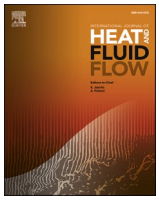
International Journal of Heat and Fluid Flow 121 (2026) 110560



Contents lists available at ScienceDirect

International Journal of Heat and Fluid Flow

journal homepage: www.elsevier.com/locate/ijhff





Linearly implicit rosenbrock methods for unsteady flow simulations☆

Johannes Löwe [a,*], Anton Schmid [b]

[a] German Aerospace Center (DLR), Bunsenstraße 10, 37073 Göttingen, Germany
[b] German Aerospace Center (DLR), Nöthnitzer Straße 46b, 01187 Dresden, Germany

A B S T R A C T

In this work, we present details on the implementation of linear Rosenbrock-type methods and their application to unsteady flow problems. In recent years, especially in the field of high-order methods, diagonally implicit Runge-Kutta (DIRK) methods have gained popularity. However, linear Rosenbrock methods are an easier-to-implement and −to-use alternative. Their implementation requires little more than what is already commonly available in most flow solvers. We present results covering direct numerical simulation (DNS), large-eddy simulation (LES), and unsteady Reynolds-averaged Navier-Stokes (URANS) applications and show that when combined with a second-order finite volume method, second-order Rosenbrock schemes provide sufficient accuracy. In particular, the two-stage second-order scheme (ROS2) shows simultaneously increased performance and accuracy compared to the popular second-order backward differentiation formula (BDF2). At the same time, it consistently outperforms similar DIRK schemes in our implementation. As an addition to a legacy finite volume (FV) solver, Rosenbrock methods can increase performance and accuracy and extend its range of applicability and lifespan.

1. Introduction

In computational fluid dynamics (CFD), unsteady flow phenomena that occur at the edge of the flight envelope, maneuver simulations with dynamic control surface deflections or gust encounters, as well as scale-resolving simulations of turbulent flow promote the need for efficient unsteady simulation capabilities over a broad range of use cases.

Most relevant applications have multi-scale characteristics in space and time. Usually, there is interest only in the time-accurate resolution of some physical phenomena, e.g., convective transport of resolved turbulence or shock oscillations, while others are mostly stationary or simply not resolved. Many promising implementations stop after implementing the most basic explicit Runge-Kutta schemes. For unsteady Reynolds-averaged Navier-Stokes (URANS) methods or hybrid RANS/LES methods (HRLM) in automotive or aerospace applications at high Reynolds numbers, the timestep size restrictions for explicit methods imposed by strongly anisotropic boundary-layer-adapted grids are prohibitive and thereby severely limit the applicability of these methods. Stiff nonlinear source terms in eddy viscosity-based turbulence models or fast chemical reactions can also be contributing factors.

For many CFD codes using a second-order finite volume method, the second-order backward differentiation formula (BDF) has been a cornerstone in unsteady applications. Especially its simplicity and strong stability properties might have contributed to its popularity.

With the advent of higher-order spatial discretization methods, implicit Runge-Kutta methods gained traction. Using multiple stages per timestep, they possess many free parameters that can be used to optimize for various different aspects besides just consistency order. However, they also increase the implementation complexity. Moreover, care has to be taken regarding the reduction of nonlinear residuals within stages; otherwise, important properties might be lost.

To avoid some of the complications of implicit nonlinear time integration methods, H. Rosenbrock suggested applying a fixed number of linearization steps. The methods quickly evolved beyond that basic idea, e.g., by an extension by G. Wanner, and became their own field of research. For a complete historical overview, we refer to Lang (2021).

In this work, we show that Rosenbrock methods are an easy-to-implement solution of low complexity that can greatly expand a code's capabilities. Furthermore, we compare their performance with that of other implicit methods on select test cases that cover direct numerical simulation (DNS), large-eddy simulation (LES), and URANS applications.

2. Time integration

Applying only the spatial discretization to the partial differential equations governing compressible flow, e.g., the Navier-Stokes equations, transforms the problem into a system of ordinary differential equations

☆ This article is part of a special issue entitled: 'AERO2025' published in International Journal of Heat and Fluid Flow.

* Corresponding author.
E-mail addresses: Johannes.loewe@dlr.de (J. Löwe), Anton.Schmid@dlr.de (A. Schmid).

https://doi.org/10.1016/j.ijheatfluidflow.2026.110560
Received 30 November 2025; Received in revised form 22 June 2026; Accepted 28 June 2026
Available online 1 July 2026
0142-727X/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

$$u'(t) = f(u(t)),\ u : \mathbb{R} \to \mathbb{R}^N,\ f : \mathbb{R}^N \to \mathbb{R}^N, \tag{1}$$

where the dimension N depends on the spatial discretization and grid used. For a finite-volume (FV) discretization of the compressible Navier-Stokes equations, it is five times the number of grid nodes or elements, for a cell-vertex or cell-centered scheme, respectively.

Eq. (1) together with an initial condition $u(t_0) = u_0$ forms an initial value problem. For steady-state problems, where we solve for a solution with $u'(t) = 0$, f is often called the stationary residual.

For finite-volume and discontinuous Galerkin (DG) discretization methods, the element-local mass matrices typically appearing on the left-hand side of Eq. (1) can be locally inverted, premultiplied and incorporated into f. Variables (or degrees of freedom) associated with strong algebraic Dirichlet boundary conditions can be eliminated from the vector of unknown variables u. For simplicity, we only consider autonomous systems, where f does not directly depend on the time t. If needed, the time t can be added to the vector of unknowns u with trivial time derivative $t' = 1$ on the right-hand side.

We consider a given time grid $\{t_0 = 0 < t_1 < \cdots < t_M = T\}$ and denote the time step size $\Delta t_n = t_n - t_{n-1}$, but usually omit the subscript n for brevity. The time grid can also be generated during the simulation, driven by an error estimation procedure, e.g., using a secondary embedded method. In this work, we consider only time grids with constant $\Delta t = \Delta t_n$.

Looking for an approximation $u_{n+1} \approx u(t_{n+1})$, a general Runge-Kutta scheme, with s stages, is given by:

$$u_{n+1} = u_n + \Delta t \sum_{j=1}^{s} b_j k_j \tag{2a}$$

$$k_i = f\left(u_n + \Delta t \sum_{j=1}^{s} a_{ij} k_j\right),\ i = 1, \cdots, s \tag{2b}$$

The coefficients are collected in a Butcher tableau $\boldsymbol{A} = (a_{ij})$, $\boldsymbol{b} = (b_j)$. These methods are one-step methods that directly advance from time level n to time level $n+1$ without older time levels appearing in the equations. Therefore, no start-up procedure is required; only a single initial state is needed.

