Master of Science Thesis

Rosenbrock time integration combined with Krylov subspace enrichment for unsteady flow simulations

Unsteady aerodynamics

David Blom

January 11, 2013

Aerodynamics

TU Delft

Faculty of Aerospace Engineering

Delft University of Technology

Rosenbrock time integration combined with Krylov subspace enrichment for unsteady flow simulations

Unsteady aerodynamics

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

David Blom

January 11, 2013

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF AERODYNAMICS

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance the thesis entitled **"Rosenbrock time integration combined with Krylov subspace enrichment for unsteady flow simulations"** by **David Blom** in fulfillment of the requirements for the degree of **Master of Science**.

Dated: January 11, 2013

Supervisors:

Prof. dr. ir. drs. H. Bijl

Dr. ir. A. H. van Zuijlen

Dr. P. Birken

Prof. dr. ir. K. Vuik

Dr. R. P. Dwight

Abstract

Many coupled problems have high computational demands, and therefore higher order time integration methods are commonly employed in order to increase the computational efficiency of unsteady fluid dynamics simulations. Usually second order, implicit schemes are used in engineering flow solvers. Substantial gains in computational efficiency can be acquired by e.g. using higher order time integration schemes, and by improving the convergence of the iterative solver.

John and Rang (2010) show that for incompressible flow, several Rosenbrock-Wanner methods outperform the Crank-Nicolson scheme and several multi-stage DIRK schemes in terms of computational efficiency. Rosenbrock-Wanner methods are derived from a linearisation of a DIRK scheme. Therefore, a gain in computational efficiency is observed, but some stability and accuracy properties are lost. W-methods use an approximation for the Jacobian, which can further increase the computational efficiency.

This thesis compares the efficiency of Rosenbrock time integration schemes with ESDIRK schemes, applicable to unsteady flow and fluid-structure interaction simulations. By solving the linear systems with the iterative solver GMRES, the preconditioner can be reused, and the Krylov subspace vectors can be reused for the different Rosenbrock stages improving computational efficiency (Carpenter et al., 2010). The influence of the convergence level of the linear solver on computational efficiency and stability is investigated, and the impact of the reuse of Krylov subspace vectors in comparison with the standard GMRES approach is studied. Results of simulations for unsteady flow show a gain in computational efficiency of approximately factor five in comparison with ESDIRK.

Preface

This report marks the end of my stay as a student in Delft, and is the product of my Master of Science research project. The research was performed at the Aerodynamics chair of the Aerospace Engineering faculty of the Delft University of Technology. I really enjoyed working and performing research on this subject.

I would like to thank my supervisors Hester Bijl and Alexander van Zuijlen for their guidance and help during the past months. Their support and advice proved to be of a great help. Many thanks to Philipp Birken and Andreas Meister for the numerous phone calls discussing the results of the performed research. And I would like thank Philipp Birken, Kees Vuik and Richard Dwight for being in my graduation committee. I would like to thank Daniel Rixen and Alexander van Zuijlen for their lectures on fluid-structure interaction, during which my interest in this field was grown.

And of course I would also like to thank all my family and friends for their support, interest and also for having a great time. Special thanks to Sara Lisa Biesbrouck for her support throughout the whole project. Thank you all!

David Blom
Delft, January 11, 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Efficient time integration methods applicable to fluid dynamics and fluid-structure interaction simulations are of high importance. During the research project, several time integration schemes and linear iterative solvers are studied. The goal of the masters thesis is to improve unsteady flow solvers by comparing high order Rosenbrock time integration schemes with implicit Runge-Kutta methods for convection-diffusion problems and viscous flows, in order to decrease computational times for fluid simulations which span several weeks for certain cases with current hardware set-ups. The motivation, goal and approach for the masters thesis can be found in this chapter.

1.1 Motivation

The interaction between fluids and structures is of high importance for many engineering applications. Air plane structures are not completely rigid structures, and several steady and dynamic aeroelasticity phenomena arise when structural deformations induce changes on the aerodynamic forces acting on the aircraft. These interactions may reach an equilibrium point, but divergence or flutter may also occur, which may cause catastrophic accidents. In September 1997, a U.S. Air Force F-117 "Stealth" fighter crashed. The aircraft crashed due to flutter excited by the vibration from a loose elevon.

A famous example of fluid-structure interaction in the field of civil engineering is the Tacoma Narrows Bridge. The first Tacoma Narrows bridge opened to traffic on July 1, 1940. Four months later on November 7, 1940, its main span collapsed into the Tacoma Narrows. Aeroelastic flutter was caused by a 68 kilometres per hour wind. Figure 1.1 shows a picture of the opening day of the bridge (Figure 1.1(a)) and the collapse is shown (Figure 1.1(b)).

(a) The opening day of the Tacoma Narrows bridge at July 1, 1940 (University of Washington, 1940a).

(b) The collapse of the Tacoma Narrows bridge at November 7, 1940 (University of Washington, 1940b).

Figure 1.1: Tacoma Narrows bridge. The opening day at July 1, 1940, and the collapse at November 7, 1940.

These two examples show that it is extremely important to analyse the dynamics of coupled systems during the design phase. The instability of coupled systems and the performance of fluid-structure systems need to be predicted. Other examples are to be able to predict the actual lift of a wing, and to evaluate the effect of surgical interventions to the functioning of the heart.

1.2 Approach

The approach followed to suggest improvements for unsteady flow solvers is to identify the main parts of an unsteady flow solver. Possible areas of improvement are identified and solutions already proposed in the literature are used. Numerical solution techniques are necessary to solve the governing equations for a fluid problem. Generally iterative methods are used in flow solvers to find the solution. The efficiency of the used approach depends on the combination of the different parts of the solver, such as the used time integration scheme, preconditioner and the iterative solver.

Currently, backward difference (BDF) time integration schemes are widely used in engineering codes. Explicit first-stage singly diagonal implicit Runge-Kutta (ESDIRK) methods have proven to be more efficient than BDF methods for engineering accuracies (Bijl et al., 2002; Jothiprasad et al., 2003; Wang and Mavriplis, 2007). An alternative to ESDIRK schemes which has not been widely applied for flow problems are Rosenbrock-Wanner schemes or Rosenbrock schemes. By linearizing the multi-stage Runge-Kutta schemes, Rosenbrock time integration schemes are derived. Rosenbrock schemes need to solve only one linear system per stage. Also, the different stages of the scheme can be effectively preconditioned by one preconditioner.

John and Rang (2010) show that for incompressible flow, several Rosenbrock-Wanner methods outperform the Crank-Nicolson scheme and several multi-stage DIRK schemes in terms of computational efficiency. These simulations were performed with the use of an adaptive time step selection algorithm.

In combination with the used time integration scheme, fluid solvers apply iterative techniques to find an approximation for the solution of the linear system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$. A subclass of Rosenbrock time integration methods are Krylov-ROW schemes, which use an approximation for the Jacobian, and apply Krylov subspace methods to compute a solution for the resulting linear system. Also, Rosenbrock time integration schemes solve the same linear system of equations multiple times per time step for different right-hand-sides. By reusing the vectors which build the Krylov subspace for subsequent stages of the Rosenbrock time integration scheme, a further efficiency gain is expected (Carpenter et al., 2010).

1.3 Goal

The objective of the masters thesis is to improve state-of-the-art unsteady flow solvers by comparing high order Rosenbrock-Wanner time integration schemes with implicit Runge-Kutta schemes for non linear convection-diffusion problems and viscous flows. The efficiency of the Rosenbrock schemes is improved by the reuse of Krylov subspace vectors for the subsequent Rosenbrock stages. Future research can focus on the application of the Rosenbrock scheme in a fluid-structure interaction problem.

The core questions of the research are:

1. How do Rosenbrock-Wanner time integration schemes compare to ESDIRK schemes in terms of numerical stability, accuracy and computational efficiency when applied to a non linear convection-diffusion and viscous flow problem?

2. Does the use of an adaptive time step control algorithm show a gain in efficiency for Rosenbrock-Wanner time integration schemes compared to ESDIRK schemes for non linear convection-diffusion and viscous flow problems?

3. Does the computational efficiency of the Rosenbrock-Wanner schemes improve when Krylov subspace vectors are reused for the subsequent stages of the Rosenbrock time integration scheme compared to standard Krylov subspace methods?

1.4 Structure

The structure of the thesis is as follows. Chapter 2 gives an overview of several time integration schemes. The ESDIRK and Rosenbrock-Wanner time integration schemes are used to simulate two test cases, namely a non linear convection-diffusion problem and a uniform flow around a cylinder. The reuse of Krylov subspace vectors is the subject of Chapter 3. The GMRES and GMRES-E algorithms are discussed, and also used to simulate the two test cases. Chapter 4 concludes the thesis with conclusions and recommendations.

Chapter 2

Time integration schemes for fluid dynamics

Chapter 1 gives the goal of the masters thesis and the approach to reach that goal, namely to improve unsteady flow solvers by comparing high order Rosenbrock time integration schemes with implicit Runge-Kutta methods for convection-diffusion problems and viscous flows. This chapter gives an overview of different time integration schemes, starting with implicit Runge-Kutta schemes in Section 2.2. Rosenbrock schemes are discussed in Section 2.3. The approach to solve the nonlinear systems of equations for implicit Runge-Kutta schemes is shortly touched upon in Section 2.4. The adaptive step size selection algorithm which increases the robustness of the time integration schemes is discussed in Section 2.5. The chapter is finalised with test results, and a short summary.

The reader is referred to Hairer and Wanner (1996) for further details on solution methods for stiff systems. The notation of Hairer and Wanner (1996) is used throughout this chapter. Also, Butcher (2003) is a good reference for several time integration schemes, with the main focus on Runge-Kutta schemes.

2.1 Introductory remarks

High order time integration methods are employed in order to increase the efficiency of unsteady computations. Currently, second order implicit schemes are commonly used in engineering codes. The use of implicit methods is advised, since explicit methods impose strict stability constraints on the time step used by the method. Contrary to explicit methods, the time step for implicit methods can be chosen based on accuracy considerations. For fluid flows, large differences in length and time scales are present, namely in the boundary layer, which increase the stiffness of

the system. Therefore, implicit schemes are preferred over explicit schemes for fluid solvers.

The different methods discussed in this chapter are multi-stage Runge-Kutta methods. Multi-step backward difference methods, which are currently widely applied in engineering codes, are not L-stable (Section 2.2.1) for third or higher orders, which is an advantageous property of a numerical scheme. These methods are therefore not part of this thesis. Also, multi-stage Runge-Kutta methods have proven to be computationally more efficient than second order time integration schemes for fluid flow computations (Van Zuijlen, 2006).

Rosenbrock-type methods are derived from diagonally implicit Runge-Kutta methods. This family of time integration methods replace the non-linear stages of implicit Runge-Kutta schemes with a sequence of linear systems. A non-linear implicit Runge-Kutta scheme, namely a diagonally implicit Runge-Kutta scheme (Section 2.2.2) is linearised.

