ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

MATRIX-FREE MODIFIED EXTENDED BACKWARD DIFFERENTIATION FORMULAE APPLIED TO THE DISCONTINUOUS GALERKIN SOLUTION OF COMPRESSIBLE UNSTEADY VISCOUS FLOWS

A. Nigro[1], C. De Bartolo[1], A. Crivellini[2], F. Bassi[3]

[1] University of Calabria-Department of Mechanical, Energy and Management Engineering-DIMEG
Ponte P. Bucci cubo 44/C, 87036 - Rende (CS), Italy
alessandra.nigro@unical.it, carmine.debartolo@unical.it

[2] Polytechnic University of Marche-Department of Industrial Engineering and Mathematical Sciences
Via Brecce Bianche, 60100 - Ancona (AN), Italy
a.crivellini@univpm.it

[3] University of Bergamo-Department of Engineering
Viale Marconi 5, 24044 - Dalmine (BG), Italy
francesco.bassi@unibg.it

Keywords: Modified Extended Backward Differentiation Formulae, Discontinuous Galerkin discretization, Matrix-free approximation, Frozen preconditioner.

Abstract. *In this work a matrix-free modified extended backward differentiation time integration method has been implemented in a high-order discontinuous Galerkin solver for the unsteady Navier-Stokes equations. The resulting non-linear systems at each time step are solved iteratively using a preconditioned inexact Newton/Krylov method. In order to speed-up the solution process a frozen preconditioner formulation and a polynomial extrapolation technique for computing a better initial guess for the Newton iterations have been considered. Numerical results for compressible inviscid and viscous test cases show the effectiveness of the proposed numerical strategies and the performance advantages of the matrix-free method compared to its matrix-explicit counterpart for this class of implicit multi-stage time schemes. Furthermore, the influence of some physical (low Mach) and space discretization (stretched grid) aspects is examined to highlight pros and cons of the proposed time integration algorithm and its potential in solving non-stiff and stiff systems with respect to the widely used explicit Runge-Kutta schemes.*

1 INTRODUCTION

Because of the availability of very fast and massively parallel computers, there is a current trend to use advanced flow models to predict the unsteady turbulent structures directly. In order of complexity from most to least complex we distinguish Direct Numerical Simulation (DNS) (no turbulence modeling at all), the Large Eddy Simulation (LES) that is based on a SubGrid Scale (SGS) model for the non-resolved turbulent scales, the Implicit Large Eddy Simulation (ILES), where the spatial discretization itself acts like a SGS model, and hybrid RANS-LES (RANS near the solid body, LES elsewhere). All of these approaches require the coupling of high space accuracy with high temporal accuracy to convect all resolved turbulent scales at the right speed with minimal dissipation and dispersion. In this respect, the use of very accurate time integration schemes in the context of high-order methods gives the possibility of efficient simulations with high accuracy. Among the high-order methods, the Discontinuous Galerkin (DG) finite element method turns out to be an ideal candidate for dealing with such complex flow computations do to its high precision together with a natural amenability to parallelization (for an overview of the DG method see [1]).

Explicit high-order Runge-Kutta schemes combined with the DG space discretization (RK-DG methods) [2, 3, 4], are commonly used to address the numerical solution of unsteady flows. Explicit methods are well suited for problems with similar spatial and temporal scales, but become inefficient for unsteady flows of low reduced frequency, as well as for stiff problems. Due to their better stability properties, implicit methods are potentially more effective than explicit methods in this respect. Several high-order semi-implicit multi-stage (Rosenbrock) and implicit multi-step multi-stage (MEBDF, TIAS) time integration schemes for DG space approximations have been recently proposed for both inviscid and laminar flows [5, 6, 7, 8]. Although these schemes allow to overcome the strong restriction to the size of the time step typical of explicit methods, their use require to compute, store and factorize large Jacobian matrices. These tasks require vast computational resources both in terms of CPU time and memory, especially for high-order spatial discretizations and large scale problems.

This problem can be addressed using the matrix-free approach. Different matrix-free methods for the numerical solution of partial differential equations have been develop in the past by Gear and Saad [9], Chan and Jackson [10] and Brown and Hindmarsh [11]. Further, Johan [12] and McHugh and Knoll [13, 14] have used these methods to solve compressible and incompressible flows. More recently, Rasetarinera and Hussani [15] and K. Hillewaert et al. [16] applied the matrix-free method in the context of the DG solution of the Euler equations. A similar approach has been adopted by Crivellini and Bassi [17] for the DG discretizion of the Navier-Stokes and RANS equations. In the above cited DG papers the authors developed implicit solvers for steady compressible flows. In this work we adopt the matrix-free approach for unsteady computations solving the DG discretized compressible Navier-Stokes equations with the high-order accurate MEBDF method. These implicit time integration formulae belong to the family of predictor-corrector methods. MEBDF schemes involve three non-linear stages: the first two are predictor stages that use a standard k-step BDF scheme, the last one is a corrector stage that uses an advanced implicit k-step formula of order k + 1. MEBDF are A-stable for k = 1, .., 3 and stiffly stable for k = 4, .., 8.

The matrix-free approach seems well suited to be coupled with this class of non-linear multi-stage time schemes [18]. In fact, the additional cost with respect to a matrix-explicit approach to approximate the matrix-vector product, at each non-linear iteration, is equal to one residual evaluation, which is much cheaper with respect to the evaluation of the Jacobian matrix required

by the matrix-explicit approach for this update. This computational advantage is expected to compensate the further residual evaluations for each linear iteration, required by the matrix-free approach and not necessary for the matrix-explicit one, as the MEBDF schemes are composed by three non-linear stages. Furthermore, at large time steps as well as for stiff problems, the update of the implicit matrix-vector product is determinant for the convergence of the non-linear Newton iterations at each stage.

The system matrix of high-order accurate DG discretizations is usually ill-conditioned, hence preconditioning is essential to accelerate the convergence of the linear solver. Here, according to [17], the preconditioner consists of an incomplete lower-upper (ILU0) factorization of the analytically computed system matrix. The matrix-free approach allows to approximate the preconditioner without affecting the quadratic convergence of the Newton loops. Furthermore, the MEBDF stages represent non-linear systems with the same system matrix. Both these features are central to the performance of the unsteady solver, avoiding the update of the preconditioning matrix within each time step by applying the same preconditioner to the linear systems of the predictor-corrector stages. Note that the efficiency of the algorithm will be further enhanced when the preconditioner can be frozen for succeeding time steps, e.g. non-stiff ODE systems, small time steps. Another important computational aspect is related to the choice of the initial guess for the Newton-type solver at each non-linear stage of the MEBDF scheme. In particular, for the second Newton loop a polynomial extrapolation technique to generate a more accurate Newton initialization is adopted using a set of previous solutions.

The proposed test cases for inviscid and viscous flows are the vortex transport by uniform flow and a laminar manufactured solution, both with an analytical solution. The shown results aim at demonstrating the capability of the proposed matrix-free approach (MF-MEBDF), in combination with the numerical strategies proposed, to improve the performance of the implicit solver with respect to its explicit counterpart (ME-MEBDF), highlighting its pros and cons and its performance in solving stiff and non-stiff systems of equations with respect to the widely used explicit Runge-Kutta schemes.

In the following of the paper the governing equations are presented in Section 2. Sections 3 and 4 are devoted to space (DG) and time (MEBDF) discretizations, respectively. Numerical results are presented and discussed in Section 5. Conclusions are finally reported in Section 6.