If $a_{ij} = 0$ for $j \geq i$, then all k_i for $i = 1, \cdots, s$ can be computed sequentially by explicit evaluations of f. These methods are called explicit Runge-Kutta (ERK) schemes. Since second-order explicit schemes with only two stages have unfavorable stability properties, we only consider the often-used classical ERK of fourth order RK4 given by

$$\boldsymbol{A} = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ 1/2 & 0 & \ddots & \vdots \\ 0 & 1/2 & 0 & \vdots \\ 0 & 0 & 1/1 & 0 \end{pmatrix},\ \boldsymbol{b}^T = \begin{pmatrix} 1/6 \\ 1/3 \\ 1/3 \\ 1/6 \end{pmatrix}. \tag{3}$$

If $a_{ij} = 0$ for $j > i$, then for each stage a single nonlinear system of dimension N needs to be solved. These methods are called diagonally implicit Runge-Kutta (DIRK) schemes. A DIRK scheme with a single constant diagonal coefficient $a_{ii} = \gamma$ for all stages is called an SDIRK scheme and the Jacobian used in a quasi-Newton method can be shared between all stages.

In this work we use the first-order implicit Euler method and the second-order midpoint (MP) scheme, both single-stage schemes, defined by

$$\boldsymbol{A}_{\text{IE}} = (1),\ \boldsymbol{b}_{\text{IE}}^T = (1),\ \boldsymbol{A}_{\text{MP}} = (1/2),\ \boldsymbol{b}_{\text{MP}}^T = (1) \tag{4a}$$

and the 2-stage second-order DIRK scheme by Alexander (1977), denoted by A22, given by

$$\boldsymbol{A}_{\text{A22}} = \begin{pmatrix} \gamma_{A22} & 0 \\ 1 - \gamma_{A22} & \gamma \end{pmatrix},\ \boldsymbol{b}_{\text{A22}}^T = \begin{pmatrix} 1 - \gamma_{A22} \\ \gamma_{A22} \end{pmatrix},\ \gamma_{A22} = 1 - \sqrt{1/2}. \tag{4b}$$

Table 1
Time integration schemes used and some of their properties.

Name	lin.	s	p	γ	stiff. acc.	Eqs. & Coeffs.	Ref.
IE	×	1	1	1.0	✓	2, 4a	Hairer and Wanner (1996), p. 3
MP	×	1	2	0.5	×	2, 4a	Hairer and Wanner (1996), p. 142
LMP	✓	1	2	0.5	×	5, 7a	Baker and Iannelli (2012)
BDF2/ TPB	×	1	2	0.6667	×	8	Hairer and Wanner (1996), p. 263
A22	×	2	2	0.2929	✓	2, 4b	Hairer and Wanner (1996), p. 106
ROS2	✓	2	2	0.2929	✓	5, 7b	Baker and Iannelli (2012)
ROS34PR	✓	4	3	0.4357	✓	5, Ref.	Rang (2013)
RODAS	✓	6	4	0.25	✓	5, Ref	Bassi et al. (2015)
exp. RK4	–	4	4	–	×	2, 3	Hairer and Wanner (1996), p. 18

Without restrictions on the coefficients, a single nonlinear system of dimension sN needs to be solved. Solving these nonlinear systems is usually done by either dual-time stepping, i.e., adding another pseudo-time derivative to Eq. (2b) and using an explicit time integration method to converge to a steady state in pseudo-time, or a (quasi-) Newton method. The latter provides potentially much faster convergence, but requires a series of linear systems to be solved.

The large gap in computational effort and algorithmic complexity between explicit and nonlinear implicit Runge-Kutta methods can be filled by Rosenbrock methods (Hairer and Wanner, 1996). In an alternative formulation taken from Bassi et al. (2015) that is more efficient to implement compared to the original formulation, these are given by:

$$u_{n+1} = u_n + \sum_{j=1}^{s} m_j y_j \tag{5a}$$

$$\left(\frac{I}{\gamma \Delta t} - J_f^n\right) y_i = f\left(u_n + \sum_{j=1}^{i-1} a_{ij} y_j\right) + \sum_{j=1}^{i-1} \frac{c_{ij}}{\gamma \Delta t} y_i \tag{5b}$$

for $i = 1, \cdots, s$. The coefficients $\boldsymbol{A} = (a_{ij})$, $\boldsymbol{C} = (c_{ij})$ and $\boldsymbol{m} = (m_i)$ can be computed from the standard coefficients $\mathrm{A} = (\alpha_{ij})$, $\Gamma = (\gamma_{ij})$ and $\boldsymbol{b} = (b_i)$ usually provided in literature by

$$\boldsymbol{A} = \mathrm{A}\Gamma^{-1},\ \boldsymbol{C} = I - \mathrm{diag}(\Gamma)\Gamma^{-1},\ \boldsymbol{m}^T = b^T\Gamma^{-1}. \tag{6}$$

Note that $\boldsymbol{A}$ and $\boldsymbol{C}$ are always strictly lower triangular. As for SDIRK methods, usually only schemes with a single diagonal coefficient $\gamma = \gamma_{ii}$ are considered. We use the single stage Rosenbrock-type linearized midpoint (LMP) scheme

$$\gamma_{\text{LMP}} = 1/2,\ \boldsymbol{A}_{\text{LMP}} = 0,\ \boldsymbol{m}_{\text{LMP}}^T = (2),\ \boldsymbol{C}_{\text{LMP}} = 0 \tag{7a}$$

and the 2-stage second-order scheme ROS2 from Bassi et al. (2015) given by

$$\boldsymbol{A}_{\text{ROS2}} = \begin{pmatrix} 0 & 0 \\ 4 - 8\gamma_{ROS2} & 0 \end{pmatrix},\ \boldsymbol{m}_{\text{ROS2}}^T = \begin{pmatrix} 1 - 1/8\gamma_{ROS2} \\ 1/8\gamma_{ROS2}^2 \end{pmatrix},\ \boldsymbol{C}_{\text{ROS2}} = 0, \tag{7b}$$

with $\gamma_{ROS2} = \gamma_{A22}$ from Eq. (4b). For schemes with more stages, we do not give all the coefficients here, but refer to the references given in Tab.1.

Eqs. (5b) have a structure similar to classical Runge-Kutta schemes in

Eqs. (2b), but with the Jacobian $J_f^n = J_f(u_n) = \frac{\partial f}{\partial u}(u_n)$ appearing on the left-hand side of the equation and an additional linear combination of already computed stage solutions on the right-hand side comparable to the Shu-Osher form of ERK schemes (Shu & Osher, 1988). Rosenbrock methods, as given in Eqs. (5b), require solving a single system of linear equations of dimension N and a single evaluation of f to prepare the right-hand side per stage.

An alternative to RK methods is linear multistep methods. A very popular example is the second-order backward differentiation formula (BDF), also commonly referred to as the three-point backward (TPB) method, given by:

$$\frac{u_{n+1} - 4/3u_n + 1/3u_{n-1}}{2/3\Delta t} = f(u_{n+1}). \tag{8}$$

For the BDF2 method a single nonlinear system of equations of dimension N needs to be solved. BDF methods of higher order are usually not considered, because they are not A-stable.

An overview of the time integration methods used in this work with their basic properties is given in Tab. 1, including the number of stages s, the consistency order p, the effective timestep size coefficient γ, and whether it is linear or stiffly accurate, together with a reference. Due to the large number of coefficients for the ROS34PR and RODAS methods, we do not reproduce them here, but refer to the references given in Tab. 1.

