On Rosenbrock and ESDIRK methods for unsteady compressible flows

D. S. Blom\textsuperscript{a}, P. Birken\textsuperscript{bc}, H. Bijl\textsuperscript{a}, F. Kessels\textsuperscript{a}, A. Meister\textsuperscript{c} and A. H. van Zuijlen\textsuperscript{a}

\textsuperscript{a}Faculty of Aerospace Engineering, Delft University, P.O. Box 5058, 2600 GB Delft, The Netherlands
\textsuperscript{b}Centre for the Mathematical Sciences, Numerical Analysis, Lunds University, Box 118, 22100 Lund, Sweden, E-mail: philipp.birken@na.lu.se
\textsuperscript{c}Institute of Mathematics, University of Kassel, Heinrich-Plett-Str. 40, 34132 Kassel, Germany

Abstract

In this article, the efficiency of time adaptive Rosenbrock and ESDIRK methods is compared for finite volume discretizations for compressible unsteady viscous flows. To make the comparison fair, first an efficient solver for both classes in the JFNK context is identified, where the tolerance TOL that steers time adaptivity is the main input parameter. To specify the tolerances in the sub solver, we suggest a simple strategy for the Rosenbrock methods of using TOL/100 that is a good compromise between stability and computational effort. It turns out that the Rosenbrock methods, in particular RODASP, outperform the ESDIRK methods.

1 Introduction

In many engineering and scientific problems, unsteady fluid dynamics play a key role in understanding physics and dynamic interactions. On the fundamental side much research is focused on e.g. inherently unsteady flow, such as turbulence. On the applied side the turbulence may be modeled using LES or RANS approaches and the unsteadiness can be introduced for only the large scale structures, through dynamical interaction with e.g. deforming structures, or by transient processes. Simulation of such unsteady phenomena can become computationally (extremely) expensive due to the
vast amount of time steps that need to be taken. An efficient (low computational time for a given accuracy) time integration scheme is therefore of the utmost importance [2].

We consider the aspect of efficient time integration for unsteady compressible viscous flow problems. There, typically strong boundary layers are present, which make an implicit time integration necessary. Especially the high aspect ratio cells introduced in the boundary layer contribute to an increased stiffness of the problem. Here, singly diagonal implicit Runge-Kutta (SDIRK) methods have proved to be efficient. The use of these schemes in the context of compressible Navier-Stokes equations was analyzed in [1] where they were demonstrated to be more efficient than BDF methods for engineering accuracies. They suggested the use of the six stage method ESDIRK4 of fourth order with an embedded method of third order and the four stage method ESDIRK3 of third order with an embedded method of second order, both designed in [10]. This result about the comparative efficiency of BDF-2 and ESDIRK4 was later confirmed by [9] and [22] for a DG discretization of unsteady Euler flow.

An alternative to implicit Runge-Kutta schemes and BDF schemes that has not been considered much for compressible flows are Rosenbrock schemes, which are also referred to as linearly implicit or semi-implicit [20]. The idea is to linearize an $s$-stage DIRK scheme, thus sacrificing some stability properties as well as accuracy, but hopefully reducing the computational effort in that per time step, $s$ linear equation systems with the same system matrix and different right hand sides have to be solved. However, the comparison between Rosenbrock and DIRK schemes is nontrivial, since solving linear systems is not necessarily faster than solving nonlinear systems and combined with time adaptivity things become very intricate.

Here, we compare time adaptive ESDIRK and Rosenbrock methods in the context of finite volume discretizations of the compressible Navier-Stokes equations based on their work error ratio for realistic problems. Obviously, the amount of work strongly depends on both the solver and the test cases chosen. To obtain a reasonable method, we thus have to carefully choose the solver, as well as the test cases. Previously, these schemes have been considered by St.-Cyr et. al. in the context of discontinuous Galerkin methods [19], as well as in [8] for the incompressible Navier-Stokes equations. In the latter paper, a large number of different SDIRK and Rosenbrock methods is compared to each other in a time adaptive setting. However, the nonlinear systems are solved using direct solvers.

Regarding solvers, we consider preconditioned Jacobian-free Newton-GMRES schemes with a smart choice of the tolerances in all iterations to
be superior to current multigrid methods [3]. In Rosenbrock schemes, there is no Newton scheme, but the implementation can nevertheless be done in a Jacobian-free manner exactly as before. Furthermore, it has been justified in a series of papers on so called Krylov-ROW methods [14, 25, 24] that solving the linear systems inexactly can be done in Rosenbrock methods without loss of order. However, it is not clear how to choose the tolerance for GMRES and thus we will develop a strategy here.

For the iterative solution strategy using a GMRES approach, a good preconditioner is essential for obtaining a high computational efficiency, especially when solving stiff, badly conditioned system, e.g. as a result of high aspect ratio cells in the boundary layer. The preconditioner is always a trade off between accuracy (effectiveness) of the preconditioner and computational effort to build the preconditioner. In this respect the Rosenbrock schemes are expected to have an additional benefit over ESDIRK schemes as the linear system to solve for Rosenbrock schemes is constant for all stages within a time step, whereas for the ESDIRK schemes the linear system to solve is updated every stage and even every Newton step. In this paper we therefore also investigate the effect of the preconditioner on the required number of GMRES iterations with increased condition numbers for the system to solve.

As for realistic test cases, it is important to note that for complex 3D flows we have grids with extreme aspect ratios and that furthermore, we do not know the error. Thus, we will first work with problems where we have an exact or reference solution. In particular, we consider a 2D nonlinear convection-diffusion problem with variable non-linearity and grid stretching to demonstrate how the schemes react to changes in these.