2 GOVERNING EQUATIONS

The compressible Navier-Stokes equations in conservative form are

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}_c\left(\mathbf{q}\right) = \nabla \cdot \mathbf{F}_v\left(\mathbf{q}, \nabla \mathbf{q}\right), \tag{1}$$

where $\mathbf{q}$ is the vector of conservative variables, $\mathbf{q} = \left[\rho, \rho u, \rho v, \rho E\right]^T$, and $\mathbf{F}_c\left(\mathbf{f}_c, \mathbf{g}_c\right)$ and $\mathbf{F}_v\left(\mathbf{f}_v, \mathbf{g}_v\right)$ are the inviscid and viscous flux vectors respectively, given by

$$\mathbf{f}_c = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho H u \end{pmatrix}, \qquad \mathbf{g}_c = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho H v \end{pmatrix}; \tag{2}$$

$$\mathbf{f}_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{xx} u + \tau_{yx} v - q_x \end{pmatrix}, \qquad \mathbf{g}_v = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{xy} u + \tau_{yy} v - q_y \end{pmatrix}. \tag{3}$$

In these equations ρ is the fluid density, u and v are the x and y velocity components, respectively, p is the pressure and E is the total internal energy for unit mass. The total enthalpy for unit mass H is given by $H = E + p/\rho$. The shear stress tensor components τ_{ij} and the heat flux vector components q_i of the viscous flux vectors can be calculated as

$$\tau_{xx} = \left(2\mu\frac{\partial u}{\partial x} + \lambda\nabla\cdot\mathbf{v}\right), \qquad \tau_{yy} = \left(2\mu\frac{\partial v}{\partial y} + \lambda\nabla\cdot\mathbf{v}\right), \tag{4}$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \tag{5}$$

$$q_x = -\kappa\frac{\partial T}{\partial x}, \qquad q_y = -\kappa\frac{\partial T}{\partial y}. \tag{6}$$

In order to close the system of equations, the Navier-Stokes equations must be augmented by algebraic expressions that relate the internal energy E, the pressure p, the dynamic viscosity μ, the second viscosity coefficient λ and the conductivity coefficient κ to the thermodynamic state of the fluid. For an ideal gas, assuming that the fluid satisfies the equation of state of perfect gas, the pressure is given by $p = \rho\left(\gamma - 1\right)\left[E - \left(u^2 + v^2\right)/2\right]$, where γ is the ratio of specific heats of the fluid, given by $\gamma = C_p/C_v$. The dynamic viscosity coefficient μ can be approximated using the power-law viscosity formula

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/4}, \tag{7}$$

where T_0, μ_0 are temperature and viscosity at the reference conditions, respectively.

3 DG DISCRETIZATION

To discretize in space Eq. (1), we consider an approximation Ω_h of the domain Ω consisting of a set of non-overlapping elements $\tau_h = \{K\}$, denoting by $\partial\Omega_h$ the boundary of the discrete approximation and by Γ_h^0 the set of internal edges. We consider piecewise polynomial functions on τ_h with no global continuity requirement. Denoting with $P_n\left(K\right)$ the space of polynomial functions of degree at most n in the element K, the solution and test functions space is defined by

$$\mathbf{V}_h = \{\mathbf{v}_h \in \left(L^2\left(\Omega_h\right)\right)^{N+2} : \mathbf{v}_h \in \left(P_n\left(K\right)\right)^{N+2} \forall K \in \tau_h\}, \tag{8}$$

where N is the number of spatial dimensions. To build a set of hierarchical and orthogonal shape functions for the space (8) the Modified Gram-Schmidt (MGS) procedure [19, 20, 21] is used, considering the starting set of monomial functions of the same degree n. By using the BR2 scheme presented in [22, 23] and theoretically analyzed in [24, 25], we obtain the DG formulation of Eq. (1), given by

$$\begin{aligned}
&\int_{\Omega_h}\mathbf{v}_h\cdot\frac{\partial\mathbf{q}_h}{\partial t}dx - \int_{\Omega_h}\nabla\mathbf{v}_h : \left(\mathbf{F}_c\left(\mathbf{q}_h\right) - \mathbf{F}_v\left(\mathbf{q}_h, \nabla\mathbf{q}_h + \mathcal{R}\left(\left[\!\left[\mathbf{q}_h\right]\!\right]_0\right)\right)\right)dx \\
&+ \int_{\Gamma_h^0}\left(\mathbf{v}_h^- - \mathbf{v}_h^+\right)\cdot\mathbf{H}\left(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}^-\right)d\sigma - \int_{\Gamma_h^0}\left[\!\left[\mathbf{v}_h\right]\!\right] : \{\mathbf{F}_v\left(\mathbf{q}_h, \nabla\mathbf{q}_h + \eta_e\mathcal{R}_e\left(\left[\!\left[\mathbf{q}_h\right]\!\right]_0\right)\right)\}d\sigma \\
&+ \int_{\partial\Omega_h}\left(\mathbf{v}_h\otimes\mathbf{n}\right) : \left(\mathbf{H}\left(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n}\right) - \mathbf{F}_v\left(\mathbf{q}_h, \nabla\mathbf{q}_h + \eta_e\mathcal{R}_e\left(\left[\!\left[\mathbf{q}_h\right]\!\right]_0\right)\right)_b\right)d\sigma = 0.
\end{aligned} \tag{9}$$

In the above equation η_e is called "penalty" parameter and its lower bound is established as the number of neighbours elements of the generic element K to guarantee the stability of the method. $\mathcal{R}_e\left([\![\mathbf{q}_h]\!]_0\right)$ and $\mathcal{R}\left([\![\mathbf{q}_h]\!]_0\right)$ are the local and global lifting operators, respectively, accounting in the gradient of the diffusive fluxes for the jumps in $\mathbf{q}_h$ occurring at the element interfaces, defined as

$$[\![\mathbf{q}_h]\!]_0 = \begin{cases} [\![\mathbf{q}_h]\!] & \text{on } \Gamma_h^0 \\ \mathbf{q}_h \otimes \mathbf{n} & \text{on } \partial\Omega_h \end{cases}, \tag{10}$$

$$[\![\mathbf{q}_h]\!] = \mathbf{q}_h^+ \otimes \mathbf{n}^+ + \mathbf{q}_h^- \otimes \mathbf{n}^-. \tag{11}$$

The trace operator $\{(\cdot)\}$ denotes the average between left $(\cdot)^-$ and right state $(\cdot)^+$, see Fig.(1). Finally, $\mathbf{H}\left(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}\right)$ and $\mathbf{H}\left(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n}\right)$ are the numerical flux functions at the interior and boundary faces, respectively, for which any of the numerical flux functions commonly considered in the finite volume method can be used.

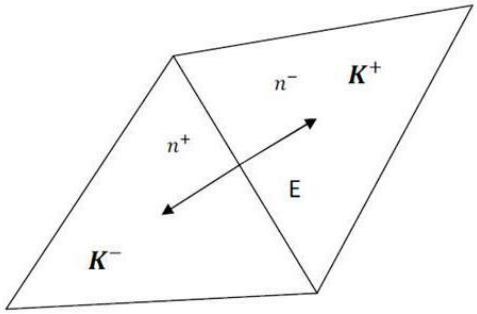

Figure 1: Two elements K^+ and K^- sharing edge E.

A detailed analysis of the Discontinuous Galerkin method, including mathematical foundations, practical implementation aspects and examples that demonstrate the advantages of this approach, can be found in the collection of articles edited by Cockburn et al. [1] and in the book written by Di Pietro et al. [21].