3. Linear systems

Applying a (quasi-)Newton method to the nonlinear equations arising from SDIRK schemes (Eqs. (2b) or from the BDF2 scheme (Eq. (8) leads to basically identical linear systems of equations as those in Eqs. (5b) for Rosenbrock methods with $\left(\frac{I}{\gamma\Delta t} - J_f^n\right)$ as the linear operator and varying right-hand sides.

The effective implicit timestep size $\gamma\Delta t$ is the only significant differentiating factor and plays an important role in relaxing the stiffness of the Jacobian. The A22 DIRK method and the ROS2 method use the same value of about $0.293\Delta t$, while the BDF2 method has $0.667\Delta t$ as its effective timestep size. This will later also be visible in a larger linear solver iteration count for BDF2. For most relevant applications, the largest feasible timestep size is used and the time integration is not operating in the asymptotic regime of convergence. This is also the reason why, in our experience, methods that maximize the consistency order at the cost of the size of the diagonal coefficient perform poorly in many applications. An example is the three-stage fourth-order method by Crouzeix and Nørsett (Crouzeix, 1976), which, despite using three stages, has an effective implicit timestep size of approximately $1.07\Delta t$, even larger than the physical timestep size. This leads to large error constants and increased stiffness of the linear system of equations.

3.1. Linear solver

A direct linear solver with a constant, condition-number-independent runtime would be ideal. However, for relevant problem sizes in three-dimensional space, such solvers are prohibitively expensive in terms of storage requirements and computational cost and are therefore not viable.

Instead, we opt for iterative Krylov subspace methods. For a comprehensive overview of these methods, we refer to Saad (2003). We use the Generalized Minimal Residual (GMRES) method without restarts and with a maximum iteration number (equivalent to Krylov subspace size) of 100 but usually aim for 50 or fewer iterations. If the desired level of convergence cannot be achieved within the targeted iteration number, preconditioning needs to be added or strengthened. Using larger Krylov subspace sizes leads to long runtimes due to unfavorable nonlinear runtime complexity.

For steady-state problems, the global couplings significantly contribute to the stiffness of the linear problems. However, for unsteady problems, the necessity to resolve all relevant physical processes in time restricts the timestep size and puts an upper bound on the stiffness of the linear problems. Therefore, we did not consider multigrid (MG) methods with large setup cost and high implementation complexity for the unsteady simulations presented below.

3.2. Implementation of the Jacobian

Iterative Krylov methods rely on matrix–vector products and inner products only and do not need access to the linear operator in matrix form (i.e. matrix entries). For the Jacobian term J_f^n, this means that we only need to be able to evaluate directional derivatives in direction d. In some programming languages this can be easily achieved without manually implementing derivatives by using the forward mode of algorithmic differentiation (AD) libraries, e.g. Eigen-AD (Peltzer et al., 2020). These utilize a data type that stores the state and derivative at the same time and has all (arithmetic) operations used in the residual/flux computation overloaded. Efficient AD libraries employed during code generation/compilation can usually compute directional derivatives in forward mode at the cost of about two residual computations, while yielding exact (up to machine precision) results.

However, when such a third-party library-based approach is not available, the directional derivative can also be computed using a basic finite difference (FD) approximation. If this is one-sided, i.e.

$$J_f^n \cdot d \approx \frac{f(u_n + hd) - f(u_n)}{h}, \tag{9}$$

with step size parameter h and direction d, then $f(u_n)$ in the numerator can be stored and only a single additional residual evaluation $f(u_n + hd)$ is required for each new directional derivative, corresponding to a matrix–vector product in a Krylov solver iteration. For a central finite difference approximation (not shown here) this cost is doubled. The step size parameter h is chosen as

$$h = \epsilon \frac{\|u_n\|}{\|d\|} \tag{10}$$

where $\epsilon = 10^{-7}$ is used for a one-sided approximation, while $\epsilon = 10^{-5}$ is used for a central approximation. These values are roughly optimized to balance linearly or quadratically decreasing truncation errors and linearly increasing cancellation errors in Eq. (9). To enforce physical constraints on variables in $u_n + hd$ (e.g. positivity), the step size parameter h might be reduced further. However, for all cases considered in this work, this was not required.

3.3. Preconditioning

In our experience, for uniform grids and small timestep sizes, preconditioning is not strictly required. This, however, changes drastically when moving to grids with anisotropic boundary layer resolution. The pressure coupling (especially at low Mach numbers) and a large effective viscosity in the turbulent boundary layer contribute to the stiffness of the linear operator, even when no significant unsteady effects need to be resolved.

Solving tridiagonal systems along grid lines aligned with the grid anisotropy by Gaussian elimination as a preconditioner essentially eliminates this stiffness from the linear system. However, it is important that the coupling along these lines is preserved during partitioning for parallel processing. The matrix used in the preconditioner can be based on a simplified first-order approximation with a compact stencil. Updates to the preconditioner can be delayed for several time steps (often 10 or more), spreading the matrix assembly and setup costs over as many time steps as possible. This reduces the setup cost to an insignificant fraction of the total computation time, without adversely affecting the

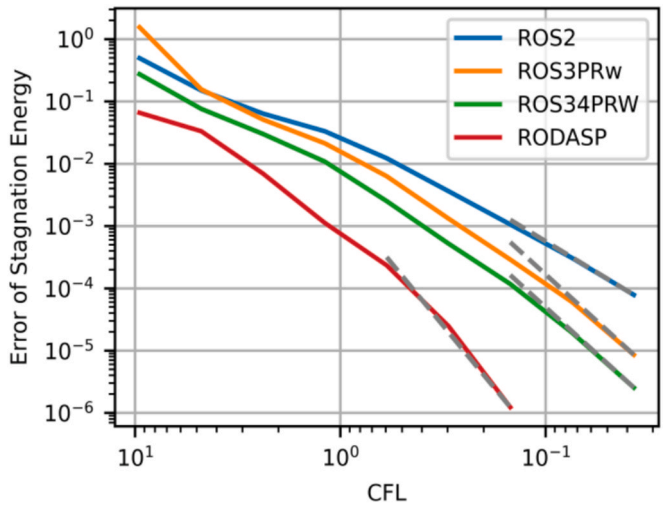


Fig. 1. Convergence behavior of several Rosenbrock schemes. The timestep size is given as CFL $= u_\infty \cdot \Delta t / \Delta x$ with respect to the convection velocity u_∞ of the vortex.