Finally, we move to viscous flow problems and use two different codes and several test cases. To obtain a fair comparison here, we use both reference solutions and the concept of tolerance scaling [17] to obtain a reasonable relation between the tolerance in the time adaptive scheme and the error.

The paper is organized as follows. Section 2 discusses the time integration schemes and time adaptivity. The methods used to solve nonlinear and linear systems of equations are the subject of section 3. We then present results for the nonlinear convection-diffusion equation and Navier-Stokes simulations in sections 4 and 5.

2 Time integration

Here, we use the method of lines paradigm, where a partial differential equation is first discretized in space and then in time. When we restrict
ourselves to autonomous problems, we obtain an initial value problem of the form
\[
\frac{d}{dt} u(t) = f(u(t)), \quad u(0) = u^0, \quad t \in [t_0, t_{\text{end}}],
\]
where \( u \in \mathbb{R}^m \) and \( f : \mathbb{R}^m \rightarrow \mathbb{R}^m \). As time integration schemes, we consider ESDIRK and Rosenbrock schemes.

### 2.1 ESDIRK schemes

An ESDIRK scheme with \( s \) stages is of the form

\[
k_i = f(u^n + \Delta t \sum_{j=1}^{i} a_{ij} k_j), \quad i = 1, \ldots, s \tag{2}
\]

\[
u^{n+1} = u^n + \Delta t \sum_{i=1}^{s} b_i k_i.
\]

and the Butcher array is illustrated in table 1, wherein the diagonal coefficient \( a_{ii} = \alpha \) \((i > 1)\) is constant, which is a property of ESDIRK schemes.

| \( c_i \) | \( a_{11} \) | \( a_{21} \) | \( \alpha \) | \( 0 \) | \( 0 \) |
|-----|------|-----|-----|-----|
| \( c_2 \) | \( a_{21} \) | \( \alpha \) | \( 0 \) | \( 0 \) |
| \( \vdots \) | \( \vdots \) | \( \ddots \) | \( \ddots \) | \( 0 \) |
| \( c_s \) | \( a_{s1} \) | \( \ldots \) | \( a_{s-1} \) | \( \alpha \) |
| \( a_{s1} \) | \( \ldots \) | \( a_{s-1} \) | \( \alpha \) |

**Table 1:** Butcher array of an ESDIRK method

Thus, \( s - 1 \) nonlinear equation systems have to be solved. The point about DIRK schemes is that the computation of the stage vectors is decoupled and instead of solving one nonlinear system with \( sm \) unknowns, the \( s \) nonlinear systems (2) with \( m \) unknowns have to be solved. This corresponds to the sequential application of several implicit Euler steps. With the starting vectors

\[
s_i = u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j, \tag{3}
\]

we can solve the equation

\[
U_i = s_i + \Delta t a_{ii} f(U_i). \tag{4}
\]
In the autonomous case, the equation (4) corresponds to one step of the implicit Euler method with starting vector \(s_i\) and time step \(a_{ii} \Delta t\). The stage derivative \(k_i\) is then obtained via

\[
k_i = \left( U_i - s_i \right) / (a_{ii} \Delta t),
\]

which avoids a costly and for stiff problems error prone evaluation of the right hand side [15].

### 2.2 Rosenbrock schemes

To circumvent the solution of nonlinear equation systems, Rosenbrock methods, also called Rosenbrock-Wanner, ROW or linearly implicit methods, can be used. The idea is to linearize a DIRK scheme, thus sacrificing some stability properties, as well as accuracy, but obtaining a method that has to solve \(s\) linear equation systems with the same system matrix and different right hand sides per time step.

To derive these schemes, we start by linearizing formula (2) around \(s_i\) to obtain

\[
k_i \approx f(s_i) + \Delta t a_{ii} \frac{\partial f(s_i)}{\partial u} k_i.
\]

To avoid a re-computation of the Jacobian, we replace \(\frac{\partial f(s_i)}{\partial u}\) by \(J = \frac{\partial f(u^n)}{\partial u}\). Finally, to gain more freedom in the definition of the method, linear combinations of \(\Delta t J k_i\) are added to the last term. Since the linearization procedure can be interpreted as performing just one Newton step at every stage of the DIRK method instead of a Newton loop, the added terms correspond roughly to choosing a specific initial guess for the first Newton iteration. If instead of the exact Jacobian, an approximation \(W \approx J\) is used, we obtain so called W-methods, which have additional order conditions [6].

If the linear system is solved using a Krylov subspace method, the scheme is called a Krylov-ROW method. Generally speaking, Rosenbrock methods are less accurate than SDIRK methods, which will result later in the time step selector choosing smaller time steps for Rosenbrock methods.

We thus obtain an \(s\)-stage Rosenbrock-Wanner method with coefficients \(a_{ij}, \gamma_{ij}\) and \(b_i\) in the form
\[(I - \gamma_{ii} \Delta t W)k_i = f(s_i) + \Delta t W \sum_{j=1}^{i-1} \gamma_{ij} k_j, \quad i = 1, ..., s\]

\[s_i = u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j, \quad i = 1, ..., s \quad (5)\]

\[u^{n+1} = u^n + \Delta t \sum_{i=1}^{s} b_i k_i. \]

Here, the coefficients \(a_{ij}\) and \(b_i\) correspond to those of the DIRK method and the \(\gamma_{ii}\) are the diagonal coefficients of that, whereas the off-diagonal \(\gamma_{ij}\) are additional coefficients. Note that in the case of a non-autonomous equation, an additional term \(\Delta t \gamma_{ii} \partial_t f(t_n, u_n)\) appears on the right hand side of (5).