4 TIME INTEGRATION

Numerical integration of the DG space discretization, reported in Eq. (9), results in the following system of Ordinary Differential Equations (ODEs) in time

$$\mathbf{M}\frac{d\mathbf{Q}}{dt} + \mathbf{R}\left(\mathbf{Q}\right) = 0, \tag{12}$$

where $\mathbf{M}$ is the global block mass matrix that, for the conservative variables, reduces to the identity matrix by using orthonormal shape functions (MGS algorithm), $\mathbf{Q}$ is the global vector of unknown degrees of freedom and $\mathbf{R}$ is the vector of residuals. For the sake of simplicity in what follows we have omitted the mass matrix, being $\mathbf{M} = \mathbf{I}$.

4.1 MEBDF method

In this work the system of ODEs of Eq. (12) is advanced in time implicitly by using the MEBDF method [7, 26, 27, 28], which has been originally proposed for the numerical solution

of stiff initial value problems in an attempt to derive a class of multi-step integration formulae characterized by better stability properties and higher-order accuracy than the standard Backward Differentiation Formulae (BDF). The result is a class of k-step formulae of order $k+1$ that are A-stable for $k = 1, .., 3$ and stiffly stable for $k = 4, .., 8$. The general k-step MEBDF algorithm consists of successively solving the following three stages to advance the solution in time:

- Stage 1. Compute the first predictor $\bar{\mathbf{Q}}_{n+1}$ of order k with a k-step BDF:

$$\hat{\alpha}_1 \bar{\mathbf{Q}}_{n+1} + \sum_{j=2}^{k+1} \hat{\alpha}_j \mathbf{Q}_{n+2-j} + \Delta t \hat{\beta}_k \mathbf{R}\left(\bar{\mathbf{Q}}_{n+1}\right) = 0; \tag{13}$$

- Stage 2. Compute the second predictor $\bar{\mathbf{Q}}_{n+2}$ of order k with a k-step BDF:

$$\hat{\alpha}_1 \bar{\mathbf{Q}}_{n+2} + \hat{\alpha}_2 \bar{\mathbf{Q}}_{n+1} + \sum_{j=3}^{k+1} \hat{\alpha}_j \mathbf{Q}_{n+3-j} + \Delta t \hat{\beta}_k \mathbf{R}\left(\bar{\mathbf{Q}}_{n+2}\right) = 0; \tag{14}$$

- Stage 3. Compute the corrected solution $\mathbf{Q}_{n+1}$ of order $k+1$ as:

$$\bar{\alpha}_1 \mathbf{Q}_{n+1} + \sum_{j=2}^{k+1} \bar{\alpha}_j \mathbf{Q}_{n+2-j} + \Delta t \left[\hat{\beta}_k \mathbf{R}\left(\mathbf{Q}_{n+1}\right) + \left(\bar{\beta}_k - \hat{\beta}_k\right) \bar{\mathbf{R}}_{n+1} + \bar{\beta}_{k+1} \bar{\mathbf{R}}_{n+2}\right] = 0, \tag{15}$$

  where

$$\bar{\mathbf{R}}_{n+1} = \mathbf{R}\left(\bar{\mathbf{Q}}_{n+1}\right) \quad \text{and} \quad \bar{\mathbf{R}}_{n+2} = \mathbf{R}\left(\bar{\mathbf{Q}}_{n+2}\right). \tag{16}$$

$\hat{\alpha}_j$ and $\hat{\beta}_j$ are BDF coefficients, while $\bar{\alpha}_j$ and $\bar{\beta}_j$ are MEBDF coefficients. The values of both these coefficients are reported in [26]. Note that, the better accuracy and stability properties of MEBDF with respect to BDF are obtained extending this latter by adding two further stages and a future point at t_{n+2}. Therefore, the first two stages of the method are standard k-step BDF schemes, while the last stage uses an advanced implicit k-step formula of order $k+1$.

4.2 Implementation issues

All the non-linear stages are solved with a Newton-type method. By reformulating Eqs. (13–15 in terms of a non-linear residual function $\mathcal{F}$ and a constant vector $\mathbf{b}$, a generic stage can be rewritten in a more compact form as

$$\mathcal{F}\left(\mathbf{Q}\right) = \mathbf{b}, \tag{17}$$

with

$$\mathcal{F}\left(\mathbf{Q}\right) = \mathbf{Q} + \beta \Delta t \mathbf{R}\left(\mathbf{Q}\right), \tag{18}$$

$$\mathbf{b} = -\sum_j \alpha_j \mathbf{Q}_j \qquad \text{(Predictor stages)}, \tag{19}$$

$$\mathbf{b} = -\sum_j \alpha_j \mathbf{Q}_j - \Delta t \sum_i \beta_i \mathbf{R}_i \qquad \text{(Corrector stage)}, \tag{20}$$

where $\mathbf{Q}_j$ is a previous known solution calculated at t_j time and $\mathbf{R}_i$ the residual evaluation at the end of each predictor stage. Notice that the overbars on the quantities computed at each predictor stage have been omitted, and that β, α_j and β_i are generic stage coefficients.

After applying the Newton scheme to Eq. (17) the following system of equations is obtained:

$$\mathbf{W}^l \Delta \mathbf{Q}^l = -\mathcal{F}\left(\mathbf{Q}^l\right) + \mathbf{b}, \tag{21}$$

with

$$\Delta \mathbf{Q}^l = \mathbf{Q}^{l+1} - \mathbf{Q}^l, \tag{22}$$

$$\mathbf{W}^l = \frac{\partial \mathcal{F}}{\partial \mathbf{Q}}\left(\mathbf{Q}^l\right) = \mathbf{I} - \Delta t \beta \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\left(\mathbf{Q}^l\right), \tag{23}$$

where $\mathbf{W}^l$ is the system matrix.

To successfully and efficiently solve the linear system of equations given by (21), the preconditioned GMRES algorithm available in PETSc [29] is used. Applying left-preconditioning we solved

$$\left[\mathbf{P}^l\right]^{-1} \mathbf{W}^l \Delta \mathbf{Q} = -\left[\mathbf{P}^l\right]^{-1} \left[\mathcal{F}\left(\mathbf{Q}^l\right) + \mathbf{b}\right], \tag{24}$$

where the preconditioning matrix $\mathbf{P}$ is chosen to be the effective incomplete lower-upper factorization ILU(0) of the analitically computed system matrix $\mathbf{W}$.

In order to ensure an accurate and efficient iterative solution of the Newton loop at each stage, the non-linear initialization procedure implemented in the algorithm is as follows:

- the solution of the first stage at the current time step is initialized with the solution of the second stage calculated at the previous time step;
- the solution of the second stage is initialized with a polynomial extrapolation using a set of previous solutions;
- the solution of the third stage is initialized with the solution of the first stage.

4.2.1 Matrix-Free formulation

Referring to Eq. (21), we note that the system matrix does not need to be computed and stored explicitly, but only its action on the vector $\Delta \mathbf{Q}^l$ is required. Therefore, the product of $\mathbf{W}^l$ times the vector $\Delta \mathbf{Q}^l$ has been replaced by the first-order Taylor series expansion

$$\mathbf{W}^l \Delta \mathbf{Q}^l \simeq \frac{\mathcal{F}\left(\mathbf{Q}^l + \varepsilon \Delta \mathbf{Q}^l\right) - \mathcal{F}\left(\mathbf{Q}^l\right)}{\varepsilon}, \tag{25}$$

where the ϵ value is computed as proposed by Pernice and Walker [30].