Thus effectively a non-linear system of equations is replaced with a sequence of linear systems. The question arises whether the linearisation of the DIRK scheme has a small or significant influence on the accuracy and stability of the simulations. An advantage of the Rosenbrock scheme is that the preconditioner can be reused for the subsequent stages of the Rosenbrock scheme, resulting in a reduction of computational costs. It is expected that Rosenbrock time integration schemes are computationally more efficient compared to ESDIRK schemes. This hypothesis is examined in this chapter.

2.2 Implicit Runge-Kutta methods

This chapter covers methods to solve stiff problems, such as the Navier-Stokes equations. Due to the nature of stiff equations, new concepts of stability and mathematical relations for order restrictions for the shown methods are necessary. Large differences in length and time scales are present when fluid flows are considered. Therefore, implicit methods should be used to solve these problems.

The discussion starts with the implicit Euler method. The implicit Euler method is given by $y_{m+1} = y_m + h\, f(x_{m+1}, y_{m+1})$. When the method is applied to Dahlquist's test equation $y' = \lambda\, y$, the relation $y_{m+1} = y_m + h\,\lambda\, y_{m+1}$ is found. This gives the following relation:

$$y_{m+1} = R\,(h\,\lambda)\, y_m, \tag{2.1}$$

wherein the stability function $R(z)$ is given by $R(z) = \frac{1}{1-z}$. The stability domain given by the set $S = \{z \in \mathbb{C}; |R\,(z)| \leq 1\}$ is visualised in the complex plane as the exterior of the circle with radius 1 and centre $+1$. Thus, the stability domain covers the complete negative half-plane and a significant part of the positive half-plane as well. Concluding, the implicit Euler method has advantageous stability properties.

The s-stage implicit Runge-Kutta method is given by

$$g_i = y_m + h \sum_{j=1}^{s} a_{ij} f\left(x_m + c_j h, g_j\right) \qquad i = 1, ..., s, \tag{2.2}$$

wherein g_i are the stage values and

$$y_{m+1} = y_m + h \sum_{j=1}^{s} b_j f\left(x_m + c_j h, g_j\right), \tag{2.3}$$

is evaluated to obtain the solution at the next time level. The coefficients a_{ij} and b_j are typically given in the Butcher tableau. Contrary to an explicit Runge-Kutta method, the Butcher tableau is completely filled with non-zero terms. Therefore, s coupled non-linear systems of equations need to be solved at every step. This increases the computational costs considerably, compared to multi-step methods, which only require that one implicit system needs to be solved per time step. In order to be computationally efficient, the gain in accuracy for large time steps should compensate the increased computational costs per time step. The Butcher tableau for a general s-stage implicit Runge-Kutta method is shown in Table 2.1.

Table 2.1: Butcher tableau for an implicit Runge-Kutta method with s stages

c_1	a_{11}	a_{12}	$\dots$	a_{1s}
c_2	a_{21}	a_{22}	$\dots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\dots$	a_{ss}
	b_1	b_2	$\dots$	b_s

A general formulation employing the implicit Runge-Kutta schemes for unsteady fluid or structure problems is

$$\mathbf{w}^{(k)} = \mathbf{w}^n + \Delta t \sum_{i=1}^{s} a_{ki}\, \mathbf{F}^{(i)}, \qquad \mathbf{F}^{(i)} = \mathbf{F}\left(\mathbf{w}^{(i)}\right), \tag{2.4}$$

wherein $\mathbf{w}^{(k)}$ represents the solution of stage k. Note that the stage solutions $\mathbf{w}^{(k)}$ are generally of a low order. The solution at the next time level $\mathbf{w}^{n+1}$ of a high order is thus obtained by the summation of the residual function given by

$$\mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t \sum_{i=1}^{s} b_i\, \mathbf{F}^{(i)}. \tag{2.5}$$

The reader is referred to Hairer and Wanner (1996) for details on designing a multi-stage Runge-Kutta method.

The stability function for an implicit Runge-Kutta method is given by (Hairer and Wanner,

1996) as

$$R(z) = 1 + z\,\mathbf{b}^T\left(\mathcal{I} - zA\right)^{-1}\mathbf{e} = \frac{\det\left(\mathcal{I} - z\,A + z\,\mathbf{e}\,\mathbf{b}^T\right)}{\det\left(\mathcal{I} - z\,A\right)}, \tag{2.6}$$

wherein $\mathbf{b}^T = (b_1,\, ...,\, b_s)$, $A = (a_{ij})_{i,j=1}^{s}$, and $\mathbf{e} = (1, ..., 1)^T$. Thus the stability function is the quotient of two polynomials of degree s for an s-stage implicit Runge-Kutta method.

2.2.1 Stability

One measure to compare the time integration methods shown in this chapter, is to compare the stability and stiffness properties of the different methods. A non-mathematical description of stiffness is the following: the differential equation is called stiff, if explicit ordinary differential equation solvers diverge or need small step sizes to obtain an admissible approximation (Rang, 2004).

The following mathematical description is used to determine whether an ordinary differential equation (ODE) is called stiff (Rang, 2004). Suppose that the ODE

$$M\dot{\mathbf{u}} = \mathbf{f}\left(t, \mathbf{u}\right), \qquad \mathbf{u}\left(t_0\right) = \mathbf{u}_0, \tag{2.7}$$

has a solution $u(t)$, then Equation (2.7) is called *stiff* if for all $(t, u) \in J \times \mathbb{R}^n$, and the inequality

$$(t_1 - t_0)\sup\mu\left[f_u\left(t, u\right)\right] \leq \hat{\mu} \ll (t_1 - t_0)\sup \parallel f_u\left(t, u\right) \parallel \tag{2.8}$$

holds, where $f_u\left(t, u\right)$ is the Jacobian of $f\left(t, u\right)$, $\hat{\mu}$ is an upper bound for $(t_1 - t_0)\sup\mu\left[f\left(t, u\right)\right]$, and $\mu\left[A\right]$ is the logarithmic norm of the matrix A, which is defined by

$$\mu\left[A\right] = \lim_{h \to 0^+} \frac{\parallel I + h\,A \parallel - 1}{h}. \tag{2.9}$$

And, given a subset S of the totally or partially ordered set T, sup or supremum of the subset S, is the least element of the set T that is greater than or equal to every element of the subset S.

An ODE-solver can be A-stable or L-stable. An ODE-solver is called *A-stable* if the inequality

$$\lim_{Re\{z\} \to -\infty} \left|R\left(z\right)\right| < 1 \tag{2.10}$$

holds, wherein $R\left(z\right)$ is the stability function of the used method. For A-stable methods, the stability domain covers the entire left half-plane $\mathbb{C}^-$. This is an advantage for a numerical method, since the exact solution of Dahlquist's test equation is stable as well on the left half-plane $\mathbb{C}^-$ (Hairer and Wanner, 1996).

An ODE-solver is called L-*stable* if the limit

$$\lim_{Re\{z\}\to-\infty} |R(z)| = 0 \tag{2.11}$$

approaches zero for $Re\{z\} \to -\infty$. From Equations (2.10) and (2.11) it follows that an L-stable method is also A-stable.

Another stability property can also be used. An ordinary differential equation solver is called $A(\alpha)$-*stable* if $|R(z)| \leq 1$ for $z \in \mathbb{C}_\alpha^-$ (Widlund, 1967), where

$$\mathbb{C}_\alpha^- = \left\{ z \in \mathbb{C}^- : |\arg(z) - \pi| \leq \alpha, \quad \alpha \in \left(0, \frac{\pi}{2}\right) \right\}. \tag{2.12}$$

Note that methods which are $A(\alpha)$-stable have weaker stability properties compared to A-stable or L-stable methods. Similarly, a method can also be $L(\alpha)$-stable.

2.2.2 Diagonally implicit Runge-Kutta (DIRK) methods

General IRK methods can use the complete Butcher tableau. However, many sub classes exists. The diagonally implicit Runge-Kutta (DIRK) methods are such a subclass of the implicit Runge-Kutta methods. The DIRK method is also given by Equations (2.2) and (2.3). The difference lies in the values of the coefficients a_{ij}. The DIRK method has only implicit coefficients on the diagonal of the Butcher tableau, i.e. $a_{ij} = 0 \; \forall \; j > i$. The Butcher tableau for a general s-stage DIRK scheme is given by Table 2.2.

Table 2.2: Butcher tableau for a diagonally implicit Runge-Kutta method with s stages

c_1	a_{11}	0	$\dots$	$\dots$	0
c_2	a_{21}	a_{22}	0		$\vdots$
c_3	a_{31}	a_{32}	a_{33}	$\ddots$	$\vdots$
$\vdots$	$\vdots$		$\ddots$	$\ddots$	0
c_s	a_{s1}	$\dots$	$\dots$	$a_{s,s-1}$	a_{ss}
	b_1	b_2	$\dots$	$\dots$	b_s

The stages of this method can be solved sequential instead of simultaneous. Thus this method demands less computational resources than a general implicit Runge-Kutta method.

2.2.3 Single diagonal, diagonally implicit Runge-Kutta (SDIRK) methods

Single diagonal, diagonally implicit Runge-Kutta (SDIRK) schemes are a subclass of DIRK methods. The SDIRK method is also given by Equations (2.2) and (2.3). The SDIRK method has constant values on the diagonal of the Butcher tableau, i.e. $a_{ii} = \gamma$, and is optimised for fast convergence of a non-linear solver. Thus when an iterative solver is used, the number of iterations required to obtain the converged solution for each stage is lower than without the optimisation. Note that flow solvers usually use iterative methods.

The Butcher tableau for a SDIRK method is shown in Table 2.3. Appendix A shows the coefficients of several time integration schemes. A second order SDIRK scheme is shown in Table A.1, which is used for the time integration of the convection-diffusion test case.

Table 2.3: Butcher tableau for a single-diagonal, diagonally implicit Runge-Kutta method with s stages

c_1	γ	0	$\dots$	$\dots$	0
c_2	a_{21}	γ	0		$\vdots$
c_3	a_{31}	a_{32}	γ	$\ddots$	$\vdots$
$\vdots$	$\vdots$		$\ddots$	$\ddots$	0
c_s	a_{s1}	$\dots$	$\dots$	$a_{s,s-1}$	γ
	b_1	b_2	$\dots$	$\dots$	b_s

2.2.4 Explicit first stage, single diagonal, diagonally implicit Runge-Kutta (ESDIRK) methods

As the name suggests, explicit first stage, single diagonal, diagonally implicit Runge-Kutta (ESDIRK) methods are a subclass of SDIRK methods. The ESDIRK method is also given by Equations (2.2) and (2.3). The Butcher tableau for a general s-stage ESDIRK method is given by Table 2.4.