The efficient implementation of Rosenbrock methods is done using a set of auxiliary variables

\[U_i = \Delta t \sum_{j=1}^{i} \gamma_{ij} k_j\]

and thus circumvents the matrix-vector multiplication in the previous formulation (5). Using the identity

\[k_i = \frac{1}{\Delta t} \left( \frac{1}{\gamma_{ii}} U_i - \sum_{j=1}^{i-1} c_{ij} U_j \right)\]

with coefficients \(c_{ij}\) explained below, we obtain

\[(I - \gamma_{ii} \Delta t W)U_i = \Delta t \gamma_{ii} f(\hat{s}_i) + \gamma_{ii} \sum_{j=1}^{i-1} c_{ij} U_j, \quad (6)\]

\[\hat{s}_i = u^n + \sum_{j=1}^{i-1} \tilde{a}_{ij} U_j,\]

\[u^{n+1} = u^n + \sum_{i=1}^{s} m_i U_i.\]

The relation between the two sets of coefficients is the following:
\[
C = \text{diag}(\gamma_{11}^{-1}, \ldots, \gamma_{ss}^{-1}) - \Gamma^{-1}, \quad \hat{A} = A\Gamma^{-1}, \quad m^T = b^T\Gamma^{-1},
\]

where \(\Gamma = (\gamma_{ij})_{ij}\).

When coding a Rosenbrock method from a given set of coefficients, extra care should be taken, since different authors define the coefficients in slightly different ways, including a different scaling by \(\Delta t\) in the definition of the method. Further possible confusion arises from the two different formulations in (5) and (6). Here, the coefficients for both formulations will always be given.

### 2.3 Time adaptivity

For DIRK and Rosenbrock methods, this is done using the embedded schemes of a lower order \(\hat{p}\). Comparing the local truncation error of both schemes, we obtain an estimate of the local error \(\hat{l}\) of the lower order scheme:

\[
\hat{l} \approx \Delta t \sum_{j=1}^{s} (b_i - \hat{b}_i)k_i,
\]

respectively, for the efficient formulation (6) of the Rosenbrock method

\[
\hat{l} \approx \sum_{j=1}^{s} (m_i - \hat{m}_i)U_i.
\]

The local error estimate is then used to determine the new step size. To do this, we decide beforehand on a target error tolerance, which we implement using a common fixed resolution test [18]. This means that we define the error tolerance per component via

\[
d_i = RTOL|u^n_i| + ATOL.
\]

Typically, we choose \(RTOL = ATOL\), so that there is only one input parameter for this. We then compare (9) to the local error estimate via requiring

\[
\|\hat{l}/d\| \leq 1,
\]

where the \(\cdot\) denotes a point-wise division operator and we use the 2-norm from now on.
The next question is how the time step has to be chosen, such that the error can be controlled. A digital filter is used for the selection of the time step size, as discussed in [18]. The next time step is computed with

\[ \Delta t^{n+1} = \| d^n / l^n \|^{\beta_1} \| d^{n-1} / l^{n-1} \|^{\beta_2} \left( \frac{\Delta t^n}{\Delta t^{n-1}} \right)^{-\alpha_2} \Delta t^n, \] (11)

with \( p \cdot \beta_1 = p \cdot \beta_2 = \alpha_2 = \frac{1}{4} \) and \( p \) being the order of the embedded method of the ESDIRK or ROW-scheme.

The controller needs to be started with the classic controller [6]:

\[ \Delta t^{n+1} = \Delta t^n \| l^n \|^{-1/k}. \] (12)

Step size rejections may be reduced by basing the test on the requested change \( \rho^n \) instead of on the error estimate. Also, discontinuities in the step size change ratio \( \frac{\Delta t^{n+1}}{\Delta t^n} \) are removed by applying a smooth limiter. Thus, the new step size is determined via

\[ \rho^n = \| d^n / l^n \|^{\beta_1} \| d^{n-1} / l^{n-1} \|^{\beta_2} \left( \rho^{n-1} \right)^{-\alpha_2}, \] (13)

and the smooth limiter gives \( \hat{\rho} \) with \( \kappa = 2 \):

\[ \hat{\rho} = 1 + \kappa \arctan \left( \frac{\rho^n - 1}{\kappa} \right). \] (14)

In order to compare the computational efficiency of different time integration schemes, it is desirable to scale and calibrate the tolerance of the adaptive time step size control algorithm. The control algorithm should run in a tolerance proportional mode: when the tolerance is changed by one order of magnitude, then the error of the solution should change by one order of magnitude. Also, the different time integration schemes should deliver the same accuracy for the same tolerance setting [17].