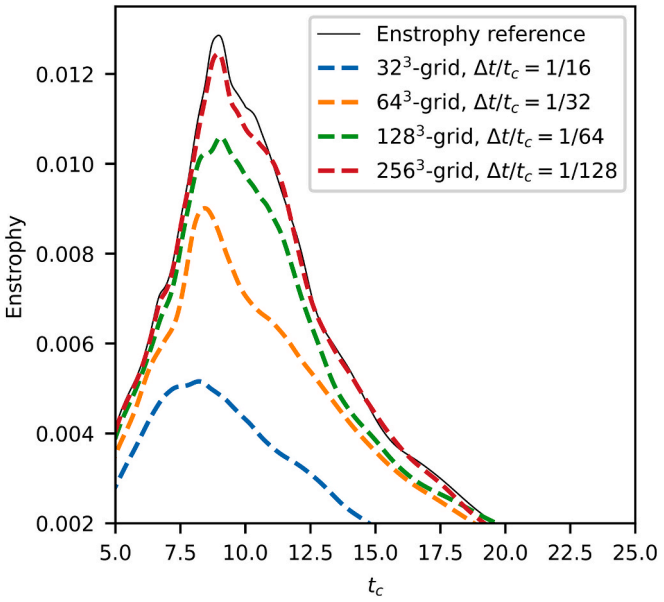


Fig. 2. Spatial convergence of the enstrophy computed with the ROS2 scheme compared to the reference solution. For our setup, we see good agreement with the reference solution when increasing the spatial and temporal resolution. The error due to the spatial discretization appears to dominate the error due to other effects.

preconditioner quality.

4. Results

We will show results over a range of use cases from inviscid and laminar compressible flow to partially resolved and fully modeled turbulent flow, using large eddy simulation (LES) and unsteady Reynolds-averaged Navier-Stokes (URANS) methods.

All results are based on the compressible Navier-Stokes equations assuming the equation of state for an ideal gas. The heat capacity ratio $\gamma = 1.4$ (not to be confused with the effective timestep size coefficient using the same symbol) of dry air at room temperature is used for all cases. For viscous flow, the Sutherland approximation for viscosity is used.

All simulation results have been produced with the CFD software by ONERA, DLR, and Airbus (CODA). CODA is the computational fluid dynamics software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR, and Airbus.

Run times reported were obtained on the DLR-owned high-performance computing cluster CARO featuring compute nodes with two AMD EPYC 7702 processors (128 cores combined) and 256 GB of DDR4 memory running at an effective 3200 MHz.

The best available discretization has been selected per case to ensure qualitatively and quantitatively correct results. It is mentioned for each case individually.

If no other values are given for the case, we use the following default tolerances. For linear methods, we use 10^{-3} relative residual reduction when solving the linear systems for each stage. For nonlinear methods, we use 10^{-3} relative residual reduction of the nonlinear residual in the Newton method and 10^{-1} relative reduction for the linear system in each Newton iteration.

4.1. Inviscid vortex transport

In order to verify our implementation of the Rosenbrock schemes, we first consider a simple test case due to Kok (2009). This test case consists of a vortex that is advected in a periodic channel for which a curvilinear grid is used. We use a Mach number of $Ma = 0.5$ and a second-order FV discretization with an energy-conserving central convection scheme that is also described by Kok (2009) with no turbulence model.

By using a reference solution obtained with an explicit RK4 scheme, we are able to isolate errors due to the time-integration scheme. The resulting convergence behavior for several Rosenbrock schemes is shown in Fig. 1. Good agreement with the theoretical order of convergence is observed for all schemes.

4.2. Taylor Green vortex

The Taylor-Green Vortex (TGV) at a Reynolds number of $Re = 1600$ is a test case that is frequently used for validation of CFD codes. It was featured in the Third Workshop on High-Order CFD Methods (Mastellone et al., 2015) which provided an exact problem description along with a reference solution using a quasi-spectral discretization. In order to be able to compare our results with the reference solution provided for this workshop, we reproduce exactly the same setup in our tests.

The TGV is a 3D direct numerical simulation (DNS) of a decaying vortex. The domain is cubical and features periodic boundary conditions in all directions. In the compressible case, the Mach number is set to $Ma = 0.1$. Furthermore, the ideal gas law is assumed and the bulk viscosity is set to zero. The initial density is constant. Time is measured using the characteristic time scale $t_c = L/V$ with domain size L and characteristic velocity V and the simulation is performed for a duration of $20t_c$. For a complete description of the test case, we refer to Mastellone et al. (2015).

The main quantity of interest for this test case is the integral of the squared vorticity ω over the domain Ω weighted with the density ρ, known as the enstrophy

$$\epsilon = \frac{1}{\rho_0 \Omega} \int_\Omega \rho \frac{\omega \cdot \omega}{2} d\Omega. \tag{11}$$

For an incompressible flow, the enstrophy is proportional to the domain integral, save for the discretization errors introduced by computing $\omega = \nabla \times v$. This relation approximately holds for the nearly incompressible case in the TGV. The enstrophy is therefore a useful diagnostic quantity to assess the accuracy of a given discretization (Mastellone et al., 2015).

We again use the second-order finite volume method with a low-dissipation central convection scheme from Section 4.1 for this test case. To evaluate its accuracy for the TGV test case, we compare the time

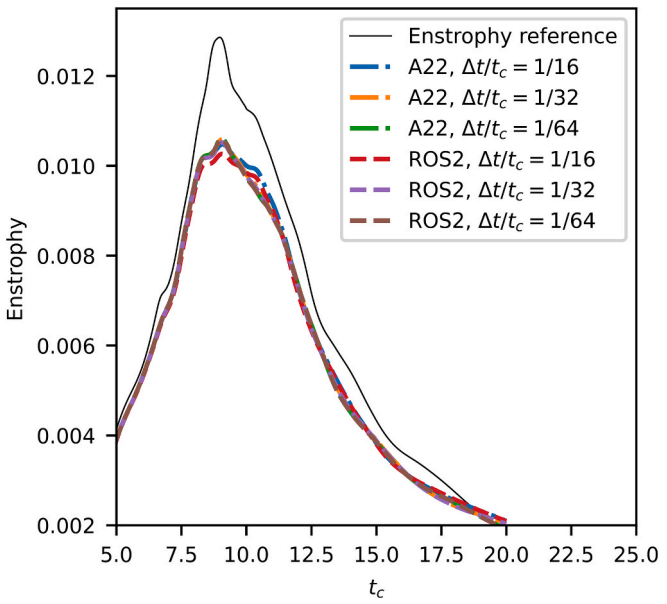


Fig. 3. Comparison of the enstrophy for a grid with 128^3 elements and varying timestep sizes used for the ROS2 and A22 schemes. For large time steps, the enstrophy shows some dependency on the timestep size for both DIRK and Rosenbrock schemes.

Table 2
Runtimes for some Rosenbrock and DIRK schemes, as well as the classical explicit fourth order Runge-Kutta scheme for the TGV case with different grid resolutions.

grid size	#HPC-nodes	runtime in seconds LMP	ROS2	MP	A22	RK4
32^3	1	7	7	27	33	4
64^3	1	78	88	263	353	104
128^3	1	1082	1185	4168	5473	1592
256^3	8	2016	2123	–	–	3329
512^3	64	4393	4536	–	–	–

evolution of the enstrophy against a pseudo-spectral reference solution (van Rees et al., 2011). This is shown in Fig. 2. We observe good agreement with the reference solution for higher grid resolutions. We conclude that our spatial discretization is well-suited for this test case, since the dissipation rate converges to the reference when the spatial resolution is increased. We therefore proceed to evaluate the impact of the time discretization for a fixed spatial resolution.