This method requires two residual evaluations per outer Newton iteration and one residual function evaluation per inner GMRES iteration. Since the residual evaluation is more expensive than a matrix-vector product, the Matrix-Free (MF) GMRES is less performing than its explicit counterpart. However, MEBDF consists of three stages and for each non-linear system it may require a system matrix construction to ensure the convergence of the Newton loop when the Matrix-Explicit (ME) approach is used. In fact, not updating the implicit matrix at the beginning of each stage or within the Newton loop reduces the convergence rate of the non-linear

iterations, especially in the case of large time steps, stiff ODEs systems and high-resolution discretizations. Since only one additional residual evaluation is required by the MF approach with respect to its explicit counterpart, which is much more cheaper than the implicit matrix evaluation, it is expected that, in the above cited cases, for this class of multi-stage schemes MF becomes competitive in cost to the ME approach.

Although the iterative matrix is not built, a preconditioning matrix $\mathbf{P}$ is still needed for the convergence of GMRES. Thus, the proposed MF solver adopts the storage of the analitically computed system matrix for preconditioning purpose. The MF approach allows to approximate the preconditioner without affecting the quadratic convergence of the Newton loop. Furthermore, the MEBDF stages are non-linear systems with the same β coefficient of the implicit matrix (see Eq.(23)). Therefore, for small time steps, non-stiff problems or low-resolution discretizations, the MF approach is expected to effectively exploit the preconditioner freeze strategy. The freezing strategy considered in this paper consists in computing the preconditioner at the beginning of stage 1 every n time steps.

4.2.2 Matrix Explicit formulation

In this version of the implicit solver the system matrix is explicitly formed. This matrix is then used as preconditioner and to avoid the matrix-vector product approximation and the related residual function evaluation. Thus, the algorithm requires the storage of both the system matrix $\mathbf{W}$ and the preconditioning matrix $\mathbf{P}$. The system matrix is computed at the beginning of each time step and it is recomputed if the convergence of the Newton method becomes too slow. In particular, the implicit matrix is re-evaluated when the ratio between the L_2 norm of two successive Newton solution variations is less than 5.

5 RESULTS

Two test cases are presented to assess the temporal accuracy and the performance of the proposed MF-MEBDF scheme for the time-dependent compressible Euler and Navier-Stokes equations: an inviscid isentropic convecting vortex and a laminar Manufactured Solution, both with a known analytical solution. Furthermore, to demonstrate the performance advantages of the MF approach with the preconditioner freeze strategy, it will be compared with its ME counterpart and with the explicit 5-stage fourth order accurate Strong Stability Preserving Runge-Kutta scheme (RK4) [31] for non-stiff and stiff systems of equations. In particular, the convection of the isentropic vortex is investigated for high and low free stream Mach numbers, while the laminar computations are performed on uniform and stretched cartesian grids.

The simulations have been performed using the 3-step fourth-order accurate MEBDF scheme (MEBDF4) and the exact Riemann solver of Gottlieb and Groth [32] to approximate the inviscid numerical fluxes. As concerns the second stage's initial guess for the Newton-type solver, Eq. (14), the following fourth-order polynomial approximation is used:

$$\bar{\mathbf{Q}}^0_{n+2} = 4\bar{\mathbf{Q}}_{n+1} - 6\mathbf{Q}_n + 4\mathbf{Q}_{n-1} - \mathbf{Q}_{n-2}. \tag{26}$$

Another issue to be addressed using the MEBDF multi-step scheme concerns the unknown initial solutions needed to start the temporal integration. For both test cases the additional starting values are obtained evaluating the exact solution at the appropriate time levels. Furthermore, as a result of preliminary computations, it has established that a linear-solver normalized-residual tolerance of 10^{-2} allows for efficient computations while the Newton tolerance of one order of magnitude smaller than the solution error allows to obtain careful estimates of the

time integration error avoiding unnecessarily restrictive and computationally more expensive tolerance levels, especially at large time steps.

All the computations are performed in serial using a 2.80 GHz Intel Xeon CPU E5-2680v2 processor with 6.4 Gb of RAM memory.

5.1 Convection of an isentropic vortex

The convection of an inviscid isentropic vortex is a standard test case to evaluate the performance of the numerical methods. As the vortex is convected without distortion by the mean flow, the exact solution is obtained by translating the initial solution at the velocity set by the freestream, providing a reference for measuring the accuracy of the numerical solution. The initial conditions are taken from [33, 34].

The setup of the problem is as follows: the computational domain is defined as $0 \leq x, y \leq 10$ and the boundary conditions are periodic. Temporal convergence studies are conducted, up to a final time T corresponding to one period of vortex revolution, to assess the accuracy and the efficiency of the implicit time integration solvers. The simulations are performed on a 50×50 uniform cartesian grid with a seventh-order accurate polynomial approximations ($P6$ elements). The analysis is carried for stiff and non-stiff ODEs systems obtained by considering the following free stream Mach numbers: $M_\infty \simeq 1.4; 0.14; 0.014$.

5.1.1 High Mach number: non-stiff problems

The results of the convergence study at the highest Mach number ($M_\infty \simeq 1.4$) are presented in Fig. 2 for the MF, freezing the preconditioner, and the ME algorithms. The L_2-norm errors of the density field as a function of the time step are presented on the left while the work-precision characteristics on the right. The left plot demonstrates that the convergence histories computed using the two algorithms are indistinguishable. Furthermore, it can be seen that the design-order of convergence of 4 is achieved over the time interval that has been investigated. The right plot highlights that MF always outperforms ME. Moreover, it appears that using the proposed MF algorithm the efficiency, in terms of CPU time, significantly improves as the time step reduces. The results of both temporal refinement study and efficiency are collected in Tab. 1. To investigate the performance improvement of MF compared to ME, the CPU ratios and the optimal updating period of the preconditioner (n_{opt}), that minimizes the overall CPU time, are also reported in the table. It appears evident the benefit of the preconditioner freeze strategy, with the CPU ratio that increases with the growing of the n_{opt} parameter as the time step reduces. For example, to reach an error level of $8.10 \cdot 10^{-3}$ the MF algorithm is about 1.34 times faster than the ME algorithm, while for an accuracy level of $\mathcal{O}\left(10^{-10}\right)$ this ratio increases to 10.10.

The influence of the time consuming Jacobian evaluation on the efficiency of the implicit solver at large ($\Delta t = T/40$) and small ($\Delta t = T/160$) time steps is described in Tab. 2, where the performance of MF and ME algorithms are compared at each stage of the time scheme.

As regards the largest time step, at stage 1, when both the algorithms evaluate the Jacobian at the beginning of the Newton iterations, the performance of the two implicit solvers are roughly the same except for the number of function evaluations. Notice that the same can be observed in all stages of both time steps. The reason is in the additional residual evaluations necessary in the GMRES and Newton methods when the MF approach is used.