Table 2.4: Butcher tableau for a single-diagonal, diagonally implicit Runge-Kutta method with s stages

c_1	0	0	$\dots$	$\dots$	0
c_2	γ	γ	0		$\vdots$
c_3	a_{31}	a_{32}	γ	$\ddots$	$\vdots$
$\vdots$	$\vdots$		$\ddots$	$\ddots$	0
c_s	a_{s1}	$\dots$	$\dots$	$a_{s,s-1}$	γ
	b_1	b_2	$\dots$	$\dots$	b_s

For this method $a_{11} = 0$, denoting the explicit first stage, and $a_{21} = \gamma$, denoting a second-order accurately Crank-Nicolson scheme. When applied in the general formulation, $s - 1$ coupled systems of equations need to be solved, whereas for the IRK, DIRK, and SDIRK methods s coupled systems need be solved. Also, the solution at the last stage of the method is equal to the solution at the next time step, i.e. $a_{sj} = b_j$.

Appendix A lists the coefficients for a third, fourth and fifth order ESDIRK scheme in Tables A.2, A.3 and A.4, respectively.

2.3 Rosenbrock-type methods

Rosenbrock methods or Rosenbrock-Wanner are part of a class of linearly implicit Runge-Kutta methods. Rosenbrock methods replace non-linear systems with a sequence of linear systems. Rosenbrock schemes are derived by linearizing a DIRK scheme. As a result, some stability and accuracy properties are lost, but the computational costs per time step are reduced: s linear equation systems with a constant coefficient matrix and different right hand sides need to be solved, instead of s non-linear systems.

As mentioned, Rosenbrock methods are derived from the diagonally implicit Runge-Kutta method. Starting with a nonlinear DIRK scheme

$$k_i = h\, f\left(y_0 + \sum_{j=1}^{i-1} a_{ij}\, k_j + a_{ii}\, k_i\right) \qquad i = 1, ..., s, \tag{2.13}$$

where the solution at the next time step is given by

$$y_{m+1} = y_m + \sum_{i=1}^{s} b_i\, k_i, \tag{2.14}$$

the method is applied to the differential equation $y' = f(y)$. Equation (2.13) is linearised around $g_i = y_0 + \sum_{j=1}^{i-1} a_{ij} k_j$, as shown in

$$k_i = h\, f(g_i) + h\, f'(g_i)\, a_{ii}\, k_i. \tag{2.15}$$

In order to accelerate the computations, the Jacobians $f'(g_i)$ are replaced by $J = f'(y_0)$, such that the Jacobian needs to be evaluated only once during the Rosenbrock computation (Calahan, 1968). Thus an s-stage Rosenbrock method is described with the following two relations:

$$k_i = h\, f\left(y_0 + \sum_{j=1}^{i-1} \alpha_{ij}\, k_j\right) + h\, J \sum_{j=1}^{i} \gamma_{ij}\, k_j, \tag{2.16}$$

and

$$y_{m+1} = y_m + \sum_{j=1}^{s} b_j \, k_j, \tag{2.17}$$

with the coefficients α_{ij}, γ_{ij} and b_i, which are generally shown in a Butcher tableau.

The linearisation step can be interpreted as performing one Newton iteration at every step of the DIRK method, instead of a complete Newton loop. W-methods are obtained, if an approximation for the Jacobian is used. W-methods have additional order conditions. Krylov-ROW schemes are applied, if a Krylov subspace method is used to compute a solution for the linear system. Generally for nonlinear problems, Rosenbrock methods are less accurate and less robust than ESDIRK methods.

For implementation purposes, Equations (2.16) and (2.17) can be rewritten by introducing the new variables u_i. This approach is applied, since a direct implementation of Equations (2.16) and (2.17) requires the solution of a linear system with the matrix $I - h\,\gamma_{ii}\,J$ and the matrix-vector multiplication $J \cdot \sum \gamma_{ij}\,k_j$. This matrix-vector multiplication is avoided by introducing the new variables u_i:

$$u_i = \sum_{j=1}^{i} \gamma_{ij}\,k_j \qquad i = 1, ..., s\ . \tag{2.18}$$

If $\gamma_{ij} \neq 0$ for $j \leqslant i$, then the matrix $\Gamma = (\gamma_{ij})$ is invertible and k_i can be determined from u_i with

$$k_i = \frac{1}{\gamma_{ii}}\,u_i - \sum_{j=1}^{i-1} c_{ij}\,u_j, \tag{2.19}$$

wherein C is given by

$$C = \text{diag}\left({y_{11}}^{-1}, ..., {\gamma_{ss}}^{-1}\right) - \Gamma^{-1}. \tag{2.20}$$

Thus the following formulation of the Rosenbrock method is found for practical implementations,

$$\left(\frac{1}{h\gamma_{ii}}I - J\right) u_i = f\left(y_m + \sum_{j=1}^{i-1} a_{ij}u_j\right) + \sum_{j=1}^{i-1}\left(\frac{c_{ij}}{h}\right) u_j, \qquad i = 1, ..., s, \tag{2.21}$$

where y_{m+1} is given by

$$y_{m+1} = y_m + \sum_{j=1}^{s} m_j\,u_j, \tag{2.22}$$

wherein the coefficients a_{ij} and m_j are given by $(a_{ij}) = (\alpha_{ij})\,\Gamma^{-1}$ and $(m_1, ..., m_s) = (b_1, ..., b_s)\,\Gamma^{-1}$. Appendix A lists the coefficients for two third order (ROS34PW2, ROSI2PW and ROS34PRW) and a fourth order Rosenbrock method (RODASP) in Tables A.5, A.6, A.7 and A.8, respectively.

Concluding, Rosenbrock-type methods are presented as an alternative to ESDIRK time integration schemes. Rosenbrock-Wanner methods can be used, which use an approximation for the Jacobian, thus effectively reducing the computational costs per time step. However, the accuracy

and stability are also reduced per time step. When Krylov-ROW schemes are applied, a Krylov subspace method is used to compute the solution for the linear system resulting from the Rosenbrock scheme. Note that Rosenbrock schemes are generally less accurate than ESDIRK methods, since stability and accuracy properties are lost after the linearisation of the implicit Runge-Kutta scheme.

2.4 Nonlinear systems of equations

The implicit Runge-Kutta schemes lead to a nonlinear system of equations of the form

$$\mathbf{u} = \tilde{\mathbf{u}} + \alpha \, \Delta t \, \hat{\mathbf{f}} \left(\mathbf{u} \right), \tag{2.23}$$

where $\mathbf{u} \in \mathbb{R}$ is the unknown vector, α is a parameter, and $\tilde{\mathbf{u}}$ is a given vector. The function $\hat{\mathbf{f}} \left(\mathbf{u} \right)$ performs the temporal and spatial discretisation of the computational domain.

Multi grid methods are currently widely applied in engineering codes to solve this nonlinear system of equations. In practice, the Full Approximation Scheme (FAS) is used to apply multi grid directly to the nonlinear problem. A nonlinear iteration is employed to smooth the error. The full system is solved on a coarse mesh, and the coarse-grid error is determined from this solution. The coarse-grid correction is interpolated and applied on the fine mesh approximation for the nonlinear problem (Van E. Henson, 2003).

Another approach is to use a linearisation scheme, namely Newton's method, and consequently use multi grid to solve the resulting linear system. Still another approach is to apply an inexact Newton method to solve the nonlinear system of equations, which uses an iterative solver for the linear system. This approach is used in this thesis and discussed in this section.

2.4.1 Newton's method

The root problem

$$\mathbf{F} \left(\mathbf{u} \right) = \mathbf{0}, \tag{2.24}$$

is solved by Newton's method, which is also commonly called the Newton-Raphson method. The following procedure is followed repeatedly until the convergence criteria are satisfied:

$$\begin{aligned} \frac{\partial \, \mathbf{F} \left(\mathbf{u} \right)}{\partial \, \mathbf{u}} \, \Delta \mathbf{u} &= -\mathbf{F} \left(\mathbf{u}_n \right) \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta \, \mathbf{u}, \qquad n = 0, 1, 2, 3, ... \end{aligned} \tag{2.25}$$

The convergence criterion

$$||\,\mathbf{F}\,(\mathbf{u}_{n+1})||_2 \leqslant \tau_r\;||\,\mathbf{F}\,(\mathbf{u}_0)||_2 + \tau_a, \tag{2.26}$$

is used to determine whether the Newton iterations have been converged. The relative error tolerance τ_r and the absolute error tolerance τ_a are input parameters of the algorithm.

Note that Newton's method assumes that the function $\mathbf{F}\,(\mathbf{u})$ is differentiable, and that the initial iterate is "sufficiently near" the solution of the nonlinear problem. An advantage of Newton's method is that when the resulting linear equations are solved exactly, the method converges quadratically. A drawback of Newton's method is that slow convergence or stall can occur when the initial iterate is not close to the solution of the nonlinear problem.

Since the initial iterate is derived from the solution at the previous time step, it is expected that the assumption that the initial iterate is "sufficiently near" the solution of the nonlinear problem holds for the test cases considered in this thesis. Besides, in case the initial iterate is not close to the root, line search methods or trust region methods can be applied to increase the robustness of the method (Kelley, 1995). Also, the procedure can be started with a number of nonlinear multi grid iterations, followed by the Newton iterations. This approach has been used by Lucas (2010) and Bijl and Carpenter (2005) for compressible flow and fluid-structure interaction problems in order to reach fast nonlinear convergence.

Since an exact Jacobian may not available for the Navier-Stokes solver, and the resulting linear system is iteratively solved, an inexact Newton method is used to solve the nonlinear system, which is the subject of the following subsection.

2.4.2 Inexact Newton method

Inexact Newton methods use an approximate solution of the linear equation for the Newton step, in which the step satisfies

$$\begin{aligned} \left|\left|\frac{\partial\,\mathbf{F}\,(\mathbf{u})}{\partial\,\mathbf{u}}\Big|_n \Delta\mathbf{u} + \mathbf{F}\,(\mathbf{u}_n)\right|\right| &\leqslant \eta_k\,||\,\mathbf{F}\,(\mathbf{u}_n)|| \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta\,\mathbf{u}, \qquad n = 0, 1, 2, 3, ..., \end{aligned} \tag{2.27}$$

where the tolerance parameter $\eta_k \in \mathbb{R}$ is called the forcing term of the inexact Newton method. Eisenstat and Walker (1994) discuss the choice of the forcing term of the inexact Newton method. The efficiency of the Newton method depends on the choice of the forcing terms. Also, robustness of the method can be affected by the choice of the forcing terms. In case the forcing terms are chosen too strict, then over solving may occur, i.e. the linear system arising from the linearisation step are solved to a precision far beyond what is necessary for a correct nonlinear iteration. In case the forcing terms are chosen relatively large, slow convergence or even divergence of the Newton

iterations may occur. It is shown that if η_k converges to zero fast enough, convergence of the method is locally quadratic.