The scaling transformation

\[ TOL' = \beta TOL_0^{\alpha-1} TOL^{1/\alpha} = \tau TOL^{1/\alpha} \] (15)

with \( \tau = \beta TOL_0^{\alpha-1} \), can be used to compare the computational efficiencies of the ESDIRK and Rosenbrock time integration schemes [17]. \( TOL' \) represents the tolerance parameter used by the adaptive step size control algorithm, \( TOL \) is the parameter specified by the user, \( TOL_0 \) is the equivalence point determined during the calibration, \( \beta \) is a constant to equally calibrate the different time integration schemes, and \( \alpha \) is the measured order of the adaptive step size control algorithm of the reference computations for the calibration.
3 Solving nonlinear and linear equation systems

The ESDIRK schemes lead to nonlinear systems of the form

\[ u = \tilde{u} + \alpha \Delta t \hat{f}(u), \tag{16} \]

where \( u \in \mathbb{R}^m \) is the vector of unknowns, \( \alpha \) the diagonal coefficient of the Butcher tableau and \( \tilde{u} \) is a given vector. As before, the boldface denotes a vector of all conservative variables from all cells. Finally, the function \( \hat{f}(u) \) consists of everything else coming from the spatial and temporal discretization.

3.1 Newton methods

In order to solve equation (16), we use Newton’s method. This solves the root problem

\[ F(u) = 0 \tag{17} \]

for a differentiable function \( F(u) \) in the following way:

\[ \frac{\partial F(u)}{\partial u} \bigg|_{u^{(k)}} \Delta u = -F(u^{(k)}) \]

\[ u^{(k+1)} = u^{(k)} + \Delta u, \quad k = 0, 1, \ldots. \tag{18} \]

If the linear equation systems (18) are solved exactly, the method is locally second order convergent. As termination criteria, we always use relative ones. For the residual based indicator we obtain

\[ \|F(u^{(k+1)})\| \leq \epsilon \cdot \|F(u^{(0)})\| \tag{19} \]

Since an exact Jacobian is rarely available and the scheme is costly and unnecessary, other variants approximate terms in (18) and solve the linear systems only approximately. Here, the linear equation systems are solved by an iterative scheme. These are terminated prematurely, based on the residual of the linear equation system. These schemes are called inexact Newton methods and have been analyzed in [4], where the inner solver is terminated if the relative residual is below a certain threshold. This type of scheme can be written as:

\[ \left\| \frac{\partial F(u)}{\partial u} \bigg|_{u^{(k)}} \Delta u + F(u^{(k)}) \right\| \leq \eta_k \|F(u^{(k)})\| \tag{20} \]

\[ u^{(k+1)} = u^{(k)} + \Delta u, \quad k = 0, 1, \ldots. \]
The $\eta_k \in \mathbb{R}$ are called forcing terms. In [5], the choice for this sequence is discussed and it is proved that this scheme converges locally quadratic, if the forcing terms go to zero fast enough in a certain sense. However, it is not necessary to solve the first few linear systems very accurately. This is in line with the intuition, that while we are far away from the solution, we do not need the optimal search direction for Newton’s method, but just a reasonable one, to get us in the generally correct direction. A way of achieving this is the following:

$$
\eta_k^A = \gamma \frac{||F(u^{(k)})||^2}{||F(u^{(k-1)})||^2}
$$

with a parameter $\gamma \in (0, 1]$. This was suggested by [5], where they also prove that this sequence has the convergence behaviour required for the theorem. Thus, the theorem says that if this sequence is bounded away from one uniformly, convergence is quadratic. Therefore, we set $\eta_0 = \eta_{\text{max}}$ for some $\eta_{\text{max}} < 1$ and for $k > 0$:

$$
\eta_k^B = \min(\eta_{\text{max}}, \eta_k^A).
$$

[5] furthermore suggest safeguards to avoid volatile decreases in $\eta_k$. To this end, $\gamma \eta_{k-1}^2 > 0.1$ is used as a condition to determine if $\eta_{k-1}$ is rather large and thus the definition of $\eta_k$ is refined to

$$
\eta_k^C = \begin{cases} 
\eta_{\text{max}}, & n = 0, \\
\min(\eta_{\text{max}}, \eta_k^A), & n > 0, \gamma \eta_{k-1}^2 \leq 0.1 \\
\min(\eta_{\text{max}}, \max(\eta_k^A, \gamma \eta_{k-1}^2)) & n > 0, \gamma \eta_{k-1}^2 > 0.1
\end{cases}
$$

Finally, to avoid over solving in the final stages, Eisenstat and Walker suggest

$$
\eta_k = \min(\eta_{\text{max}}, \max(\eta_k^C, 0.5\epsilon/||F(u^{(k)})||)), \quad (21)
$$

where $\epsilon$ is the tolerance at which the Newton iteration would terminate, see (19).

Furthermore, there is a feedback loop between the nonlinear iterations and the time step size. If the nonlinear iteration does not terminate after 40 iterations, the time step is repeated with $\Delta t = \Delta t/4$, respectively in the Rosenbrock case, it can happen that a result obtained in between or after the time step is nonphysical. If this happens, meaning that the norm of the new function evaluation is NaN, we repeat the time step with $\Delta t_n/4$. 

10
3.2 Jacobian-free GMRES

To solve the linear systems, we use GMRES(m), meaning restarted GMRES. There, the system matrix appears only in matrix vector products. Thus it is possible to formulate a Jacobian free version of Newton’s method [11].

To this end, the matrix vector products \( Ax \) are replaced by a difference quotient via

\[
Ax = \frac{\partial F(u)}{\partial u} x \approx \frac{F(u + \epsilon x) - F(u)}{\epsilon}
\]

This works for the linear systems arising in Newton scheme, but also for those from the Rosenbrock scheme.

If the parameter \( \epsilon \) is chosen very small, the approximation becomes better, however, cancellation errors become a major problem. A simple choice for the parameter that avoids cancellation but still is moderately small is given by [13] as

\[
\epsilon = \frac{\sqrt{\text{eps}}}{\|x\|_2},
\]

where \( \text{eps} \) is the machine accuracy.

