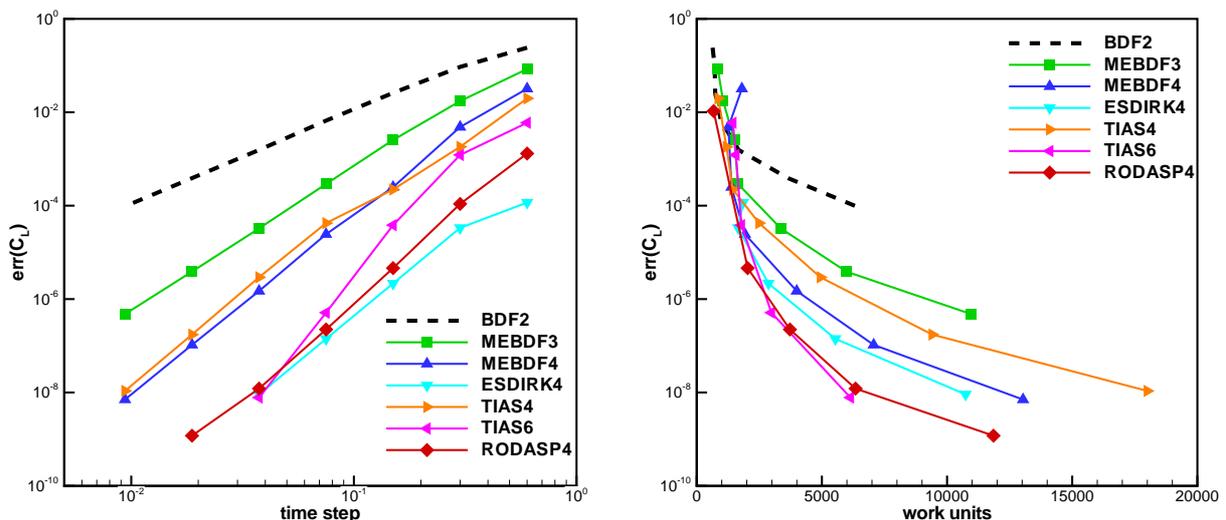


Figure 4: Cylinder: C_L error as a function of the time step (left) and of the normalized CPU time (right), \mathbb{P}^5 solution

4 CONCLUSIONS

In this work different high-order temporal schemes, used to advance in time the DG space discretized equations, have been investigated. The accuracy and the efficiency of these methods have been compared by computing two test cases: the convection of an inviscid isentropic vortex and the unsteady laminar flow around a circular cylinder.

The design-order convergence has been verified for all schemes on both testcases, showing the greater accuracy reached by these schemes in comparison to the standard BDF2 for the same time step. The computational efficiency of the schemes has been also evalu-

ated and compared with BDF2. In general BDF2 scheme seems to be more efficient only for large time steps and high error levels (10^{-2}). Best schemes in terms of efficiency are the RODASP4 and TIAS6.

Acknowledgements

This work has been carried out within the EU FP7 IDIHOM project [15].

REFERENCES

- [1] Bassi F, Botti L, Colombo A, Crivellini A, Franchina N, Ghidoni A, Rebay S. Very high-order accurate discontinuous Galerkin computation of transonic turbulent flows on aeronautical configurations. *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 113, Kroll N, Bieler H, Deconinck H, Couaillier V, van der Ven H, Sørensen K (eds.). Springer Berlin / Heidelberg, 2010; 25–38.
- [2] Ghidoni A, Colombo A, Rebay S and Bassi F. Simulation of the transitional flow in a low pressure gas turbine cascade with a high-order discontinuous Galerkin method. *Journal of Fluids Engineering, Transactions of the ASME* 2013; **135**(7).
- [3] Carpenter MH, Kennedy CA, Bijl H, Viken SA, Vatsa VN. Fourth-order Runge—Kutta schemes for fluid mechanics applications. *J. Sci. Comput.* October 2005; **25**(1):157–194.
- [4] Cash JR. The integration of stiff initial value problems in ODEs using Modified Extended Backward Differentiation Formulae. *Computers & Mathematics with Applications* 1983; **5**(9):645–657.
- [5] GY Psihoyios, JR Cash. A stability result for general linear methods with characteristic function having real poles only. *BIT Numerical Mathematics* 1998; **38**(3):612–617.
- [6] Bassi, F., and Rebay, S., 1997. “A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations”. *J. Comput. Phys.*, **131**, pp. 267–279.
- [7] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D., 2002. “Unified analysis of discontinuous Galerkin methods for elliptic problems”. *SIAM J. Numer. Anal.*, **39**(5), pp. 1749–1779.
- [8] Brezzi, F., Manzini, G., Marini, D., Pietra, P., and Russo, A., 2000. “Discontinuous Galerkin approximations for elliptic problems”. *Numer. Meth. for Part. Diff. Eq.*, **16**, pp. 365–378.
- [9] F. Bassi, L. Botti, A. Colombo, D. Di Pietro, and P. Tesini, 2012. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *J. Comput. Phys.*, **231**, pp. 45–65.

- [10] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, 2001. PETSc Web page. [Http://www.mcs.anl.gov/petsc](http://www.mcs.anl.gov/petsc).
- [11] Integrated Performance Analysis of Computer Systems – Benchmarks for distributed computer systems 2005. <http://www.ipacs-benchmark.org>.
- [12] F. Steinebach, 1995. Order reduction of ROW methods for DAEs and method of lines applications. Techn. Hochsch., Fachbereich Mathematik.
- [13] A. Crivellini, V. DAlessandro and F. Bassi, 2013. Assessment of a high-order discontinuous Galerkin method for incompressible three-dimensional NavierStokes equations: Benchmark results for the flow past a sphere up to $Re = 500$. *J. Comput. Phys.*, **86**, pp. 442–458.
- [14] E. Hairer and G. Wanner, 2002. Solving ordinary differential equations II, Stiff and differential-algebraic problems. Springer Verlag.
- [15] IDIHOM, Industrialisation of High-Order Methods – a top-down approach, Specific Targeted Research Project supported by European Commision, URL: http://www.dlr.de/as/en/desktopdefault.aspx/tabid-7027/11654_read-27492/