The following procedure is followed to determine the forcing terms, as Eisenstat and Walker (1994) have proposed in their article. First the parameter ${\eta_k}^A$ is determined with

$$ {\eta_n}^A = \gamma \frac{||\mathbf{F}(\mathbf{u}_n)||^2}{||\mathbf{F}(\mathbf{u}_{n-1})||^2}, \tag{2.28} $$

for a given parameter $\gamma \in (0, 1]$. If the sequence ${\eta_n}^A$ is uniformly bounded away from one, convergence of the method is quadratic. In order to start the sequence and bound it away from one, the value of η_n is limited with

$$ {\eta_n}^B = \begin{cases} \eta_{max}, & n = 0, \\ \min\left(\eta_{max},\, {\eta_k}^A\right), & n > 0 \end{cases} . \tag{2.29} $$

Eisenstat and Walker (1994) suggest to apply safeguarding in order to avoid volatile decreases of the forcing term η_n by

$$ {\eta_n}^C = \begin{cases} \eta_{max}, & n = 0, \\ \min\left(\eta_{max},\, {\eta_k}^A\right), & n > 0, \gamma\, {\eta_{k-1}}^2 \leqslant 0.1 \\ \min\left(\eta_{max},\, \max\left({\eta_k}^A, \gamma\, {\eta_{k-1}}^2\right)\right) & n > 0, \gamma\, {\eta_{k-1}}^2 > 0.1 \end{cases} . \tag{2.30} $$

The term $\gamma\, {\eta_{n-1}}^2$ is used as a criterion to determine whether η_{n-1} is sufficiently large. η_n is not decreased by more than a fraction of η_{n-1}.

In order to avoid over solving of the final step of Newton method, the norm of the current nonlinear residual $||\mathbf{F}(\mathbf{u_n})||$ is compared to the nonlinear residual norm at which the iterations would stop

$$ \tau_t = \tau_a + \tau_r ||\mathbf{F}(\mathbf{u}_n)|| . \tag{2.31} $$

η_n is bounded from below by a constant multiple of $\dfrac{\tau_t}{||\mathbf{F}(\mathbf{u_n})||}$. Kelley (1995) suggests to use

$$ \eta_n = \min\left(\eta_{max},\, \max\left({\eta_k}^C,\, \frac{0.5\,\tau_t}{||\mathbf{F}(\mathbf{u}_n)||}\right)\right), \tag{2.32} $$

and with the parameter $\gamma = 0.9$.

To solve the linear system of the Newton iteration, an iterative solver is used, such as GMRES. A Jacobian free version of Newton's method can be formulated, which is the subject of the next subsection.

2.4.3 Jacobian-free Newton-Krylov method

Jacobian-free Newton-Krylov methods are a combination of Newton methods and Krylov subspace methods to solve the Newton correction equations. The Jacobian-vector product links the Newton method and the Krylov subspace method, which can be approximated with a finite difference approach. For preconditioning of the Krylov iterations, an approximation of the Jacobian matrix can still be required. The reader is referred to Knoll and Keyes (2004) for a comprehensive overview of Jacobian-free Newton-Krylov methods. Lucas (2010) discusses the implementation details of the used Jacobian-free Newton-Krylov solver.

As mentioned, the matrix vector products required by GMRES, which will be discussed in Chapter 3, can be approximated with a finite difference quotient via

$$\frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}}\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}+\epsilon\,\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon}, \tag{2.33}$$

where the parameter ϵ is determined with

$$\epsilon = \frac{\sqrt{\epsilon_{mach}}}{||\mathbf{v}||_2}, \tag{2.34}$$

and where ϵ_{mach} is the machine precision. Here the approach of Qin et al. (2000) is followed to determine the parameter ϵ. Cancellation errors can become a major problem if the parameter ϵ is chosen too small. The use of Equation (2.34) has proven to be effective.

A second order approximation for the matrix-vector product can be used to increase the robustness of the solver:

$$\frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}}\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}+\epsilon\,\mathbf{v}) - \mathbf{F}(\mathbf{u}-\epsilon\,\mathbf{v})}{2\,\epsilon}. \tag{2.35}$$

Note that the amount of computational work is increased, since two function evaluations are necessary to determine the matrix-vector product, instead of one for the first order approximation shown in (2.33). However, the number of Newton iterations necessary for convergence is decreased in case of an SDIRK or ESDIRK time integration scheme (Knoll and Keyes, 2004). When the Rosenbrock time integration is applied, the accuracy of the approximation of the solution of the linear system is increased. During preliminary computations performed during this research project, the use of this second order approximation of the matrix-vector product has shown to be necessary when Rosenbrock schemes are used, since with the first order approximation computations crashed due to function evaluations equal to NaN. Stability of the Rosenbrock scheme is still an issue for relatively large time steps, but the robustness of the Rosenbrock scheme can be further increased with the use of an adaptive step size selection algorithm, which is discussed in the following section.

2.5 Adaptive time step control

Time step control is an important measure to increase the efficiency and robustness of a time integration method. A constant time step often results in a large number of small steps, increasing the computational costs of a simulation significantly. A simple approach for an adaptive time step control method would be to compare the change of the solution of two subsequent time steps in the L^2-norm of the space-time interval. For Runge-Kutta and Rosenbrock schemes, an embedded scheme can be used as an error estimator.

The numerical error needs to be determined for the adaptive step control algorithm. Often, an absolute error is applied, as shown in

$$r_n = ||\widehat{\mathbf{w}}_n - \mathbf{w}_n||, \tag{2.36}$$

where $\widehat{\mathbf{w}}_n$ and $\mathbf{w}_n$ are two solutions at time step t_n of different order. Hairer and Wanner (1996) propose the error

$$r_n = \sqrt{\frac{1}{N}\sum\left(\frac{w_i - \widehat{w}_i}{\delta + \max\{|w_i|, |\widehat{w}_i|\}}\right)^2}, \tag{2.37}$$

where δ is a scaling factor, which is typically chosen in between 10^{-6} and 1. In the test cases discussed in this thesis, Equation (2.36) is used to determine the numerical error.

2.5.1 Standard controller

The model for the error - step size relation is the most important for the design of the adaptive step size selection algorithm (Gustafsson, 1991). The next time step can be chosen by using the standard controller (Hairer and Wanner, 1996; Lang, 1999)

$$h_{n+1} = \rho\left(\frac{TOL_t}{r_n}\right)^{\frac{1}{p}} h_n, \tag{2.38}$$

where ρ denotes a safety factor, usually $\rho = 0.8$ or 0.9, $TOL_t > 0$ represents a given tolerance, p the order of the local error estimator, and h_{n+1} is the new time step. If the numerical error r_n at time step t_n is bigger then the prescribed tolerance, then time step t_n is rejected, and is recomputed with a smaller step $h_n := h_{n+1}$. Notably, the standard controller is the simplest and most common candidate for the model of the error - step size relation.

A disadvantage of the standard controller is that too many time steps are rejected (Lang, 1999). Gustafsson (1994) proposes to use more data from previous time steps to decide on the new step size, as shown in the step size selection rule

$$h_{n+1} = \rho\,\frac{{h_n}^2}{h_{n-1}}\left(\frac{TOL_t\, r_{n-1}}{{r_n}^2}\right)^{\frac{1}{p}}. \tag{2.39}$$

This approach has been thoroughly tested by John and Rang (2010) for several DIRK and Rosenbrock schemes. John and Rang show that the computational efficiency and accuracy of the solution highly depend on the tolerance TOL_t for the adaptive time step control. The selection of the tolerance depends on the used time integration scheme. Yet another approach is the use of digital filters, as discussed in the following section.

2.5.2 Digital filters

A further improvement of the step size selection rule is the use of digital filters and limiters. The application of digital filters and limiters has been shown to have a major impact on computational stability (Söderlind and Wang, 2006a), contrary to the classical approach. Thus a small change of the tolerance parameter TOL_t only leads to a small change in the accuracy of the produced solution, as well as on the work required to achieve that accuracy. The standard controller, PI-controller and H211b digital filter are all covered by the generic step size recursion

$$h_{n+1} = \left(\frac{\epsilon}{r_n}\right)^{\beta_1} \left(\frac{\epsilon}{r_{n-1}}\right)^{\beta_2} \left(\frac{h_n}{h_{n-1}}\right)^{-\alpha_2} h_n, \tag{2.40}$$

if the dynamics are limited to at most second order (Söderlind, 2003). Here ϵ is determined with $\epsilon = \rho TOL_t$, where ρ again represents a safety factor. Table 2.5 shows the parameterisations of three different controllers. Here p represents the order of the local error estimator. Note that the PI-controller and the H221b digital filter are not self-starting. In actual implementations, the procedure starts with the standard controller, and thereafter the PI-controller or digital filter continues the process.

Table 2.5: Three first order adaptive controllers which can be used in the adaptive time step control algorithm

$p\,\beta_1$	$p\,\beta_2$	α_2	Name
1	-	-	Standard controller (Hairer and Wanner, 1996)
$\frac{3}{5}$	$-\frac{1}{5}$	-	PI.4.2 controller (Gustafsson, 1994; Söderlind, 2002)
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	H211b digital filter ($b = 4$) (Söderlind, 2003)

2.5.3 Limiters and anti-windup

Smooth limiters can also be used to further improve the adaptive time step control algorithm. Discontinuities are introduced in the step size ratio $\frac{h_{n+1}}{h_n}$ when the maximum step size increase and decrease are limited (Söderlind and Wang, 2006b). By applying a smooth limiter this problem

can be solved efficiently. The new step size is determined with $h_{n+1} = \rho_n \, h_n$ where ρ_n is given by

$$\rho_n = \left(\frac{\epsilon}{r_n} \right)^{\beta_1} \left(\frac{\epsilon}{r_{n-1}} \right)^{\beta_2} {\rho_{n-1}}^{-\alpha_2}. \tag{2.41}$$

Although (2.41) is equivalent to (2.40), step size rejections may be reduced by basing the test on the requested change ρ instead on the error. When the limiter

$$\widehat{\rho}_n = 1 + \kappa \arctan\left(\frac{\rho_n - 1}{\kappa} \right), \tag{2.42}$$

with $h_{n+1} = \widehat{\rho}_n \, h_n$ is used to determine the time step, the previously mentioned discontinuities in the step size change ratio $\frac{h_{n+1}}{h_n}$ are removed. The parameter κ determines the maximum step size increase or reduction. The effect of the limiter is increased by decreasing the value of κ. Hence, with a smaller value of κ, the interval where $\widehat{\rho} \approx \rho_n$ is shortened. A common choice for κ is $\kappa = 1$. Figure 2.1 shows a plot of $\widehat{\rho}_n$ for $\kappa = 0.5$, $\kappa = 1.0$ and $\kappa = 2.0$, giving an visual indication of the effect of the limiter.