To study differences in accuracy of Rosenbrock schemes compared to DIRK schemes in this test case, we restrict ourselves to the second-order A22 SDIRK scheme and the ROS2 Rosenbrock scheme. These two time integration schemes are closely related: the ROS2 scheme can be viewed as a linearized version of the A22 scheme (see Tab. 1).

Due to their similarities, we expect the ROS2 and A22 schemes to perform comparably in terms of accuracy and robustness. We test this assumption by observing the enstrophy of both schemes for different timestep sizes on a fixed grid with 128^3 hexahedral elements. This is shown in Fig. 3. For larger time steps, small time discretization errors can be observed for both schemes. For smaller time steps, both schemes yield virtually identical results. The A22 scheme allowed for a slightly larger timestep size. Compared to the spatial discretization error, the error from time integration is small, even for the largest shown timestep size.

In addition to the two-stage second-order schemes from the previous section, we also included two single-stage second-order methods in our performance tests: the implicit midpoint SDIRK scheme MP and its linearized Rosenbrock counterpart, the LMP scheme. The performance results can be seen in Tab. 2. The timestep size has been chosen proportional to the grid resolution in order to maintain accuracy. All computations were performed on the DLR compute cluster CARO.

Our tests show that in our implementation, the performance of Rosenbrock schemes is much better than the performance of comparable SDIRK schemes. The Rosenbrock schemes were also able to outperform the explicit Runge-Kutta scheme for this test case.

Looking at the ROS2 and LMP schemes specifically, it can be seen that using the single-stage Rosenbrock scheme yields little advantage compared to the two-stage Rosenbrock scheme: the advantage of fewer stages is offset by the need for smaller time steps. Additionally, we observed relatively large time-discretization errors with the LMP scheme, which we therefore do not consider competitive compared to the ROS2 scheme. In contrast, using the implicit midpoint rule showed some performance benefits over the A22 scheme.

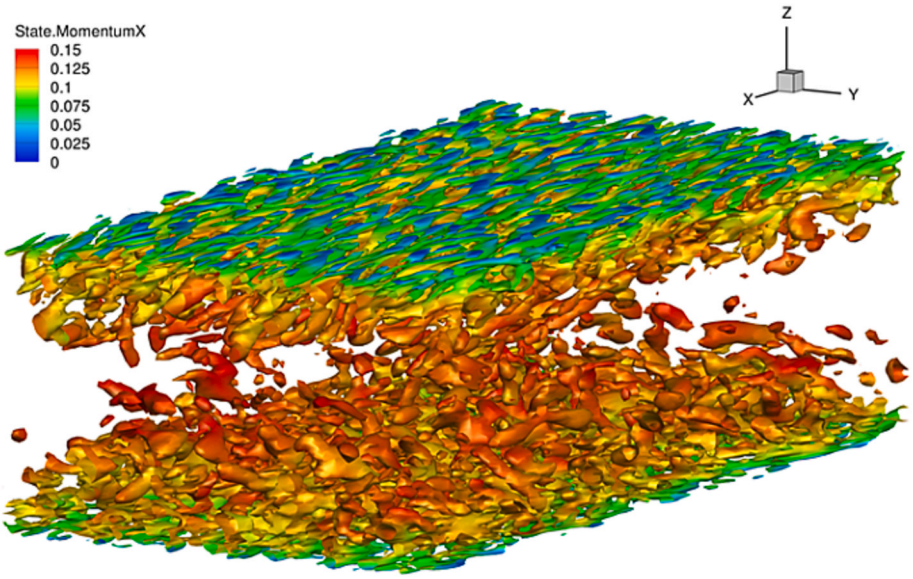


Fig. 4. Iso-surface of Q-criterion coloured by x-component of momentum for the instantaneous flow field of fully developed turbulent channel flow at $Re_\tau = 395$ used for restarts.

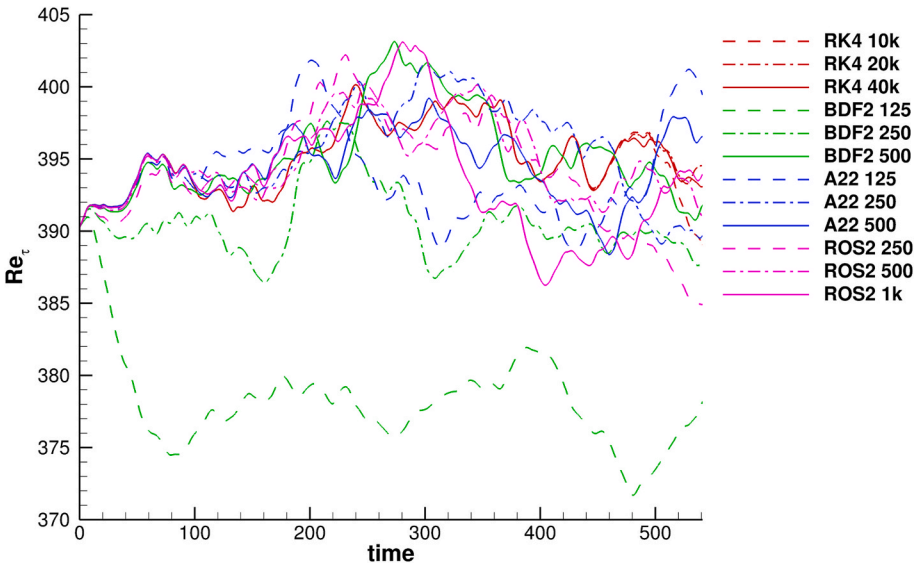


Fig. 5. Evolution of Re_τ from fully developed initial state with different time integration methods and number of time steps per CTU.

4.3. Turbulent channel flow at $Re_\tau = 395$

Moving on from the direct numerical simulation of the TGV with prescribed initial state and synthetic turbulence decay, we now consider natural self-sustaining turbulence in wall-bounded turbulent channel flow at $Re_\tau = 395$. Almost incompressible flow at Mach number $Ma = 0.1$ is considered.

The geometry dimensions are $L \times W \times H = 6.4 \times 3.2 \times 2$ and periodicity is assumed in stream- and spanwise directions. The top and bottom walls use an adiabatic viscous wall boundary condition. A dynamically adjusted artificial pressure gradient term is used to counteract friction at the walls and to ensure that the bulk-velocity-based Reynolds number $Re_{bulk} = 6881.75$, as extracted from DNS data provided by Moser et al. (1999), is maintained. This term is also considered in the energy equation to ensure correct long-term energy balance.