The method terminates based on a relative criterion, namely

\[
\|Ax^k - b\|_2 \leq \eta_k \|b\|_2,
\]

where in the Newton case, the \( \eta_k \) are the forcing terms as just described, whereas in the Rosenbrock case, there is no theory available that tells how to best choose the tolerance. We will discuss this in section 5.1.1.

3.3 Preconditioning strategy

Two different preconditioning strategies are employed. For the first strategy, ILU(0) is employed as a preconditioner, and is refactored periodically after 30 time steps. Thereby, the matrix the factorization is based on is the Jacobian corresponding to the first order discretization.

The second preconditioning strategy consists of a measure to automatically determine whether it is preferred to compute a new preconditioner [16]. The preconditioner update strategy is based on the principle that the accuracy of the preconditioner influences the number of iterations of the Krylov subspace solver. The preconditioner is kept frozen while the total computational time spent on Krylov subspace iterations with the current frozen...
preconditioner is less than the computational time needed for an evaluation of the preconditioner.

The preconditioner freeze strategy in [16] is modified in the sense that computational times for rejected time steps are ignored by the algorithm. Also, the number of stages per time step differs per time integration scheme. Therefore, only the computational time needed for the first stage is used to determine whether it is necessary to update the preconditioner.

4 A nonlinear convection-diffusion equation

As a first test case, we consider a generalized nonlinear convection diffusion equation for two purposes: 1) investigate the effect of non-linearity on the accuracy of Rosenbrock schemes versus ESDIRK schemes, and 2) investigate the effect of increased stiffness on the efficiency of the preconditioner. As the Rosenbrock schemes are linearly implicit, it is expected that their accuracy is less compared to ESDIRK schemes when solving a nonlinear problem. On the other hand, the constant stage matrix in Rosenbrock schemes promises a benefit when reusing the preconditioner compared to ESDIRK schemes, especially in the presence of a badly conditioned system matrix. Which of these effects weighs the most and whether increased computational efficiency can be observed for Rosenbrock schemes is investigated for an academic problem.

4.1 Model problem

The governing equation for this problem is given by

\[ u_t = \beta u^k_c \cdot \nabla u + \nabla \cdot (u^k_d \nabla u), \quad x \in \Omega := (0,1) \times (0,1) \]  

(23)

with Dirichlet boundary conditions \( u = 1 \) on the outer boundary \( \partial \Omega \) and initial condition \( u(x, y, 0) = u_0(x, y) \). Here,

\[ \beta = \tilde{\beta} \left( \begin{array}{c} \sin \gamma \\ \cos \gamma \end{array} \right) \]

with \( \tilde{\beta} = 200 \) the magnitude and \( \gamma = 0.35\pi \) the angle of the direction of forced convection. Finally, the coefficients \( k_c, k_d \in \mathbb{N} \) determine the degree of non-linearity. With \( k_c = k_d = 0 \), we obtain the linear convection diffusion equation, with \( k_c = 1, k_d = 0 \) the nonlinear convection diffusion equation often used as a model for the Navier-Stokes equations. For larger values, the strength of the non-linearity increases. As initial data we use the function
that is one everywhere, except on the square $[0.2, 0.3] \times [0.2, 0.3]$, where the initial value is $1+\Delta u$, with the initial jump $\Delta u = 0.1$ for the baseline case, see figure 1. We discretize this problem using finite differences, where we use first order upwind for the convective part and second order central differences for the diffusive part. The computational domain is discretized by $N \times N$ points with a stretching ratio $SR$ to define the amount of stretching in the mesh. In the following test cases the stretching is equal in both $x$ and $y$ direction and clusters the nodes towards the center of the domain (see Fig. 1 for mesh generated with the baseline $SR = 1.1$). The stretching in the mesh is an easy way to increase the condition number of the system matrix for the investigation into preconditioner effectiveness.

The error at the end of the simulation is computed with respect to a temporally exact solution which is obtained with a fifth order ESDIRK scheme and a time step of $\Delta t = 0.002/256$. The $L_2$-norm of the error is normalized by the $L_2$-norm of the difference between the temporally exact solution and the steady state solution (uniform field $u = 1$); for an $L$-stable time integration scheme the expected solution for $\Delta t \to \infty$ is the steady state solution, which is considered an error of 100%.

### 4.2 Effect of non-linearity on accuracy

As a first investigation, the effect of non-linearity in the model problem on the (reduction of) accuracy of Rosenbrock schemes versus ESDIRK schemes is considered. To this end the error of the solution with respect to a temporally exact solution is compared for a range of time steps $\Delta t =$
0.002/2^m, m = 1 \ldots 8. The linear systems are solved directly as we do not consider computational efficiency for this investigation. The non-linearity is varied from none (linear), baseline \( k_c = 1, k_d = 0 \), stronger nonlinear convection \( k_c = 3 \) to stronger non-linearities due to a larger variation in the solution \( u \) by prescribing a larger initial jump \( \Delta u = 0.5 \). It was chosen not to vary the non-linearity of the diffusion term as the ratio of 200 between convection and diffusion coefficients \( \hat{\beta} \) puts more emphasis on convection than diffusion.