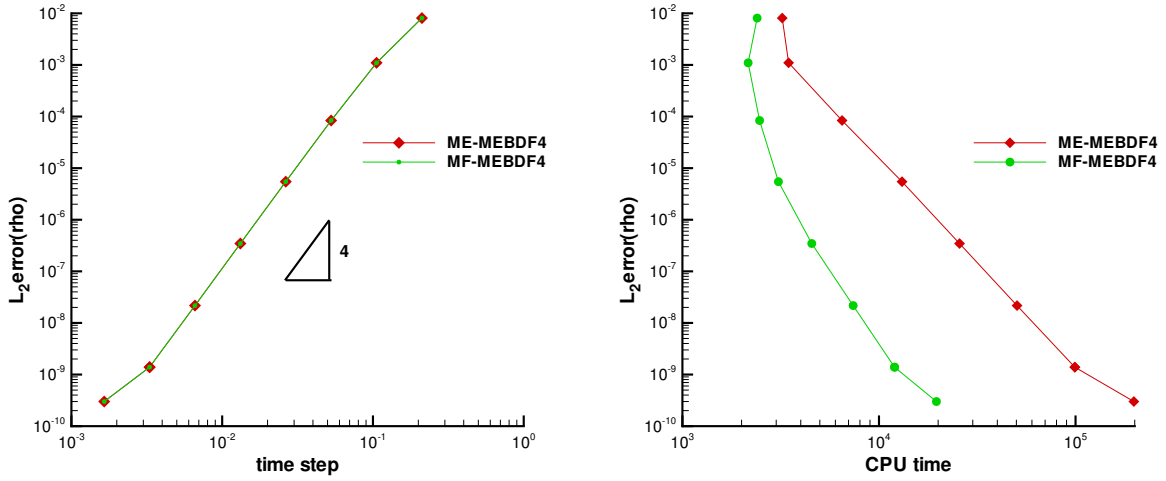

Figure 2: L_2-norm errors of the density field as a function of the time step (left) and of the CPU time (right). Comparison between MF, freezing the preconditioner, and ME algorithms at $M\infty \simeq 1.4$ using $P6$ elements.

				MF-MEBDF4		ME-MEBDF4	
$T/\Delta t$	Tol.$_{Newton}$	L_2 error(ρ)	order	n_{opt}	CPU(s)	CPU(s)	CPUs ratio
40	$8.10 \cdot 10^{-4}$	$8.10 \cdot 10^{-3}$		1	$2.40 \cdot 10^3$	$3.22 \cdot 10^3$	1.34
80	$1.10 \cdot 10^{-4}$	$1.10 \cdot 10^{-3}$	2.88	2	$2.16 \cdot 10^3$	$3.47 \cdot 10^3$	1.61
160	$8.38 \cdot 10^{-6}$	$8.38 \cdot 10^{-5}$	3.71	4	$2.47 \cdot 10^3$	$6.49 \cdot 10^3$	2.63
320	$5.47 \cdot 10^{-7}$	$5.47 \cdot 10^{-6}$	3.94	11	$3.08 \cdot 10^3$	$1.31 \cdot 10^4$	4.25
640	$3.46 \cdot 10^{-8}$	$3.46 \cdot 10^{-7}$	3.98	26	$4.55 \cdot 10^3$	$2.57 \cdot 10^4$	5.65
1280	$2.17 \cdot 10^{-9}$	$2.17 \cdot 10^{-8}$	3.99	39	$7.39 \cdot 10^3$	$5.04 \cdot 10^4$	6.82
2560	$1.39 \cdot 10^{-10}$	$1.39 \cdot 10^{-9}$	3.96	83	$1.20 \cdot 10^4$	$9.92 \cdot 10^4$	8.27
5120	$3.00 \cdot 10^{-11}$	$3.00 \cdot 10^{-10}$	2.21	115	$1.96 \cdot 10^4$	$1.98 \cdot 10^5$	10.10

Table 1: Newton tolerance, L_2-norm error of the density field, order of convergence and CPU time for different numbers of time steps. Comparison between MF, freezing the preconditioner for n_{opt} time steps, and ME algorithms at $M_\infty \simeq 1.4$ using $P6$ elements. In the last column the CPUs ratio between ME and MF is reported.

At stage 2, although no difference exists in the number of non-linear iterations, marked discrepancies appear in the number of linear iterations and Jacobian evaluations. The reason of the equal number of non-linear iterations between the two algorithms is due to the updating of the system matrix in the ME algorithm caused by the lack of/reduced convergence of this Newton loop. This is because at large time steps the system matrix is rapidly varying from stage 1 to stage 2. The significant increase in the number of linear iterations of the MF approach, about 7 times greater than that of its explicit counterpart, is due to a worse preconditioner with respect to the ME approach. In fact, using MF the preconditioner is never recomputed during a time step, unlike what happens in the ME algorithm, where the system matrix updating implies the recomputation of the preconditioner.

At stage 3 ME is less performant, in terms of number of non-linear/linear iterations, than MF because of the less effectiveness of the system matrix remained frozen from stage 2 and

not recomputed in this stage. On the contrary, the MF approach take advantages of the updated matrix-vector product at each Newton iteration and of the preconditioner evaluated at the first predictor stage, showing better convergence properties with respect to ME.

At $\Delta t = T/160$ the system matrix is slowly varying and it remains more effective even if frozen. Thus, by using the ME strategy the system matrix remains unchanged for all the time step without the need to be recomputed. The number of non-linear iterations is roughly the same for both algorithms in all stages. The significant increase in the number of linear iterations of MF with respect to ME is due to the effect of the frozen preconditioner. Note in fact that the number of total Jacobian evaluations for the MF approach is equal to 40 as the preconditioner is updated every 4 time steps. Furthermore, we can notice that the negative effect of the frozen preconditioner is more evident for stage 2 where at each time step the solution is predicted two time steps ahead with respect to the previous known solution at time t_n.

This stage analysis demonstrates that the MF implementation of the MEBDF scheme improves the robustness of the implicit solver. Furthermore, it can be argued that MF outperforms its explicit counterpart since at each time step the ME Jacobian evaluations exceed the cost of the larger number of GMRES iterations plus the additional cost of the function evaluations of the MF algorithm.

		MF-MEBDF4				ME-MEBDF4			
$T/\Delta t$	Stage	non-linear it.	linear it.	func. eval.	Jacob. eval.	non-linear it.	linear it.	func. eval.	Jacob. eval.
40	1	77	346	538	38	76	342	76	39
	2	114	3537	3803	0	114	495	114	38
	3	76	344	534	0	114	518	114	0
160	1	316	942	1732	40	316	632	316	158
	2	337	1438	2270	0	316	632	316	0
	3	316	933	1723	0	316	632	316	0

Table 2: Total number of non-linear iterations, linear iterations, function evaluations and Jacobian evaluations, at each stage and for large and small time steps. Comparison between MF, freezing the preconditioner, and ME.

However, in the above results the recomputed system/preconditioner matrix in the ME approach at $\Delta t = T/40$ and the use of the freezing strategy in the MF approach at $\Delta t = T/160$ do not allow to clarify the effect of the update of the matrix-vector product at each outer iteration on the performance of the implicit solver.

Therefore, the stage analysis is now carried-out at $\Delta t = T/160$ by comparing the performance of the algorithms updating the system/preconditioner matrix, for the ME approach, and the preconditioner matrix, for the MF approach, only at the beginning of stage 1. The data reported in Tab. 3, that refer to one time step iteration, show that at each stage the number of non-linear/linear iterations are almost the same for both algorithms, whereas the number of MF function evaluations is about $4 - 5$ times greater than that of ME. Nevertheless, the CPUs ratio indicates that the matrix free algorithm is slightly more efficient than its explicit counterpart. Thus, it can be concluded that the update of the matrix-vector product at each outer iteration allows to solve more efficiently the non-linear stages, compensating the cost of the additional function evaluations. We remark that this result mainly depends on the conditioning of the system matrix. Further studies are ongoing to investigate this aspect by extending the analysis to stiff systems and other polynomial degrees and by considering the set of primitive variables.

		MF-MEBDF4				ME-MEBDF4				
$T/\Delta t$	Stage	non-linear it.	linear it.	func. eval.	Jacob. eval.	non-linear it.	linear it.	func. eval.	Jacob. eval.	CPUs ratio
160	1	2	4	9	1	2	4	2	1	1.03
	2	2	5	10	0	2	4	2	0	
	3	2	4	9	0	2	4	2	0	

Table 3: Total number of non-linear iterations, linear iterations, function evaluations and Jacobian evaluations at each stage for one small time step. Comparison between MF and ME computing the system matrix only at the beginning of stage 1.