The purely hexahedral simulation grid has a resolution of $80 \times 144 \times 80$ with equidistant spacing in stream- and spanwise directions. A wall-normal distribution with $y^+(1) \leq 1$ and a constant expansion factor of 1.1 is used. We again use the energy-conserving skew-symmetric convection scheme introduced by Kok (2009) without any artificial dissipation. The WALE subgrid-scale model (see Nicoud and Ducros, 1999) is used with the standard value for the coefficient $C_m = 0.325$. Altogether this setup is a well-resolved LES that provides quantitatively excellent results on the given grid for the small Reynolds number, comparing well with results from Probst et al. (2016). The time- and space-averaged friction-velocity Reynolds number Re_τ is considered as an accuracy indicator. With sufficient temporal accuracy and averaged over 10 convective time units (CTUs), the friction Reynolds number Re_τ lies within 0.5% of DNS reference values.

Starting from an initial state with a randomly perturbed constant velocity profile, 20 CTUs are simulated with the highly accurate explicit RK4 scheme. All other simulations are started from this statistically fully developed state, visualized in Fig. 4, and run for 10 CTUs.

We start with an investigation of the accuracy, based on Fig. 5. The accuracy of the explicit RK4 scheme with 10^4 or more time steps per CTU is enough to have virtually identical flow for almost the entire benchmark duration of 10 CTUs. A larger timestep size is not permitted due to the stability limit of explicit schemes. Implicit methods with suitable accuracy lose correlation earlier, after around 1 to 2 CTUs, with no relevant effect on the statistical quantities.

The BDF2 scheme with 250 steps per CTU shows first signs of excessive dissipation already before one CTU, while the BDF2 scheme with only 125 steps per CTU and the first-order implicit Euler (IE) scheme for all tested step sizes (not shown) give inaccurate results.

Table 3
Runtime and averaged friction Reynolds number Re_τ for turbulent channel flow for various time integration methods using different number of time steps per CTU.

method	steps/CTU	runtimein seconds	avg.Re_τ	error vs. RK4/40 k
RK4	40,000	9195	395.2	
	20,000	4808	395.2	+ 0.00%
	10,000	2368	395.0	−0.05%
BDF2	500	2560	395.3	+ 0.03%
	250	2000	390.6	−1.18%
	125	1610	378.2	−4.49%
A22	500	2744	394.6	−0.15%
	250	2083	395.5	+ 0.08%
	125	1791	394.0	−0.30%
ROS2	1000	1632	394.9	−0.08%
	500	1442	393.7	−0.38%
	250	1604	393.3	−0.48%
	125	1639	394.2	−0.25%

Tab. 3 gives some performance numbers and averaged Re_τ for an interval of 10 CTUs. No preconditioning was used in the linear solver for implicit methods, because for this relatively small Reynolds number and the time step sizes used, the stiffness of the equations was moderate. The RK4 scheme with constant cost per time step shows almost perfect scaling with the number of time steps but is restricted in timestep size long before loss of accuracy becomes a concern.

For the A22 SDIRK method, the additional overhead from nonlinear iterations shifts the sweet spot to a timestep size where accuracy loss becomes relevant.

Rosenbrock methods come with an overhead compared to ERK methods, but by allowing larger timestep sizes they quickly outperform ERK methods. An unfavorable timestep size dependency of the linear solver iteration count impacts scaling. At some point, increasing the timestep size even increases the total simulation time. So, there is a sweet spot with low sensitivity. This optimum can be shifted towards larger timestep sizes by adding preconditioning to the linear solver or using a stronger preconditioner.

The overall fastest scheme is ROS2 with 500 steps per CTU and 1442 s, providing slightly improved runtime on top of higher accuracy compared to the A22 scheme with 125 steps per CTU and 1791 s. The explicit RK4 method is significantly slower with 2400 s and excessive accuracy, but already operates close to the stability limit. It benefits

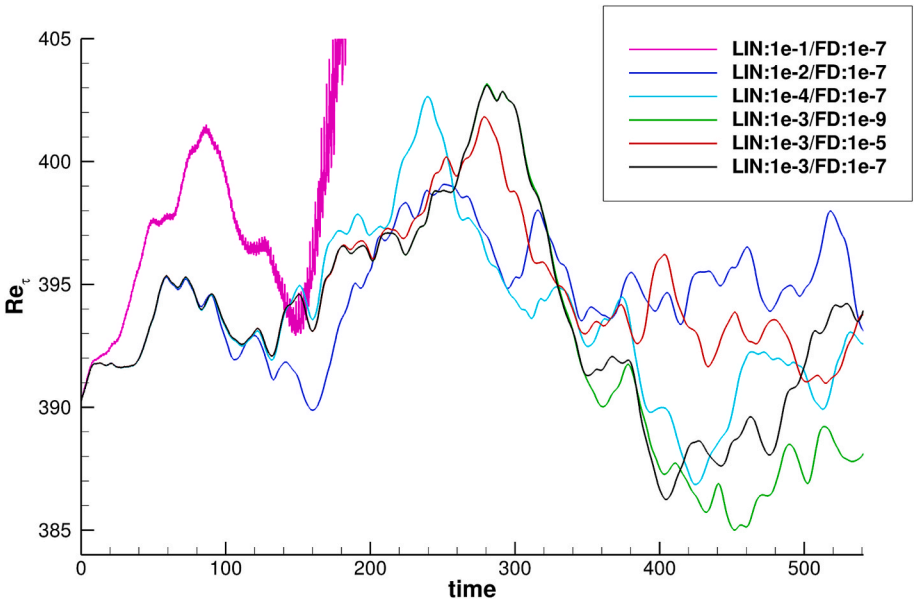


Fig. 6. Sensitivity of ROS2 results for 500 steps per CTU with respect to the linear solver tolerance and the FD step size parameter in the Jacobian approximation.

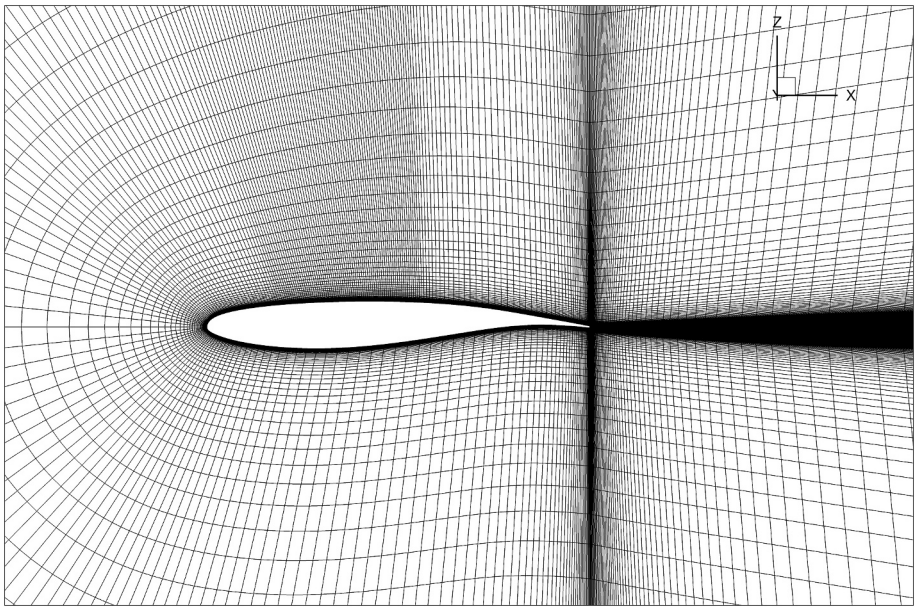


Fig. 7. Grid in the vicinity of the OAT15A airfoil.

from the relatively small aspect ratio due to the small Reynolds number. Only a few small grid cells could force the ERKs to use much smaller timestep sizes.