The results for the different fixed time step sizes are shown in Fig. 2. In

![Graphs showing normalized L2 error vs. timestep for different schemes and non-linearities.](image)

(a) Linear, \( k_c = 0, k_d = 0 \)  
(b) Baseline, \( k_c = 1, k_d = 0 \)

(c) Increased nonlinear convection, \( k_c = 3 \)  
(d) Increased initial jump \( \Delta u = 0.5 \), \( k_c = 1 \), \( k_d = 0 \)

Figure 2: Accuracy of Rosenbrock and ESDIRK schemes with varying non-linearity.

In the linear case the chosen Rosenbrock and ESDIRK methods reduce to the same scheme and hence provide the same accuracy, therefore any observed differences in accuracy for the nonlinear cases is caused by the difference
in dealing with non-linearities by either Rosenbrock or ESDIRK. For the baseline case, Fig. 2(b), the non-linearity is not that strong and resembles the non-linearity observed in the convection term of the Navier-Stokes equations. The linearization of stages by the Rosenbrock schemes shows a small increase of the error, with the largest increase of about a factor 3 for the fourth order schemes. Increasing the nonlinear convection component to $k_c = 3$ in Fig. 2(c) reduces the accuracy of Rosenbrock schemes compared to ESDIRK schemes even further, especially for the third order schemes. Increasing the non-linearity by increasing the $\Delta u$ jump in the initial condition, Fig. 2(d), shows about the same effect as increasing the nonlinear convection component. Additionally, for the larger time steps instability was observed for the Rosenbrock schemes. This indicates that, although all methods possess L-stability, the nonlinear stability properties of the Rosenbrock schemes are reduced compared to their ESDIRK counterparts.

### 4.3 Effect of mesh stretching on efficiency

In this section we wish to investigate the benefit of the constant stage matrix of the Rosenbrock schemes compared to the ESDIRK schemes when iteratively solving the system using preconditioned GMRES. It is expected that when the system becomes less well conditioned, the effectiveness of the preconditioner starts to play a more prominent role. Since stiffness can be introduced to the system by high aspect ratio cells, often encountered in boundary layers, the mesh stretching is adjusted from $SR = 1.0$ (uniform mesh) to $SR = 1.3$ (mesh with highly stretched cells).

An indication of the mesh properties and resulting condition numbers for the stage matrix $[I - \gamma \Delta t J]$ and preconditioned (ILU(0)) stage matrix are presented in Table 2. The condition numbers are determined for the baseline model $k_c = 1, k_d = 0$, on an $80 \times 80$ grid for the initial solution, the diagonal coefficient for the third order schemes $\gamma \approx 0.436$, and a time step of $\Delta t = 0.001$. Note that for this test case the exact Jacobian is used for both ESDIRK and Rosenbrock. Simulations are run with the iterative solution strategies aligned with the settings used for the more complicated problems: i.e. the ESDIRK schemes use the Eisenstat-Walker update strategy with a tolerance $10^{-6}$ and GMRES tolerance selection. For the Rosenbrock schemes the tolerance for GMRES is set to $10^{-6}$. For this test case we do not use adaptive time stepping as the time scales over the simulation time are not varying much.

For the ESDIRK schemes, two update strategies for the preconditioner have been considered: a) one update at the start of a new time step, b)
Table 2: Grid properties (stretching ratio SR and maximum aspect ratio AR) and stage matrix properties (condition number and preconditioned condition number)

<table>
<thead>
<tr>
<th>SR</th>
<th>max. AR</th>
<th>cond. number</th>
<th>prec. cond. number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1</td>
<td>47</td>
<td>4.7</td>
</tr>
<tr>
<td>1.1</td>
<td>41</td>
<td>11670</td>
<td>216</td>
</tr>
<tr>
<td>1.2</td>
<td>1225</td>
<td>11.6·10^6</td>
<td>1827</td>
</tr>
<tr>
<td>1.3</td>
<td>27784</td>
<td>6.55·10^9</td>
<td>7150</td>
</tr>
</tbody>
</table>

an update for every stage and Newton update. Although the second option results in less GMRES iterations, the amount of CPU time required was increased significantly compared to the first option. We therefore estimate the amount of work as the total number of GMRES iterations required for the simulation using update strategy a) for ESDIRK.

Fixed time step simulations were run as before for a range of time steps $\Delta t = 0.002/2^m, m = 1 \ldots 8$. From work-precision plots the speedup between the work required for Rosenbrock compared to ESDIRK was determined as the ratio between the required number of GMRES iterations for ESDIRK and Rosenbrock. The results of this investigation is shown in Fig. 3. The

Figure 3: Computational speedup in terms of GMRES iterations of Rosenbrock over ESDIRK schemes.

results show that the Rosenbrock schemes particularly have a benefit at lower precision (larger time steps) and higher stretching ratios. Both can be related to the conditioning of the stage matrix as both increase the condi-
tion number. The results support the idea that the preconditioner is more effective when the stage matrix does not change within the time step. For the highest stretching ratio the speedup of Rosenbrock can be as large as a factor of three for the largest times steps (lowest accuracies). For higher accuracies the speedup gradually reduces. For the better conditioned systems (with uniform mesh and stretching ratio of 1.1), the ESDIRK schemes become more efficient for errors below $10^{-2}$. From these results it is concluded that the Rosenbrock schemes are expected to perform well in terms of efficiency in the presence of substantial stiffness in the system and for larger time steps (engineering levels of accuracy). However, as seen in the previous section, Rosenbrock schemes may suffer from instability for large time steps.