5.1.2 Low Mach numbers: stiff problems

The convection of the isentropic vortex is here investigated for different free stream Mach numbers to highlight pros and cons of the proposed time integration algorithm and its potential in solving non-stiff and stiff systems with respect to the explicit Runge-Kutta schemes. It is in fact well known that RK schemes become inefficient for the solution of stiff-problems, whereas the MEBDF method is well suited for ill-conditioned ODE systems.

In Tab. 4 the results obtained for high and low free stream Mach numbers by using MF-MEBDF4 and RK4 schemes are reported. As the vortex is simply convected, for the RK4 scheme a fixed time step size has been considered. To fairly compare the performance between the two time integration algorithms, at the highest Mach number the maximum cfl number ensuring that the error differs by less than 5% from the lowest achieveable error value ($2.99 \cdot 10^{-10}$) has been chosen for the explicit computations. For the lower Mach numbers the maximum cfl value allowed by the restrictive stability conditions has been used. Furthermore, a fine tuning, up to 2 significant digits, of the cfl number has been performed. For the MF-MEBDF4 scheme the fixed time step has been chosen such that, for the higher Mach numbers, very similar values of the error with respect to the ones obtained by the RK4 scheme are achieved ($\Delta\%$ error $= -2.88\%$ and 2.27% for $M_\infty \simeq 1.4$ and 0.14, respectively). For the lowest Mach number the error is even lower than that achieved by RK4, with $\Delta\%$ error$= -18.53\%$.

It is worth noting that the period of revolution of the vortex T varies according to the free stream velocity as highlighted in Tab. 4, where the number of time steps, $T/\Delta t$, and the time step size, Δt, are both shown. The last column of Tab. 4, where the ratio between the CPUs required by RK4 and MF-MEBDF4 is reported, highlights that RK4 largely outperforms MF-MEBDF4 for $M_\infty \simeq 1.4$, 0.14 (CPUs ratio = 0.03, 0.10). However, at $M_\infty \simeq 0.014$ the proposed time integration algorithm outperforms the explicit scheme with a CPUs ratio equal to 1.31, despite the lower error value computed.

	MF-MEBDF4					RK4					
M_∞	$T/\Delta t$	Δt	L_2 error(ρ)	n_{opt}	CPU (s)	$T/\Delta t$	Δt	L_2 error(ρ)	CPU (s)	$\Delta\%$ error	CPUs ratio
1.4	5000	$1.69 \cdot 10^{-3}$	$3.03 \cdot 10^{-10}$	100	$1.94 \cdot 10^{4}$	2381	$3.55 \cdot 10^{-3}$	$3.12 \cdot 10^{-10}$	$6.51 \cdot 10^{2}$	-2.88%	0.03
0.14	4000	$2.11 \cdot 10^{-2}$	$5.85 \cdot 10^{-10}$	50	$2.04 \cdot 10^{4}$	7875	$1.07 \cdot 10^{-2}$	$5.72 \cdot 10^{-10}$	$2.09 \cdot 10^{3}$	2.27%	0.10
0.014	640	1.32	$4.09 \cdot 10^{-7}$	4	$1.63 \cdot 10^{4}$	78741	$1.07 \cdot 10^{-2}$	$5.02 \cdot 10^{-7}$	$2.14 \cdot 10^{4}$	-18.53%	1.31

Table 4: Number of time steps, time step value, L2-norm error of the density field and CPU time obtained with MF-MEBDF4, freezing the preconditioner for n_{opt} time steps, and RK4 by using $P6$ elements and different M_∞ values. In the last two columns the percentage relative difference of the errors the and the CPUs ratio between RK4 and MF-MEBDF4 are reported.

5.2 Laminar manufactured solution

For viscous flows, the proposed time integration scheme has been evaluated on the following Manufactured Solution [35]

$$\rho(x,y,t) = \rho_0 \left[\sin\left(x^2+y^2+\omega t\right)+1.5\right] \quad (27)$$
$$u(x,y,t) = u_0 \left[\sin\left(x^2+y^2+\omega t\right)+\epsilon\right] \quad (28)$$
$$v(x,y,t) = v_0 \left[\cos\left(x^2+y^2+\omega t\right)+\epsilon\right] \quad (29)$$
$$e_t(x,y,t) = e_{t0} \left[\cos\left(x^2+y^2+\omega t\right)+1.5\right] \quad (30)$$

In the above equations x and y are the spatial coordinates of a generic point in the computational domain. Furthermore, the constants of Eqs. (27–30) are defined as: $\rho_0 = 1$, $u_0 = 0.1$, $v_0 = 0.001$, $e_{t0} = 1$, $\epsilon = 1/2$ and $\omega = 2\pi$.

The computational domain is a square of side one, the boundary conditions are time-dependent dirichlet boundary conditions and the simulation are performed up to a final time equal to one period T for progressively finer time step by using $P6$ elements. The numerical solution, starting from $t_0 = 0$, evolves in time according to the exact solution imposed by adding a source term to the Navier-Stokes equations. The reader is referred to the Appendix D of [35] for the vector of source terms.

The results obtained by the proposed time integration scheme are evaluated for different discretizations of the computational domain by using uniform and stretched cartesian grids. In particular, the cartesian grid used are: $20 \times 20; 5 \times 80; 2 \times 200$. These grids have the same total number of elements but are characterized by different Aspect Ratio (AR): $1; 16; 100$.

5.2.1 Cartesian elements: non-stiff problems

The results obtained by performing a temporal refinement study on the cartesian uniform grid ($AR = 1$) by using the MF approach are here presented and compared with the the results obtained with the ME approach.

In the left plot of Fig. 3 the L_2-norm error of the density field as a function of the time step is presented while in the right plot the work-precision characteristics are shown. As for the inviscid test case, the convergence histories obtained using the two algorithms are indistinguishable and achieve the design order convergence of 4. Furthermore, the performance advantages of the MF approach with respect to the ME approach are remarkable and become increasingly more significant as the time step reduces. As it can be seen form Tab. 5 , where the above results are summarized, the MF algorithm is 2.76-7.84 times faster than the ME approach.

Finally, we remark that the analysis carried out to investigate the influence of both the MF approach and the freeze strategy, on the performance of the implicit solver, is not presented since the results are similar to those shown for the inviscid test case.

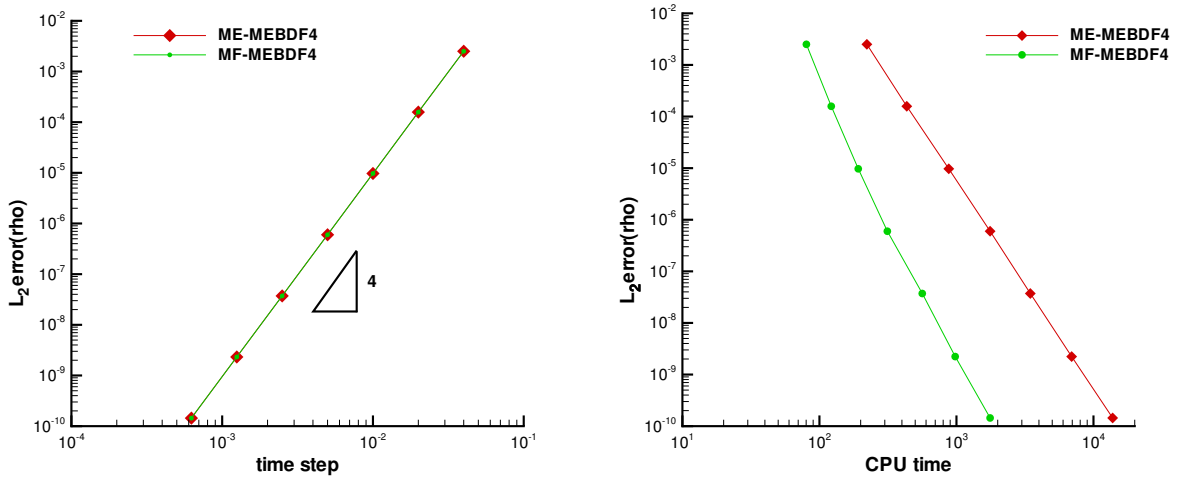

Figure 3: L_2-norm errors of the density field as a function of the time step (left) and of the CPU time (right). Comparison between MF, freezing the preconditioner, and ME algorithms on the cartesian grid with $AR = 1$ using $P6$ elements