We also use this test case to assess the sensitivity of the results with respect to the linear solver tolerance and the FD step size parameter in the Jacobian approximation. Results are shown in Fig. 6. We recommend and use a linear residual reduction by 10^{-3} and a step size factor of $\epsilon = 10^{-7}$. Using instead only one order of linear residual reduction leads to an unstable simulation. For two orders of linear residual reduction, the simulation gives acceptable results, but the introduced error is larger than the error of the time integration method itself (compare Fig. 5). When comparing three and four orders of residual reduction, we see differences smaller than the discretization error, which justifies stopping after three orders of residual reduction.

The sensitivity with respect to ϵ is much less pronounced. A value of $\epsilon = 10^{-5}$ already leads to virtually identical results for almost 4 CTUs. Since the choice of ϵ has basically no effect on runtime, we use the theoretically justified value of $\epsilon = 10^{-7}$.

4.4. Shock buffet for OAT15A airfoil

The next case considered is unsteady shock buffet for a supercritical airfoil (OAT15A) at Mach number $Ma = 0.73$ and Reynolds number $Re = 3 \times 10^6$ (Brunet, 2003). The airfoil features a thickness-to-chord ratio of 12.3% and a thick trailing edge of 0.5% of the chord length.

The simulation grid, depicted in Fig. 7, uses anisotropic boundary layer adaptation with $y^+(1) \approx 0.5$ and up to 50 layers within the boundary layer. The far-field boundary is located at a distance of 50 chord lengths. The region of shock travel has increased resolution and the resolution towards the blunt trailing edge is also significantly increased, which turned out to be crucial for correct buffet prediction. The number of elements in the grid is 48 k. For spatial discretization we

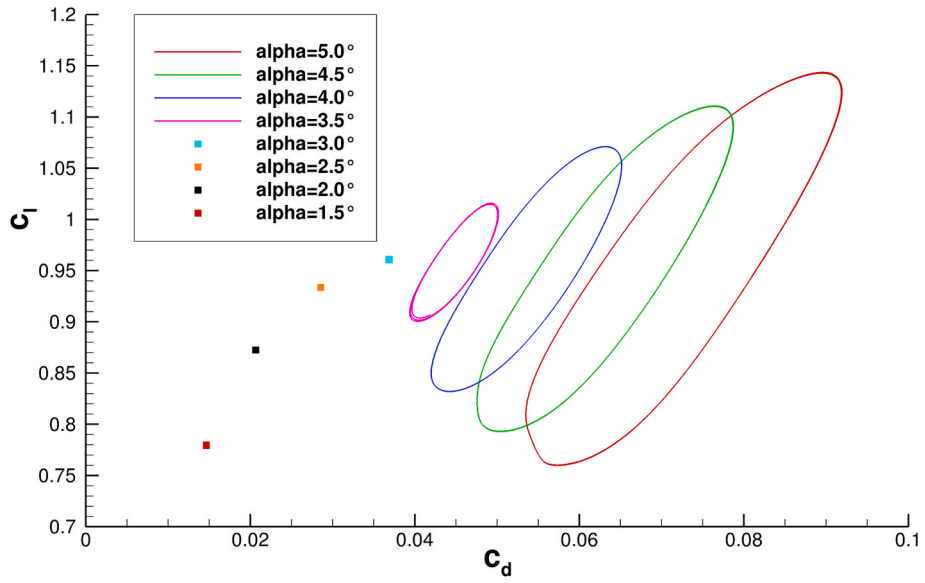


Fig. 8. Phase plot of lift and drag coefficients show steady flow for angle of attack $\alpha \leq 3^{\circ}$ and periodic oscillations for $\alpha \geq 3.5^{\circ}$.

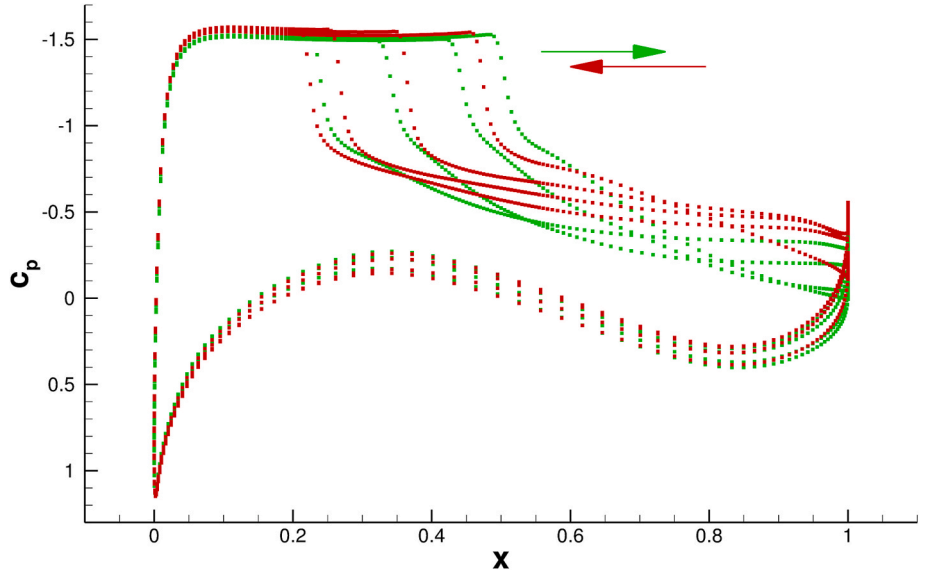


Fig. 9. Pressure coefficient cycle for 5° angle of attack.

use a second-order FV method with linear reconstruction and Roe upwinding. The negative Spalart-Allmaras one-equation linear eddy-viscosity model (SA-neg) (Allmaras et al., 2012) is used as a turbulence model.

Overall, this case is very sensitive to the spatial discretization and modeling parameters. There is also some uncertainty in the numerical setup that uses a distant far-field boundary condition compared to the experimental setup in a wind tunnel. The lift and drag oscillation amplitudes and frequency (see Fig. 8) and the shock motion range (see Fig. 9) are in reasonable agreement with other numerical results and the experiment (Brunet, 2003).

Table 4
Simulation time for a single shock cycle with different time integration schemes and timestep sizes for the oat15a airfoil at $\alpha = 4^{\circ}$.

method	$\Delta t/t_c$	Runtime (min)
A22	0.04	7.94
	0.08	5.78
	0.12	4.63
ROS2	0.005	6.83
	0.01	4.8
	0.02	3.53
	0.04	2.67

The largest possible timestep size for explicit methods is on the order of 10^{-5} which agrees well with an a priori estimate based on the Courant–Friedrichs–Lewy condition using just the shortest edge length and the speed of sound. This is three to four orders of magnitude smaller than the timestep sizes used for implicit methods.