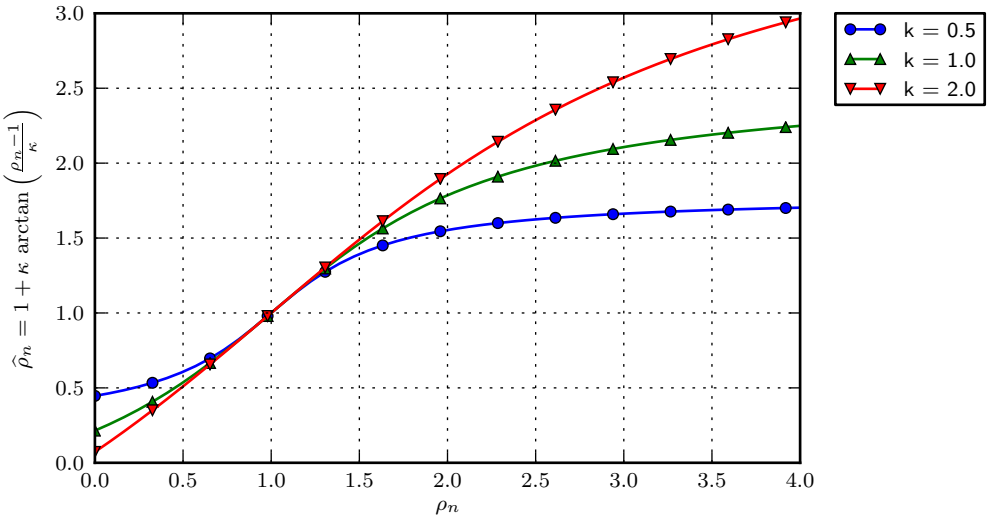


Figure 2.1: Plot of the smooth limiter $\widehat{\rho}_n = 1 + \kappa \arctan\left(\frac{\rho_n - 1}{\kappa} \right)$ where $\kappa = 0.5$, $\kappa = 1.0$ and $\kappa = 2.0$ for a range of values for ρ_n.

The pseudo code for the complete controller is shown in Algorithm 2.1.

2.5.4 Tolerance scaling and calibration

In order to compare the computational efficiency of different time integration schemes, it is desirable to scale and calibrate the tolerance of the adaptive 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,

Algorithm 2.1 Adaptive time step controller: the standard controller is combined with the digital filter and the smooth limiter for a stable time integration

$c_1 = \frac{\epsilon}{r}$
if first time step or first step after successive rejects **then**
    $\rho = {c_1}^{\frac{1}{k}}$
else
    $\rho = {c_1}^{\frac{1}{4\,k}} \cdot {c_0}^{\frac{1}{4\,k}} \cdot \rho^{-\frac{1}{4}}$
end if
$\widehat{\rho} = 1 + \kappa \cdot \arctan\left(\frac{\rho - 1}{\kappa}\right)$
$c_0 = c_1$
$h = \widehat{\rho} \cdot h$
if $\widehat{\rho} < 0.9$ **then**
    Time step is rejected. Recompute with new time step
end if

the different time integration schemes should deliver the same accuracy for the same tolerance setting (Söderlind and Wang, 2006b).

The scaling transformation

$$TOL_t{}' = {TOL_{t0}}^{\frac{\alpha-1}{\alpha}}\, {TOL_t}^{\frac{1}{\alpha}} \tag{2.43}$$

can be used to compare the computational efficiencies of the ESDIRK and Rosenbrock time integration schemes (Söderlind and Wang, 2006b). $TOL_t{}'$ represents the tolerance parameter used by the adaptive step size control algorithm, TOL_t is the parameter specified by the user, TOL_{t0} is the equivalence point determined during the calibration, and α is the measured order of the adaptive step size control algorithm of the reference computations for the calibration.

2.5.5 Error criteria

Reliable error measurements are necessary in order to determine the accuracy of time step evaluations. The standard approach is to use the ratio $\left(\frac{\epsilon}{r}\right)$ for the adaptive step size control algorithm. The fixed resolution test and fixed scaling test are also used to judge the accuracy of a solution after a time step evaluation. The fixed resolution test and fixed scaling test are shortly discussed.

Fixed resolution test

The fixed resolution test is given by the inequality

$$|| \hat{\mathbf{l}}./\mathbf{d}\left(\mathbf{w}_n, \boldsymbol{\rho}\right) || \leqslant 1, \tag{2.44}$$

where the vector $\mathbf{d}\left(\mathbf{w}_n, \boldsymbol{\rho}\right)$ is defined as

$$d_i\left(\mathbf{w}_n, \boldsymbol{\rho}\right) = RTOL_t \left|w_{n_i}\right| + \rho_i, \tag{2.45}$$

where $RTOL_t$ is a predefined relative tolerance parameter and the vector ρ specifies the resolution level of w_{n_i}. $\hat{\mathbf{l}}$ is a given local error estimate, typically implemented as $\widehat{\mathbf{w}}_n - \mathbf{w}_n$ for Runge-Kutta and Rosenbrock schemes, thus subtracting the solutions of different orders of the time integration schemes.

The error estimate is accepted if the inequality (2.44) is satisfied. The inclusion of the resolution level acts as a noise floor, typically a characteristic of the simulated problem. For fluid dynamic simulations, a combination of a fixed resolution test and fixed scaling test can be used, which is discussed next.

Fixed scaling test

The error estimate is accepted if the fixed scaling test

$$|| \hat{\mathbf{l}}./\mathbf{d}\left(\mathbf{w}_n, \boldsymbol{\eta}\right) || \leqslant TOL_t \tag{2.46}$$

is satisfied, where the vector $\mathbf{d}\left(\mathbf{w}_n, \boldsymbol{\eta}\right)$ is defined as

$$d_i\left(\mathbf{w}_n, \boldsymbol{\eta}\right) = \left|w_{n_i}\right| + \eta_i. \tag{2.47}$$

Thus the norm depends on the solution $\mathbf{w}_n$, and the scaling η_i of each component of the solution w_{n_i}. The fixed scaling test measures relative accuracy for the different components of $\mathbf{w}_n$ with $|w_{n_i}| > \eta_i$ and absolute accuracy with $|w_{n_i}| < \eta_i$.

For the simulations performed in this thesis a combination of the fixed resolution test and fixed scaling test is used by using the vector $d_i = RTOL_t\left(\left|w_{n_i}\right| + 1\right)$, thus enforcing the scale 1 with $RTOL_t$ as the noise floor. Also, a purely relative error criterion is applied with $d_i = TOL_t \left|w_{n_i}\right|$ for the compressible flow test case. The new step size is determined by applying Equation (2.41), which is adapted to apply the new error criterion, as shown below:

$$\rho_n = \left(|| \mathbf{d}./\hat{\mathbf{l}} ||_n\right)^{\beta_1} \left(|| \mathbf{d}./\hat{\mathbf{l}} ||_{n-1}\right)^{\beta_2} {\rho_{n-1}}^{-\alpha_2}. \tag{2.48}$$

Concluding, an adaptive time step selection algorithm can be efficiently implemented for Runge-Kutta schemes and Rosenbrock methods. By employing the embedded schemes, the error of the solution can be efficiently approximated. Also, the use of digital filters and limiters ensures computational stability, meaning that a small change of the tolerance parameter TOL_t only leads to a small change in the accuracy of the produced solution, as well as on the work required to achieve that accuracy.

2.6 Nonlinear convection-diffusion equation

In order to compare the Rosenbrock time integration schemes with the implicit Runge-Kutta schemes, several test problems have been used to verify the computational efficiency of the different methods discussed earlier in this chapter. A nonlinear convection-diffusion equation is used as the first test case, and is discussed in this section. A description of the problem is given, and the performance of the different time integration schemes is discussed. Fixed time step studies and adaptive time step studies have been performed for non-linear and linear convection-diffusion problems.

2.6.1 Description of the nonlinear convection-diffusion problem

This first test case consists of a generalised nonlinear convection-diffusion equation:

$$u_t = \beta\, u^n \cdot \nabla u + \alpha \nabla \cdot (u^m\, \nabla u)\,, \qquad x \in \Omega := (0,1) \times (0,1)\,, \tag{2.49}$$

where $u\,(x,y,0)$ is given by

$$u\,(x,y,0) = u_0\,(x,y)\,. \tag{2.50}$$

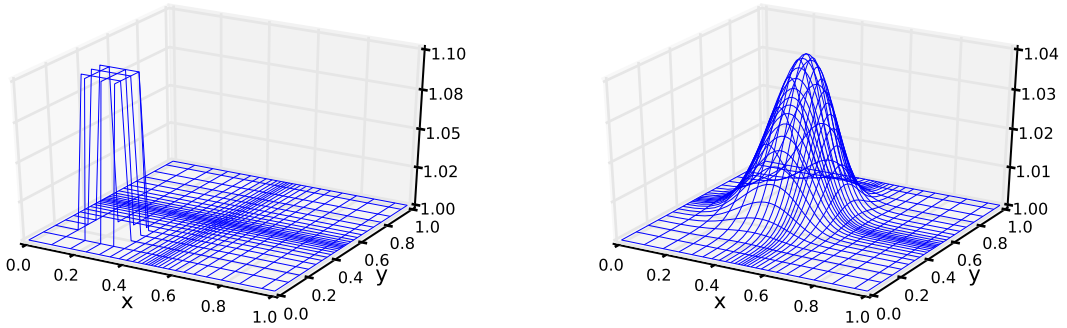
The strength of the diffusion is determined by the parameter $\alpha \in \mathbb{R}$. The strength and the direction of the convection is determined with

$$\beta = \tilde{\beta} \begin{pmatrix} \sin(\gamma) \\ \cos(\gamma) \end{pmatrix}, \tag{2.51}$$

where $\tilde{\beta} \in \mathbb{R}$ is a user specified parameter, and γ determines the angle of the direction of the convection. The degree of non linearity is determined with the parameters m and n. The initial solution used for this test case is shown in Figure 2.2(a). The initial solution is one in the complete domain, except on the square $[0.1, 0.3] \times [0.1, 0.3]$, where the initial value is 1.1. The values for α, β and γ are set to $\alpha = 1$, $\beta = 200$ and $\gamma = 0.35\,\pi$. The reference solution at the end of the simulation is shown graphically in Figure 2.2(b) for $n = m = 3$.

A non-uniform mesh is used for the computations shown in this section. Preliminary computations showed that for a uniform mesh, the Krylov subspace solver finished in one or two iterations.

As a result, the difference in computational time between Rosenbrock and ESDIRK was negligible. As shown in Figure 2.2, the mesh is refined close to $x = 0.5$ and $y = 0.5$ resulting in cells with a high aspect ratio.



(a) Initial solution for a convection-diffusion simulation. **(b)** Reference solution for a non-linear convection-diffusion simulation ($n = m = 3$) at time $t = 0.002$.

Figure 2.2: Initial solution and reference solution for a non-linear convection-diffusion simulation ($n = m = 3$). The reference solution has been obtained with ESDIRK5 and $\Delta t = 10^{-7}$. A non-uniform mesh of size 50×50 is used.

2.6.2 Performance of the time integration schemes

The performance of the different time integration schemes is discussed in this section. The effect of the non linearity parameters is studied first. Thereafter, a series of fixed time steps simulations have been performed in order to judge the computational efficiency of the different time integration schemes. Also, an adaptive time step study is performed in order to investigate the influence of the use of a time step control algorithm.