5 Numerical results for the Navier-Stokes equations

The Navier-Stokes equations are a second order system of conservation laws (mass, momentum, energy) modelling viscous compressible flow. Written in conservative variables density $\rho$, momentum $\mathbf{m}$ and energy per unit volume $\rho E$:

$$
\begin{align*}
\partial_t \rho + \nabla \cdot \mathbf{m} &= 0, \\
\partial_t m_i + \sum_{j=1}^{d} \partial_{x_j} (m_i v_j + \rho \delta_{ij}) &= \frac{1}{Re} \sum_{j=1}^{d} \partial_{x_j} S_{ij}, \quad i = 1, \ldots, d \\
\partial_t (\rho E) + \nabla \cdot (H \mathbf{m}) &= \frac{1}{Re} \sum_{j=1}^{d} \partial_{x_j} \left( \sum_{i=1}^{d} S_{ij} v_i - \frac{1}{Pr} W_j \right).
\end{align*}
$$

Here, $d$ stands for the number of dimensions, $H$ for the enthalpy per unit mass, $S$ represents the viscous shear stress tensor and $W$ the heat flux. As the equation are dimensionless, the Reynolds number $Re$ and the Prandtl number $Pr$ appear. The equations are closed by the equation of state for the pressure $p = (\gamma - 1)\rho e$, where we assume an ideal gas.

5.1 Two dimensional flow around a cylinder

The first test case is a two dimensional flow around a circular cylinder. The cylinder is held fixed in a uniform inflow, resulting in a vortex-street behind
it. When the initial transient has disappeared, an unsteady periodic flow is present. This test case has been used in [1] and [21] to study the order of the ESDIRK schemes in comparison with BDF2.

The cylinder with diameter $D$ is located in a fixed position in a uniform flow field with Mach number $M_\infty = 0.3$ and Reynolds number $Re_\infty = 1,000$, simulating a laminar flow. The radius of the cylinder is used as the characteristic length to determine the Reynolds number. The flow solver used for this test case is the commercial flow solver Hexstream, which is developed by NUMECA Int. A cell centered finite volume scheme is applied, with a second order central discretization with Jameson type scalar artificial dissipation [7]. The same flow solver is also used for the three dimensional test case discussed in Section 5.3.

The computational domain consists of $2.5D$ upstream of the centre of the cylinder, $4.5D$ above and below the cylinder centre, and $16.5D$ downstream of the centre of the cylinder. The mesh is refined in twelve steps to obtain a highly refined region close to the cylinder and in the wake downstream, resulting in an unstructured mesh with 10,608 cells. Close to the cylinder five extra layers of body conformal cells are generated resulting in an accurate representation of the boundary layer. The smallest cells which are located in the boundary layer, are of size $6.6\cdot10^{-5}D$. The maximum aspect ratio of the cells in the mesh is 6.3, and the minimum aspect ratio is 1.0. Refinement in the wake is performed, since the vortex street needs to be resolved accurately to obtain a good accuracy for the simulations. The generated mesh is shown in Figure 4.

The second preconditioning strategy is employed with the ILU(1) preconditioner. Thus, the preconditioner is kept frozen until the total computational time spent on Krylov subspace iterations with the current frozen preconditioner is less than the computational time needed for an evaluation of the preconditioner. The maximum number of Newton iterations is 40, and the maximal dimension of the Krylov subspace is 50.

5.1.1 Effect of GMRES tolerance on numerical accuracy for Rosenbrock time integration

Figure 5 and 6 show the results for a sequence of computations where the tolerance for the adaptive time step controller is varied for third order and fourth order Rosenbrock time integration schemes. As shown in the figures, it is required to use a more strict tolerance for the GMRES solver compared to the tolerance set for the adaptive time step controller. As shown in Figure 5, when a stricter tolerance setting is used for GMRES, the relative
error will show less irregular behavior. When the tolerance settings for the linear solver is not strict enough, a large number of time steps is rejected resulting in a significantly larger computational costs for the simulation. It is suggested to use the factor $10^{-2}$ in order to have a good compromise between robustness and computational efficiency.

![Figure 4: Computational mesh used for the uniform flow around a circular cylinder case](image1.png)

![Figure 5: Two-dimensional cylinder benchmark: effect of GMRES tolerance on numerical accuracy for Rosenbrock time integration schemes. Tolerance for the adaptive time step controller costs versus relative error is shown.](image2.png)

5.1.2 Effect of tolerance calibration on numerical accuracy

When the time step is selected with the adaptive time step controller, a large difference in accuracy is observed between the applied time integration schemes, compare figure 7. The difference in accuracy is more than one order of magnitude between ESDIRK3 and RODASP4. Relative large computational times are observed for the most inaccurate computations of ESDIRK3 and ESDIRK4. This is caused by the fact that many time steps
are being rejected, which leads to an increase in computational time.

To reduce the large difference in accuracy between the different time integration schemes, the tolerance calibration procedure (15) is now used. The coefficients used are shown in table 3. Thus, the difference in accuracy for a given tolerance between the time integration schemes is essentially eliminated (see figure 8). Furthermore, it can be seen that RODASP outperforms the other time integration schemes for smaller tolerances with ESDIRK4 being second best. There is no noticeable difference between the different schemes for coarse tolerances.
Table 3: Calibration coefficients determined with the two dimensional flow around a cylinder test case

<table>
<thead>
<tr>
<th>Time integration scheme</th>
<th>$\alpha$</th>
<th>$\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESDIRK3</td>
<td>0.86</td>
<td>44.49</td>
</tr>
<tr>
<td>ESDIRK4</td>
<td>0.85</td>
<td>107.73</td>
</tr>
<tr>
<td>ROS34PW2</td>
<td>0.99</td>
<td>125.05</td>
</tr>
<tr>
<td>RODASP</td>
<td>0.96</td>
<td>897.02</td>
</tr>
</tbody>
</table>

Figure 8: Adaptive time stepping with tolerance calibration: accuracy and computational efficiency for different time integration schemes for the two dimensional flow around a cylinder test case
5.2 Two dimensional flow in cooling of flanged shaft

Figure 9: Grid around flanged shaft. Left: Complete computational domain. Right: Zoom on region around lower tube and shaft.