				MF-MEBDF4		ME-MEBDF4	
$T/\Delta t$	Tol.$_{Newton}$	L_2 error(ρ)	order	n_{opt}	CPU(s)	CPU(s)	CPUs ratio
25	$2.51 \cdot 10^{-4}$	$2.51 \cdot 10^{-3}$	3.99	5	$8.03 \cdot 10^{1}$	$2.22 \cdot 10^{2}$	2.76
50	$1.58 \cdot 10^{-5}$	$1.58 \cdot 10^{-4}$	4.02	10	$1.22 \cdot 10^{2}$	$4.34 \cdot 10^{2}$	3.56
100	$9.71 \cdot 10^{-7}$	$9.71 \cdot 10^{-6}$	4.02	20	$1.92 \cdot 10^{2}$	$8.79 \cdot 10^{2}$	4.58
200	$5.99 \cdot 10^{-8}$	$5.99 \cdot 10^{-7}$	4.01	35	$3.13 \cdot 10^{2}$	$1.76 \cdot 10^{3}$	5.62
400	$3.72 \cdot 10^{-9}$	$3.72 \cdot 10^{-8}$	4.00	67	$5.63 \cdot 10^{2}$	$3.46 \cdot 10^{3}$	6.15
800	$2.32 \cdot 10^{-10}$	$2.32 \cdot 10^{-9}$	4.01	140	$9.79 \cdot 10^{2}$	$6.93 \cdot 10^{3}$	7.08
1600	$1.44 \cdot 10^{-11}$	$1.44 \cdot 10^{-10}$		200	$1.76 \cdot 10^{3}$	$1.38 \cdot 10^{4}$	7.84

Table 5: Newton tolerance, L_2-norm error of the density field, order of convergence and CPU time for different numbers of time steps. Comparison between MF, freezing the preconditioner for n_{opt} time steps, and ME algorithms on the grid with $AR = 1$ using $P6$ elements. In the last column the CPUs ratio between ME and MF is reported.

5.3 Stretched elements: stiff problems

The comparison between the performance of MF-MEBDF4 and RK4 algorithms is reported in Tab. 6. Unlike the inviscid test case, the shape of the solution in this case is varying over time. Therefore, to fairly compare the performance of the two time integration schemes, the RK4 time integration uses a variable time step.

For the uniform grid case ($AR = 1$), according to Subsection 5.1.2, the maximum cfl number ensuring that the error differs by less than 5% from the lowest achievable error value ($1.44 \cdot 10^{-10}$) has been chosen to not penalize the performance of the explicit computation. For the higher aspect ratios ($AR = 16$ and $AR = 100$) the maximum cfl numbers ensuring stability have been used. We remark that, also in this case the cfl number has been tuned up to 2 significant digits. For the computations that refer to the MF-MEBDF4 algorithm the

largest time step values that ensure almost the same error levels of the corresponding RK4 solutions have been adopted ($\Delta\%$ error $= -4.00\%$, -0.33% and 0.47% for $AR = 1$, 16 and 100, respectively). The entries of the last column of Tab. 6, where the ratio between the CPUs required by MF-MEBDF4 and RK4 is reported, show that the RK4 scheme outperforms the MF-MEBDF4 scheme on the uniform cartesian grid (CPUs ratio = 0.49). However, increasing the ill-conditioning by increasing the aspect ratio, the severe time step restriction of the explicit scheme leads to an inefficient time integration, that becomes more expensive as the stiffness increases. At the intermediate AR level of 16 the MF-MEBDF4 becomes 1.10 times faster than RK4. Increasing the aspect ratio to 100 the remarkable speed-up of 18.64 is achieved.

	MF-MEBDF4				RK4				
AR	$T/\Delta t$	L_2 error(ρ)	n_{opt}	CPU (s)	$T/\Delta t_{ave}$	L_2 error(ρ)	CPU (s)	$\Delta\%$ error	CPUs ratio
1	1600	$1.44 \cdot 10^{-10}$	200	$1.76 \cdot 10^{3}$	3273	$1.50 \cdot 10^{-10}$	$8.66 \cdot 10^{2}$	-4.00%	0.49
16	1600	$9.05 \cdot 10^{-10}$	58	$1.73 \cdot 10^{3}$	7122	$9.08 \cdot 10^{-10}$	$1.90 \cdot 10^{3}$	-0.33%	1.10
100	400	$4.30 \cdot 10^{-7}$	28	$5.58 \cdot 10^{2}$	38532	$4.28 \cdot 10^{-7}$	$1.04 \cdot 10^{4}$	0.47%	18.64

Table 6: Number of time steps, L2-norm error of the density field and CPU time obtained with MF-MEBDF4, freezing the preconditioner for n_{opt} time steps, and RK4 by using $P6$ elements and cartesian grids with different aspect ratio (AR). In the last two columns the percentage relative difference of the errors and the CPUs ratio between RK4 and MF-MEBDF4 are reported.

6 CONCLUSIONS

We have presented a matrix-free (MF) preconditioned Newton/GMRES method for the solution of the unsteady compressible Navier-Stokes equations. To ensure high accuracy and efficiency of the proposed implicit method, the three-stage fourth-order accurate MEBDF time integration scheme has been coupled with a high-order accurate DG discretization. The MF solver uses less computer memory than the corresponding matrix-explicit (ME) version since the system matrix is stored only for preconditioning purpose. Furthermore, MF-MEBDF outperforms its explicit counterpart for the problems here investigated. In particular, MF-MEBDF is more robust and faster than ME-MEBDF for large time steps. In these cases, in fact, ME-MEBDF shows a lack of/reduced convergence and requires the time consuming updating of the system matrix. At small time steps, when the system matrix is slowly varying and remains effective for all non-linear stages, MF-MEBDF slightly outperforms ME-MEBDF, although for the matrix-free computations a larger number of function evaluations with respect to ME are required. These findings allow to conclude that the MF-MEBDF algorithm benefits from the inherent update of the matrix-vector product at each Newton iteration. This mainly depends on the multi-stage structure of the time scheme and on the conditioning of the system matrix. We remark that a seventh-order accurate DG discretization has been adopted in our computations, leading to ill-conditioned stage matrices. Thus, a better performance is expected to be achieved when additional sources of ill-conditioning are considered, such as low speed flows, RANS equations and stretched grids. The proposed MF algorithm, equipped with a preconditioner freeze strategy, shows a noteworthy computational time reduction with respect to ME for non-stiff ODE systems, with an increasing CPUs ratio, ranging from 1.34 to 10.10, as the time step reduces. Furthermore, MF-MEBDF outperforms the five-stage fourth-order accurate Runge-Kutta scheme in solving stiff systems, with an increasing saving of computational time as the stiffness increases. For the inviscid vortex transport test-case, the CPUs ratio of 1.31 is achieved at $M \simeq 0.014$. For the laminar test case the speed up ratio significantly improves with

the grid aspect ratio from 1.10 ($AR = 16$) to 18.64 ($AR = 100$).