Incomplete LU factorisations are used as a preconditioner for the GMRES method, which is used as the linear solver of the incomplete Newton-Krylov method, and to solve the stages of the Rosenbrock scheme. The error E of one computation is determined by taking the root mean square of the difference between the solution and the reference solution, and scaling with the total number of cells N:

$$E = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \left(u_i - u_{i_{ref}} \right)^2 \tag{2.52}$$

Effect of non linearity parameters on accuracy

A fixed time study has been performed for different settings of the parameters n and m, which influence the degree of non linearity of the test problem. Figure 2.3 shows the results for the

SDIRK, ESDIRK and Rosenbrock time integration schemes. The linear systems arising from the Newton iterations, and the Rosenbrock stages are directly solved, thus no iterative error exists. By changing the non linearity parameters, its effect on the accuracy and efficiency of the different time integration schemes can be studied.

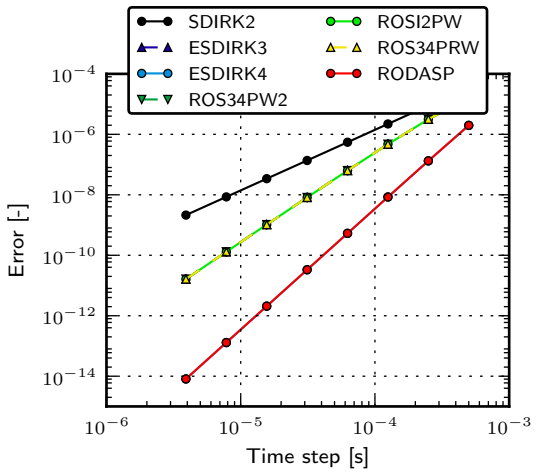
For the linear convection-diffusion case, the theoretical order of the different schemes is confirmed, which indicates that the SDIRK, ESDIRK and Rosenbrock schemes are correctly implemented. Since a linear convection-diffusion problem is considered, the ESDIRK and Rosenbrock schemes of the same order have the same error. For the non-linear cases, the error of the ESDIRK and Rosenbrock schemes is different. The error of the ESDIRK time scheme is smaller than the error of the Rosenbrock scheme of the same order. This is expected for the Rosenbrock schemes, since by linearizing the DIRK scheme, as discussed in Section 2.3, some accuracy and stability properties are lost. When the degree of non linearity is increased, the difference in error between the ESDIRK and Rosenbrock schemes of the same order increases, as can be clearly seen by comparing Figures 2.3(b) and 2.3(d).

Effect of the Newton-Krylov method and Krylov subspace method on accuracy and efficiency

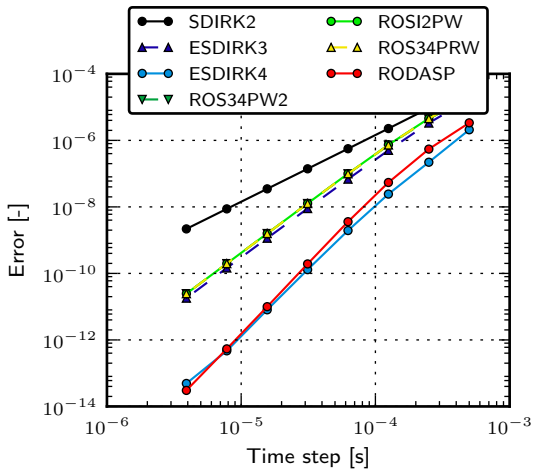
Another fixed time step study has been performed for the nonlinear convection-diffusion test case. The effect of the use of an incomplete Newton-Krylov solver and a Krylov subspace method for the ESDIRK and Rosenbrock schemes is investigated. The Eisenstat-Walker method has been used for the Newton iterations to determine the optimal forcing term for the Krylov subspace method. The tolerance setting for the Newton iterations TOL_{NK}, and GMRES in case of the Rosenbrock scheme TOL_{GMRES}, has a significant influence on the computational stability and accuracy. The accuracy and computational time for ESDIRK and Rosenbrock is compared graphically in Figure 2.4 for different tolerances for the Newton iterations and the GMRES algorithm. The nonlinear convection-diffusion test case is simulated with $n = m = 3$.

When the tolerance TOL_{NK} or TOL_{GMRES} is chosen too large, the order of the time integration schemes reduces to zero for relatively small time steps. In order to increase the computational stability of the simulations, the tolerance for the Newton iterations (TOL_{NK}) or GMRES (TOL_{GMRES}) has to be chosen more strict, as can be clearly seen in Figures 2.4(a), 2.4(c) and 2.4(e). Note that the order of the ESDIRK schemes reduces to zero earlier compared to the Rosenbrock schemes of the same order. When the tolerance for the Newton iterations or GMRES algorithm is chosen strict enough, a reduction of order does not appear, in this case for $TOL_{NK} = TOL_{GMRES} = 10^{-6}$, and $TOL_{GMRES} = 10^{-4}$ for the third order Rosenbrock schemes.

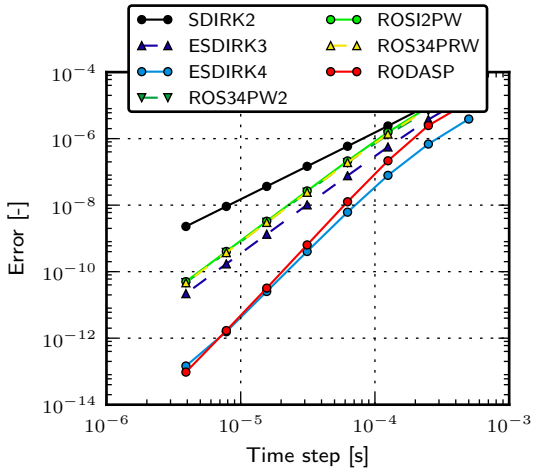
The Rosenbrock time integration schemes show a gain in computational efficiency compared to ESDIRK for all the different cases considered. For the smallest tolerance, the gain in computational



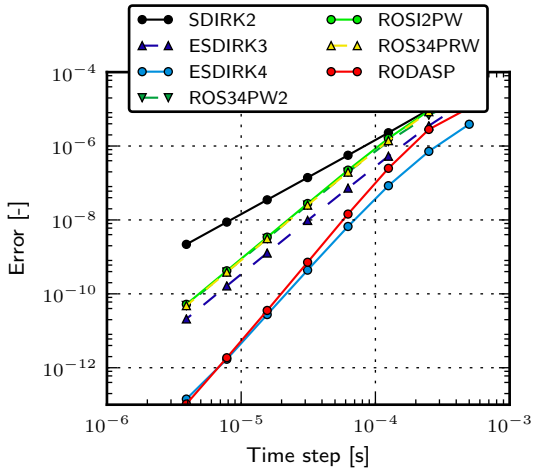
(a) Accuracy for non-uniform mesh for the linear convection-diffusion equation ($n = m = 0$)



(b) Accuracy for non-uniform mesh for the non-linear convection-diffusion equation ($n = m = 1$)



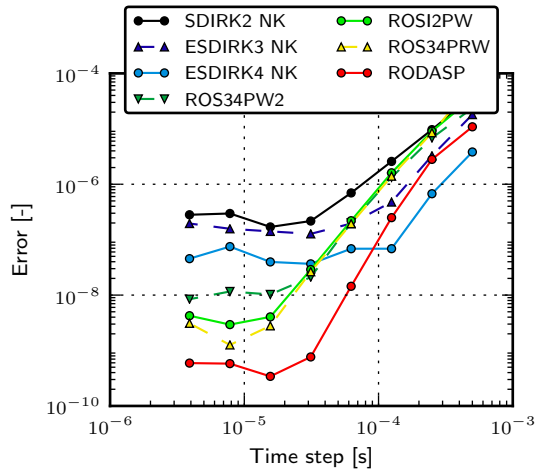
(c) Accuracy for non-uniform mesh for the non-linear convection-diffusion equation ($n = 3$, $m = 1$)



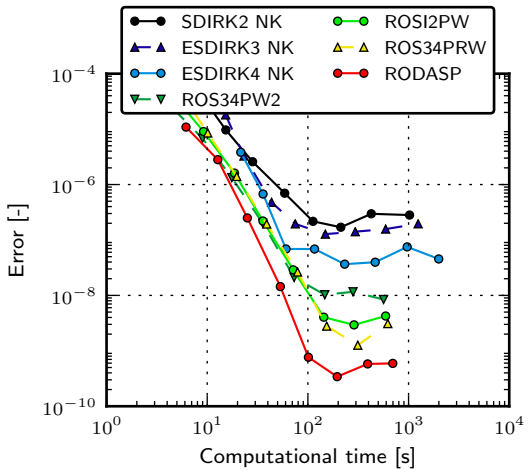
(d) Accuracy for non-uniform mesh for the non-linear convection-diffusion equation ($n = m = 3$)

Figure 2.3: Fixed time study studies for SDIRK, ESDIRK and Rosenbrock time integration schemes for the convection-diffusion problem with varying non-linearity settings. The linear systems arising from the Newton iterations, and Rosenbrock stages are directly solved. The simulations are performed on a non-uniform mesh (50×50).

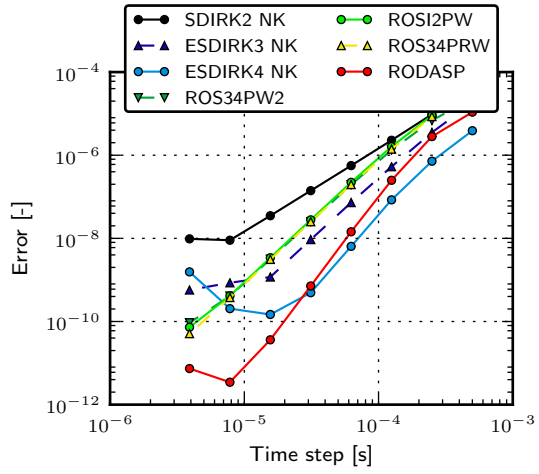
efficiency is the greatest. The second order SDIRK scheme is computationally more efficient than ESDIRK3 and ESDIRK4 for small accuracies. In case higher accuracies are needed, then the higher order methods are preferred over SDIRK2. Comparing the third and fourth order Rosenbrock schemes, the fourth order RODASP method has the greatest potential for use in a flow solver, due to the significant gain in computational efficiency.