Another two dimensional case is considered, where we use a problem stemming from gas quenching [23]. Here, high pressured air is blown from two tubes at a flanged shaft in a cooling process. The Mach number at the outlets of the tubes is 1.0, the Reynolds number 1,000. The grid is unstructured and has 142,052 triangular cells. It is illustrated in figure 9.

Regarding initial conditions, the initial velocity is zero, the density 1.2 and the temperature of the gas is 289 K. We compute five seconds of real time using the code TEMPO, developed at the University of Kassel. In particular, a cell centered finite volume scheme with linear reconstruction, the Barth Jesperson limiter and AUSMDV flux function is employed [12]. The maximal number of Newton iterations is 30, the maximal dimension of the Krylov subspace is 30 and ILU preconditioning is used, with the ILU decomposition being updated every 30 time steps. To determine the new time step, the H211PI controller of Söderlind is employed. The solution after five seconds of real time is depicted in figure 10.

Furthermore, tolerance scaling is employed. Hereby, tolerance is scaled ($\tau$) by 1/100 for RODASP, 1/1000 for ROS34PW2 and 1/10000 for ESDIRK3 and ESDIRK4, in all cases $\alpha = 1$. The results of the simulations are shown in Figure 11.

As can be seen, the Rosenbrock schemes are about a factor two faster than the ESDIRK schemes with ROS34PW2 being the fastest one. The ES-
Figure 10: Pressure contours after 5 seconds. Left: Zoom on region around lower tube and shaft. Right: Zoom on region around upper tube and shaft.

Figure 11: Adaptive time stepping: accuracy and computational efficiency for different time integration schemes for the cooling of a flanged shaft test case.
DIRK schemes can end up in a situation where a right hand side is evaluated with NaN, causing the time step to be repeated with reduced time step size without this being a remedy, resulting in a stall of the computations with the time step converging to zero.

5.3 Three dimensional flow around a square cylinder

Finally, we consider a three dimensional test case, the flow around a square cylinder is considered. The Mach number is 0.3, and the square cylinder is rotated 45°. The Reynolds number for this test case is 300. A laminar flow is simulated with the Hexstream compressible flow solver. The object is held fixed in a uniform flow field, resulting in a vortex-street behind the square cylinder. When the initial transient has disappeared after approximately 100 seconds, an unsteady periodic flow is present. We then compute 20 s of real time.

The computational domain consists of 20.5L upstream of the centre of the square cylinder, 10.5L above and 11.5L below the centre, and 60.5L downstream of the centre of the square cylinder. Close to the square cylinder twenty extra layers of body conformal cells are generated, resulting in cells with a maximum aspect ratio of 365. In the third dimension, the computational domain has length 5L. The unstructured hexahedral mesh has 166,160 cells and 180,869 vertices, as shown in figure 12.

![Figure 12: Grid around a three dimensional square cylinder](image)

A series of computations is performed with the adaptive time stepping algorithm, and with the tolerance calibration applied reusing the coefficients shown in table 3. Again, the second preconditioner update strategy with the ILU(1) preconditioner is employed to automatically update the preconditioner. The maximum number of Newton iterations is 40, and the maximal dimension of the Krylov subspace is 50.
As shown in figure 13, the accuracies of the different time integration schemes lie close to each other as is expected. The fourth order time integration schemes clearly outperform the third order schemes in terms of computational efficiency. RODASP outperforms ESDIRK4 in terms of computational efficiency for the largest tolerances which are the most interesting computations from an engineering point of view, since for a relative error of $10^{-3}$, the relative error of the lift coefficient is approximately $10^{-2}$.

### 6 Summary and conclusions

We considered time adaptive implicit and linearly implicit time integration schemes for unsteady compressible viscous flows and judged them on computational efficiency. Specifically, Rosenbrock time integration schemes were compared to several ESDIRK schemes in the context of finite volume discretizations of the compressible Navier-Stokes equations. To solve the nonlinear and linear systems, a fast JFNK strategy is employed. Choosing the termination criterion for the linear solver in Rosenbrock schemes as the time adaptive tolerance $TOL/100$ results in an efficient termination of the linear solver.

Nonlinear convection-diffusion simulations show that the loss in accuracy for the Rosenbrock schemes is compensated by a more effective preconditioner for the constant stage matrix in particular for stiff problems and large tolerances. This is also observed in three more sophisticated two and three...
dimensional compressible flow problems where for low accuracies Rosenbrock time integration schemes outperform ESDIRK schemes in terms of computational efficiency.

For a proper use of the adaptive time stepping strategy a calibration is required. It is observed that the relative calibration between different time integration schemes is insensitive to the test problem although the absolute levels of the calibration constant $\tau$ may differ an order of magnitude depending on the test case.

It can be concluded that Rosenbrock time integration schemes are suitable candidates for engineering accuracies.

Acknowledgement

Financial support has been provided by the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) via the Sonderforschungsbereich Transregio 30, project C2.

References