Further work is ongoing to analyze the performance of the solver for different polynomial degrees and by considering the set of primitive variables.

REFERENCES

[1] Cockburn B., Karniadakis G.E., Shu C.W. Eds. (2000): Discontinuous Galerkin Methods. Theory, Computation and Applications. New York: Springer Verlag, vol. 11, Lect. Notes Comput. Sci. Eng.

[2] Bassi F., 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, 267–279.

[3] Luo H., Baum J.D., Lohner R. (2006): A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. J. Comput. Phys. 211, 767–783.

[4] Cockburn B. (2001): Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws. J. Comput. Appl. Math. 128, 187–204.

[5] Nigro A., Renda S., De Bartolo C., Hartmann R., Bassi F. (2013): A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows. Internat. J. Numer. Methods Fluids 72 (1), 43–68.

[6] Bassi F., Botti L., Colombo A., Ghidoni A., Massa F. (2015): Linearly implicit Rosenbrock-type Runge-Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows. Computer & Fluids 118, 305–320.

[7] Nigro A., Ghidoni A., Bassi F., Rebay S. (2014): Modified extended BDF scheme for the Discontinuous Galerkin solution of unsteady compressible flows. Int. J. Numer. Methods Fluids 76, 549–574.

[8] Nigro A., De Bartolo C., Bassi F., Ghidoni A. (2014): Up to sixth-order accurate A-stable implicit schemes applied to the Discontinuous Galerkin discretized Navier-Stokes equations. J. Comput. Phys. 276, 136–162.

[9] Gear C.W., Saad Y. (1983): Iterative solution of linear equations in ODE codes. SIAM Journal on Scientific and Statistical Computing 4, 583–601.

[10] Chan T.F., Jackson K.R. (1984): Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms. SIAM Journal on Scientific and Statistical Computing. 5, 533–542.

[11] Brown P.N., Hindmarsh A.C. (1986): Matrix-free methods for stiff systems of ODEs. SIAM Journal on Numerical Analysis. 23, 610–638.

[12] Johan J. (1992): Data parallel finite element techniques for large scale computational fluid dynamics. PhD thesis, Department of Mechanical Engineering, Stanford University, U.S.A, (Chapter 2).

[13] McHugh P.R., Knoll D.A. (1994): Comparison of standard and matrix-free implementations of several Newton-Krylov solvers. AIAA Journal. 32, 2394–2400.

[14] McHugh P.R., Knoll D.A. (1993): Inexact Newton's method solutions to the incompressible Navier-Stokes and energy equations using standard and matrix-free implementations. AIAA Paper, 93–3332.

[15] Rasetarinera P., Hussaini M.Y. (2001): An efficient implicit discontinuous spectral Galerkin method. J Comput Phys. 172(2), 718–738.

[16] Hillewaert K., Remacle J.F., Chevaugeon N., Bernard P.E., Geuzaine P. (2006): Analysis of a hybrid P-multigrid method for the discontinuous Galerkin discretisation of the Euler equation. In: Wesseling P., Onate E., Periaux J. (Eds.), ECCOMAS CFD Conference2006, TU Delft, The Netherlands.

[17] Crivellini A., Bassi F. (2011): An implicit matrix-free Discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. Computer & Fluids. 50, 81–93

[18] Hosseini S.M., Hojjati G. (1999): Matrix free MEBDF method for the solution of stiff systems of ODEs. Mathematical and Computer Modelling 29(4), 67–77

[19] Tesini P. (2008): An h-Multigrid Approach for High-Order Discontinuous Galerkin Methods. Ph.D. thesis, University of Bergamo, Bergamo, Italy.

[20] Bassi F., Botti L., Colombo A., Di Pietro D.A., Tesini P. (2012): On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. J. Comput. Phys. 231 (1), 45–65.

[21] Di Pietro D.A., Ern A. (2012): Mathematical Aspects of Discontinuous Galerkin Methods. Vol. 69 of Mathematiques et Applications, Springer-Verlag.

[22] Bassi F., Rebay S., Mariotti G., Pedinotti S., Savini M. (1997): A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: R. Decuypere and G. Dibelius (Eds.), Proceeding of the 2nd European Conference on turbomachinery Fluid Dynamics and Thermodynamics, pp. 99-108, Antwerpen, Belgium, Technologisch Instituut

[23] Bassi F., Rebay S. (2000): A high order discontinuous Galerkin method for compressible turbulent flows. In: Discontinuous Galerkin Methods: Theory, Computation and Applications, B. Cockburn, G. Karniadakis, and C.-W. Shu (Eds.), Springer, 11: 77–88.

[24] Brezzi F., Manzini M., Marini D., Pietra P., Russo A. (2000): Discontinuous Galerkin approximations for elliptic problems. Numer. Methods Partial Differential Equations, 16: 365–378.

[25] Arnold D.N., Brezzi F., Cockburn B., Marini D.(2002): Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal., 39(5): 1749–1779.

[26] Nigro A., Ghidoni A., Rebay S., Bassi F. (2012): High-order discontinuous Galerkin solution of unsteady problems using Modified Extended Backward Differentiation Formulae. In: 6th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2012, pp. 6672–6690.

[27] Cash J.R. (1983): The integration of stiff initial value problems in ODEs using Modified Extended Backward Differentiation Formulae. Computers & Mathematics with Applications, 5 (9) pp. 645–657.

[28] Cash J.R. (2000): Modified extended backward differentiation formulae for the numerical solution of stiff initial value problems in ODEs and DAEs. Journal of Computational and Applied Mathematics, 125(1-2), pp. 117–130.

[29] Balay S., Abhyankar S., Adams M.F., Brown J., Brune P., Buschelman K., Dalcin L., Eijkhout V., Gropp W.D., Kaushik D., Knepley M.G., McInnes L.C., Rupp K., Smith B.F., Zampini S., Zhang H.(2016): Petsc-web-page: http://www.mcs.anl.gov/petsc

[30] Pernice M., Walker H.F. (1998): NITSOL: a Newton iterative solver for nonlinear systems. SIAM J. Sci . Comput. 19, 302–318

[31] Spiteri R.J., Ruuth S.J. (2002): A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. 40(2), 469–491.

[32] Gottlieb J.J., Groth C.P.T. (1988): Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. J. Comput. Phys. 78, 437–458.

[33] Nigro A., De Bartolo C., Bassi F., Ghidoni A. (2014): High-order discontinuous Galerkin solution of unsteady flows by using an advanced implicit method. Lecture Notes in Computational Science and Engineering 99, 135–149.

[34] Bassi F., Botti L., Colombo A., Crivellini A., Ghidoni A., Nigro A., Rebay S. (2015): Time integration in the Discontinuous Galerkin code MIGALE - Unsteady Problems. In: Kroll N., Hirsch C., Bassi F., Johnston C., Hillewaert K.(Eds.), IDIHOM: Industrialization of high-order methods - A top-down approach. Vol. 128 of Notes on numerical fluid mechanics and multidisciplinary design, Springer International Publishing, pp. 205–230.

[35] Salari K., Knupp. P. (2000): Code Verification by the Method of Manufactured Solutions. SAND 2000-1444, Sandia National Laboratories, Albuquerque, NM.