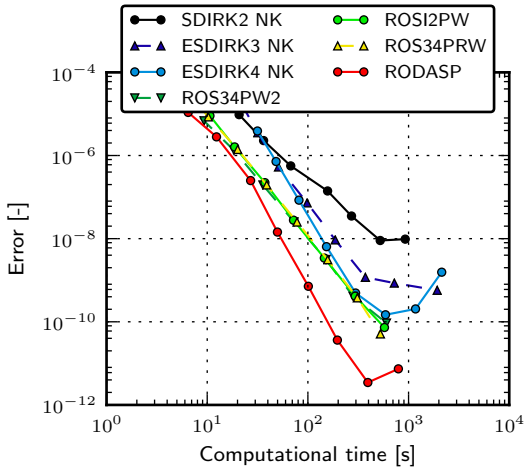
(a) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-2}$



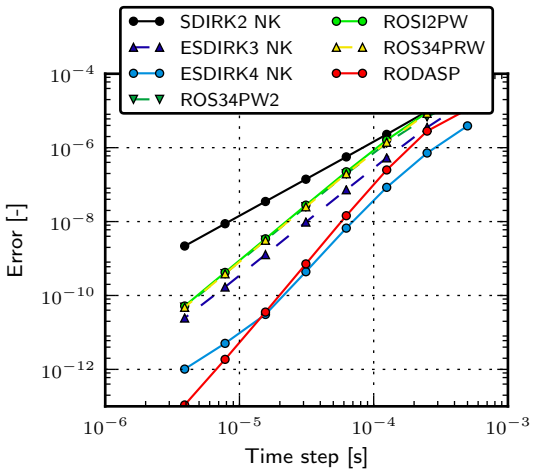
(b) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-2}$



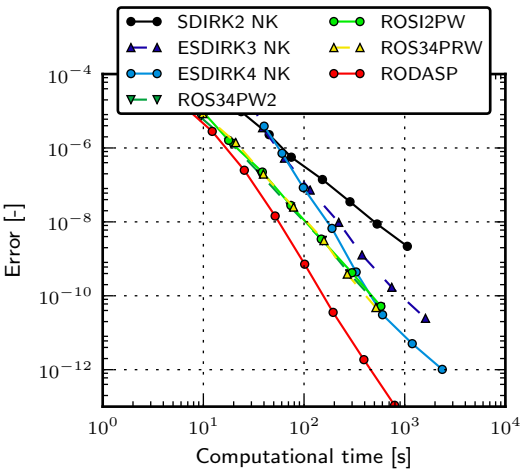
(c) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-4}$



(d) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-4}$



(e) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-6}$



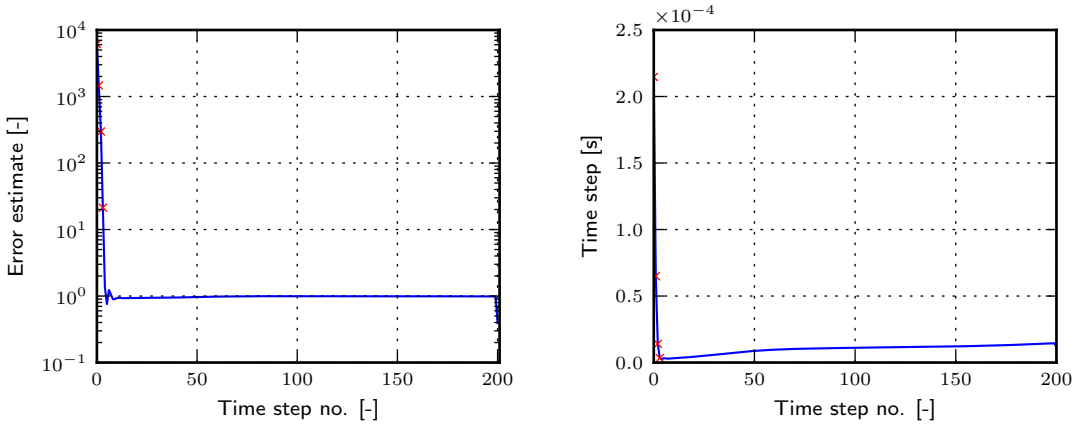
(f) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-6}$

Figure 2.4: Fixed time step study for the Newton-Krylov and Krylov subspace method comparing (E)SDIRK and Rosenbrock schemes with different tolerance settings used for the Newton iterations and the GMRES solver with $n = m = 3$.

Effect of adaptive time step control on accuracy and computational stability

Besides the fixed time step studies, also the use of the adaptive time step algorithms has been investigated. The classical controller and the digital filter combined with the smooth limiter and time step rejections, as discussed in Sections 2.5.1 and 2.5.2, are used to simulate the nonlinear convection-diffusion problem. For one simulation, the resulting error estimates and time step history are shown in Figure 2.5. Time steps which are rejected are denoted with a red cross.

As shown in Figure 2.5, the error estimate is controlled effectively by the digital filter, and is close to one for a large part of the simulation. The time step is increased by the controller, and shows no discontinuities due to the fact that the smooth limiter is applied. The initial time step is chosen relatively large for this simulation. Therefore, the first time steps are rejected. Note that time steps are only rejected at the start of the simulation. Thereafter, the error estimate is controlled and close to 1.0. Thus the remaining time steps are accepted resulting in an efficient time integration.



(a) Error estimate history for a non-linear convection-diffusion simulation

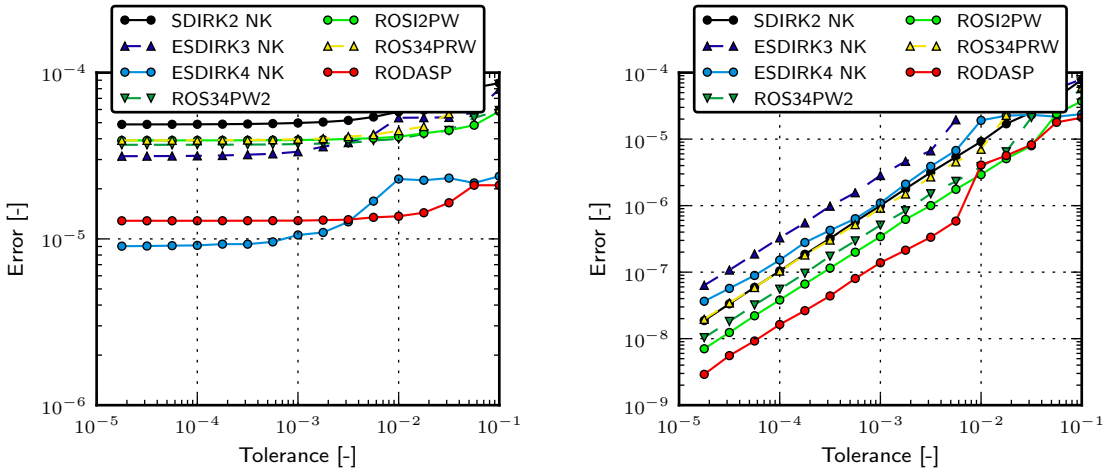
(b) Time step history for a non-linear convection-diffusion simulation

Figure 2.5: The error estimates and time steps for a non-linear convection-diffusion simulation. Rejected time steps are indicated with a red cross.

The performance of the classical controller and the digital filter is compared with a tolerance study of the adaptive time step control algorithm, as shown in Figure 2.6. A relatively large initial time step is used to start the simulations. The classical controller does not include a step size rejection algorithm, and is therefore not able to converge when strict tolerances are applied due to the large initial time step. The digital filter shows good computational stability for the different time integration schemes, meaning that a small change of the tolerance parameter only leads to a small change in the accuracy of the solution.

The use of the fixed resolution test and fixed scaling test has been investigated, and shown to have a negligible influence on the convergence behaviour of the different time integration schemes.

The fixed resolution test has been used for the second test case, which is a uniform flow around a cylinder, and has shown to be necessary due to the scaling of the variables. This is discussed in Section 2.7.



(a) Classical time step control without time step rejection

(b) Digital filter with the smooth limiter and time step rejection applied

Figure 2.6: Comparison of two different adaptive time step control algorithms: the classical controller, and the digital filter with the smooth limiter and step rejections. The nonlinear convection-diffusion test case is simulated with $n = m = 1$ on a non-uniform grid. The tolerance used for the adaptive time step selection is plotted versus the error of the simulations. The initial step size is chosen relatively large.

Effect of adaptive time step control on accuracy and computational efficiency with varying non linearity settings

The effect of the use of the digital filter for the time step selection on accuracy and computational efficiency is studied for two different nonlinear convection-diffusion cases: $n = m = 1$ and $n = m = 3$. Figure 2.7 shows the results for the tolerance studies of the adaptive step control algorithm, comparing the efficiency of ESDIRK and Rosenbrock. The incomplete Newton-Krylov solver, and the GMRES algorithm are used to solve the stages of the SDIRK, ESDIRK and Rosenbrock time integration schemes. The tolerance of the Newton iterations and of GMRES depends on the tolerance used for the adaptive time step selection algorithm. For the computations shown in this subsection, the tolerances of the adaptive time step controller and the appropriate solvers of the stages of ESDIRK and Rosenbrock are equal, thus $TOL_t = TOL_{NK} = TOL_{GMRES}$.

From the graphs it follows that the use of the digital filter for the adaptive time step selection results in good computational stability, i.e. a small change of the tolerance parameter TOL_t leads to a small change in the accuracy of the solution and also to a small change in the amount of computational work.

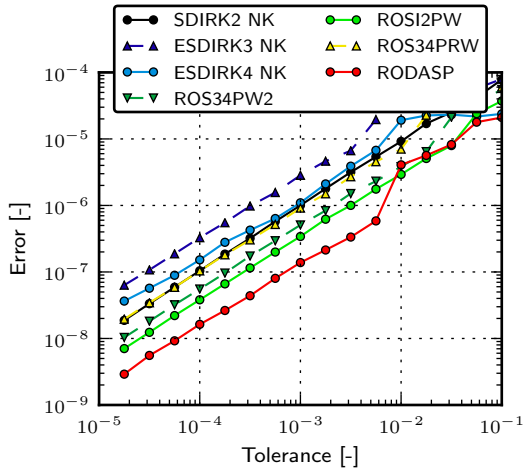
Also, comparing the convergence graphs of the different time integration schemes shows that the error of the simulation at the same tolerance differs for the different schemes. For example when RODASP is applied, the error of the solution is approximately one magnitude smaller in comparison with SDIRK2, judging from Figures 2.7(a) and 2.7(c). Therefore, a tolerance calibration procedure has been performed in order to have a better comparison of the computational efficiency of the different time integration schemes.

The results of the nonlinear convection-diffusion test case $n = m = 1$ have been used to determine the coefficients of the tolerance calibration via a least squares fit. The resulting tolerance $TOL_t{}'$ is multiplied with a factor β in order to have similar accuracies for the different time integration schemes at the end of the simulations. The used coefficients are shown in Table 2.6 with the equivalence point set to $TOL_{t0} = 10^{-5}$. The results of the tolerance study after calibration is shown in Figure 2.8.