Results from the ROS2 scheme with non-dimensional convective time step size $\Delta t/t_c = 0.005$, with the convective time scale t_c computed from chord length and freestream velocity, are used to assess the general solution quality. Fig. 8 shows the lift coefficient c_l over drag coefficient c_d for several angles of attack with shock buffet onset between $\alpha = 3^{\circ}$ and $\alpha = 3.5^{\circ}$. Over a single cycle the separation bubbles at the shock and the trailing edge grow and merge during upstream movement of the shock. During the downstream movement of the shock, the separation at

the trailing edge completely collapses for a short duration.

For this case, solving the linear systems for implicit schemes efficiently always required the above-mentioned lines preconditioning to overcome the stiffness from anisotropic boundary layer resolution. The preconditioner was updated every 10 physical time steps. Only for unreasonably small timestep sizes $\Delta t/t_c \leq 10^{-3}$ it was viable to use the GMRES method with no preconditioning at all.

No significant differences in the flow solution could be observed for the time step sizes presented in Tab. 4. For the FV method used, the accuracy of the second-order SDIRK and Rosenbrock methods is sufficient, up to the stability limit of the schemes. The only relevant difference between the schemes then is the runtime, given in Tab. 4. At the same timestep size, the Rosenbrock method is more than two times faster than the SDIRK method. The SDIRK scheme A22 became unstable for $\Delta t/t_c = 0.16$, while the ROS2 scheme already became unstable for $\Delta t/t_c = 0.08$. Nevertheless, this again did not translate into a performance advantage for the SDIRK scheme, because of reduced rates of convergence for larger timestep sizes.

5. Conclusion

We showed that Rosenbrock methods can be implemented and operated with little overhead compared to ERK methods. All required components can be implemented with functions commonly available in (legacy) explicit CFD codes. Even such a bare-bones implementation can run cases, where explicit time integration is not feasible.

Rosenbrock methods show a moderate but consistent performance advantage over SDIRK methods. In the presence of strong nonlinearity of the equations, the SDIRK schemes remain stable for larger timestep sizes. However, this does not translate into increased performance when iterative linear solvers are used, because the resulting systems of linear equations become very stiff. Perhaps somewhat counterintuitive, the best performance is not achieved at the largest possible timestep size. Furthermore, Rosenbrock methods are also easier to use from the user's perspective, since there is one less layer of iterative solvers.

Further investigations with W-type Rosenbrock methods, which permit the usage of approximate Jacobians, might provide additional performance improvements by reducing the implicitness of the method to the necessary minimum.

CRediT authorship contribution statement

Johannes Löwe: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Anton Schmid:** Writing – review & editing, Writing – original draft, Visualization, Software, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARO is partially funded by "Ministry of Science and Culture of Lower Saxony" and "Federal Ministry for Economic Affairs and Climate Action".

Data availability

Data will be made available on request.

References

Alexander, Roger (1977). Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s. *SIAM Journal on Numerical Analysis*, 14(6), 1006–10210.1137.

Allmaras, Steven R. & Johnson, Forrester T. & Spalart, Philippe R. (2012). Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. *Seventh International Conference on Computational Fluid Dynamics* (ICCFD7). 1-11.

Baker, A. & Iannelli, G. S. (2012), A stiffly-stable implicit Runge-Kutta algorithm for CFD applications, *26th Aerospace Sciences Meeting*, published online: 17 Aug 2012. 10.2514/6.1988-416.

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. Comput. Fluids 118, 305–320. https://doi.org/10.1016/j.compfluid.2015.06.007.

Brunet, V., 2003. Computational Study of Buffet Phenomenon with Unsteady RANS Equations. In: *21st AIAA Applied Aerodynamics Conference*. https://doi.org/10.2514/6.2003-3679.

Crouzeix, M., 1976. In: Sur Les Methodes De Runge Kutta Pour L'approximation Des Problemes D'evolution. *Computing Methods in Applied Sciences and Engineering*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-85972-4_12.

Nicoud, F., Ducros, F., 1999. Subgrid-Scale stress Modelling based on the Square of the Velocity Gradient Tensor. Flow Turbul. Combust. 62, 183–200. https://doi.org/10.1023/A:1009995426001.

Hairer, E., Wanner, G., 1996. Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Springer Ser. Comput. Math. 14. https://doi.org/10.1007/978-3-662-09947-6.

Kok, J.C., 2009. A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids. J. Comput. Phys. 228 (18), 6811–6832. https://doi.org/10.1016/j.jcp.2009.06.015.

J. Lang Rosenbrock-Wanner Methods: Construction and Mission, Springer International Publishing 2021 10.1007/978-3-030-76810-2_1 1 17.

Mastellone, A. & Capuano, F. & Di Benedetto, S. & Cutrone, L. (2015) Direct Numerical Simulation of the Taylor-Green Vortex at Re = 1600, *Third International Workshop on High-Order CFD,* https://www1.grc.nasa.gov/wp-content/uploads/C3.3_CIRA.pdf (retrieved 27 November 2025).

Moser, Robert D. & Kim, John & Mansour, Nagi N. (1999), Direct numerical simulation of turbulent channel flow up to . *Physics of Fluids*, 1 April 1999; 11 (4): 943–945. 10.1063/1.869966.

Peltzer, Patrick & Lotz, Johannes & Naumann, Uwe (2020), Eigen-AD: Algorithmic Differentiation of the Eigen Library. *Computational Science - ICCS 2020*: 20th International Conference, Amsterdam, The Netherlands, June 3-5, 2020, Proceedings, Part I, 12137, 690-704. 10.48550/arXiv.1911.12604.

Probst, Axel & Löwe, Johannes & Reuß, Silvia & Knopp, Tobias & Kessler, Roland (2016) Scale-Resolving Simulations with a Low-Dissipation Low-Dispersion Second-Order Scheme for Unstructured Flow Solvers. *AIAA Journal*, 54 (10), Seiten 2972-2987. American Institute of Aeronautics and Astronautics (AIAA). 10.2514/1.J054957.

Rang, J., 2013. A New Stiffly Accurate Rosenbrock-Wanner Method for solving the Incompressible Navier-Stokes Equations. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. 120. https://doi.org/10.1007/978-3-642-33221-0_18.

Saad, Y., 2003. *Iterative Methods for Sparse Linear Systems*, SIAM. Other Titles in Applied Mathematics. DOI 10 (1137/1), 9780898718003.

Shu, C.-W., Stanley, O., 1988. Efficient implementation of essentially nonoscillatory shock-capturing schemes. J. Comput. Phys. 77 (2), 439–471. https://doi.org/10.1016/0021-9991(88)90177-5.

van Rees, W.M., Leonard, A., Pullin, D.I., Koumoutsakos, P., 2011. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. J. Comput. Phys. 230 (8), 2794–2805. https://doi.org/10.1016/j.jcp.2010.11.031.