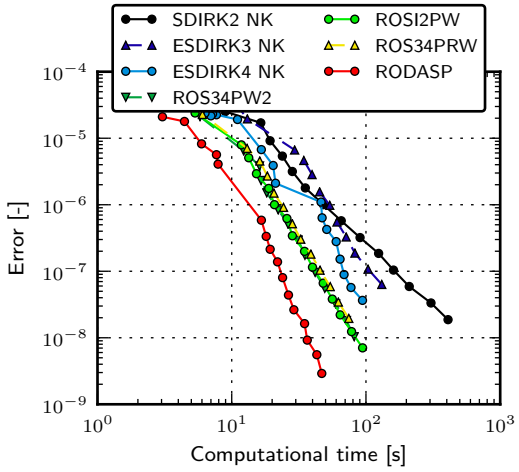
Again, the two nonlinear convection-diffusion test cases $n = m = 1$ and $n = m = 3$ are simulated. The convergence graphs for the different time integration schemes are now nearly on top of each other. Comparing the computational efficiency of ESDIRK and Rosenbrock shows a gain in efficiency for the Rosenbrock schemes. RODASP is the most computationally efficient scheme in comparison with ROS34PW2, ROSI2PW, ROS34PRW, SDIRK2, ESDIRK3 and ESDIRK4.

Table 2.6: Coefficients determined after the tolerance calibration for the nonlinear convection-diffusion case

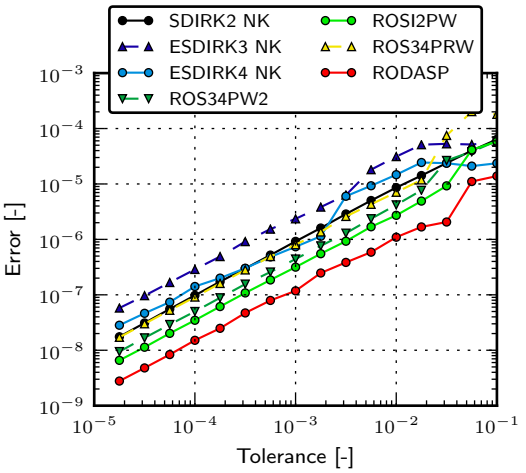
Scheme	α	β
SDIRK2	0.99	1064
ESDIRK3	0.95	423
ESDIRK4	0.90	1267
ROS34PW2	0.95	2714
ROSI2PW	0.96	3734
ROS34PRW	0.94	1468
RODASP	0.91	15039



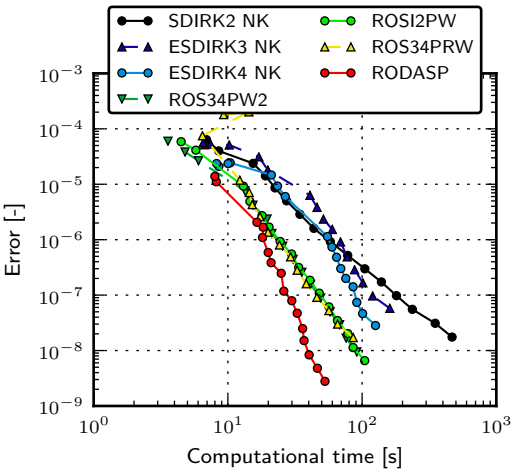
(a) Accuracy for convection-diffusion case $n = m = 1$



(b) Computational work for convection-diffusion case $n = m = 1$

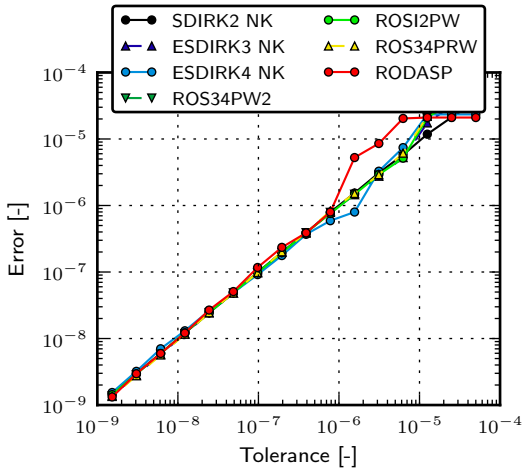


(c) Accuracy for convection-diffusion case $n = m = 3$

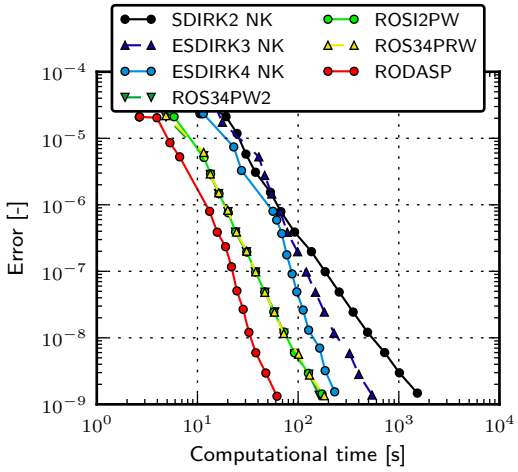


(d) Computational work for convection-diffusion case $n = m = 3$

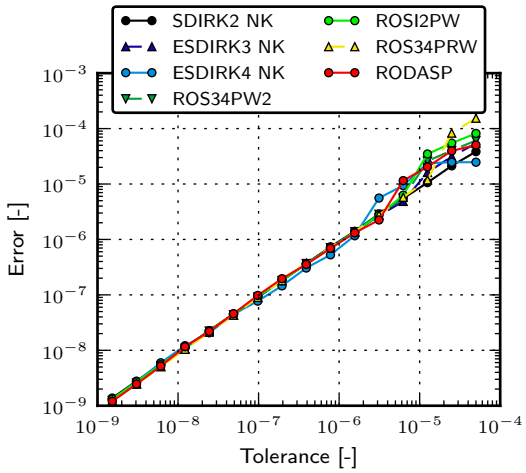
Figure 2.7: Comparison of computational efficiency and accuracy of the different time integration schemes with the digital filter used for the adaptive time step selection. Two non-linear convection-diffusion case are simulated: $n = m = 1$ and $n = m = 3$.



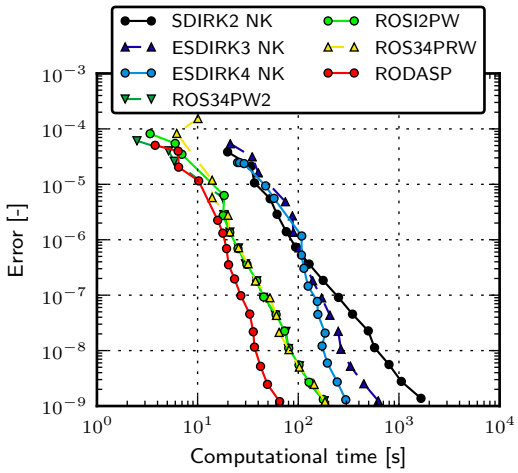
(a) Accuracy for convection-diffusion case $n = m = 1$



(b) Computational work for convection-diffusion case $n = m = 1$



(c) Accuracy for convection-diffusion case $n = m = 3$



(d) Computational work for convection-diffusion case $n = m = 3$

Figure 2.8: Tolerance scaling and calibration for two nonlinear convection-diffusion cases: $n = m = 1$ and $n = m = 3$. The digital filter combined with the smooth limiter and time step rejection is used for the adaptive step size selection. The results of the nonlinear convection-diffusion case $n = m = 1$ are used to determine the coefficients for the tolerance calibration.

2.7 Uniform flow past a circular cylinder

Following the previous nonlinear convection-diffusion test case, the question remains how the Rosenbrock time integration scheme compare to the ESDIRK scheme in terms of computational efficiency and stability when applied to viscous flows. The second test case consists of a two-dimensional flow around a cylinder. The circular cylinder is held fixed in a uniform flow field, resulting in a vortex-street behind the cylinder. When the initial transient has been disappeared, an unsteady periodic flow is present. This test case has been used by Bijl et al. (2002) and Van Zuijlen (2006) to study the order of the ESDIRK schemes for laminar and turbulent flow. The findings of this research are that the theoretical order and efficiency of the ESDIRK schemes are confirmed for laminar flows. For turbulent flows, the order of the schemes reduces, but only for a small amount at moderate Reynolds numbers. In Van Zuijlen (2006), the performance of the ESDIRK schemes is compared with the BDF2 scheme. For this test case the time integration has been performed more efficiently with the ESDIRK scheme in comparison with the second order BDF2 scheme.

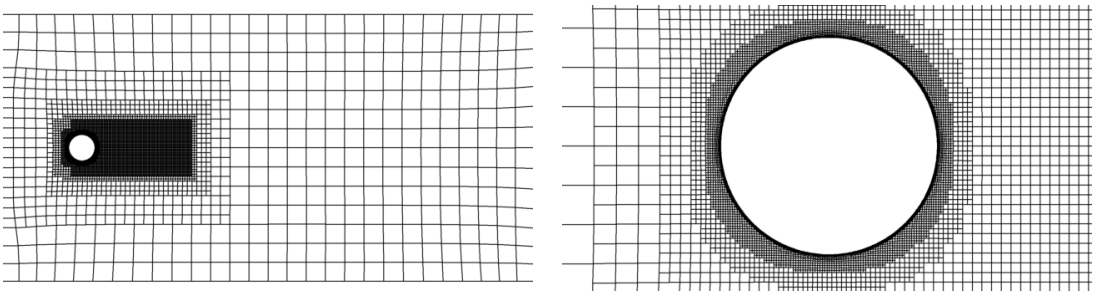
A fixed time step study and an adaptive time step study have been performed to study the performance of ESDIRK and Rosenbrock for this test case. The simulations are performed with the Reynolds-averaged Navier-Stokes solver HEXSTREAM.

2.7.1 Description of the flow past a circular cylinder

The cylinder with diameter D is located on a fixed position in a uniform flow field with Mach number $M_\infty = 0.3$ and Reynolds number $Re_\infty = 1.0 \cdot 10^3$, simulating a laminar flow. The radius of the cylinder is used as the characteristic length to determine the Reynolds number.

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 a 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 $6.6 \cdot 10^{-5}\, D$. 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 in shown in Figure 2.10.

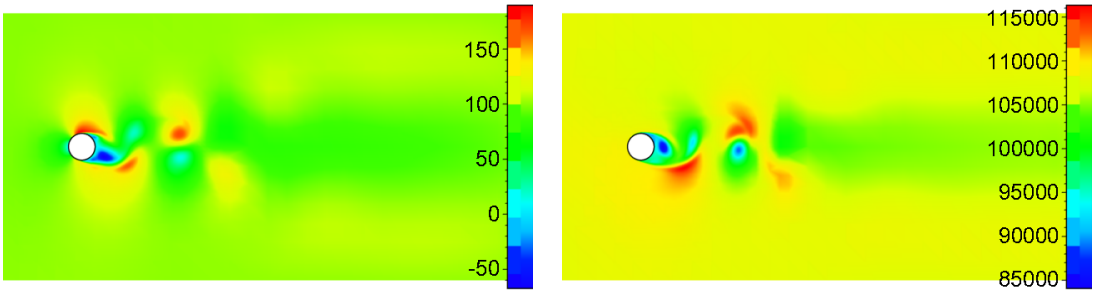
An initial solution for the convergence and tolerance studies is computed with the BDF method. During this simulation the transient from the initial uniform flow to a periodic solution is obtained. The solution at the end of this simulation is used as the initial solution for the time step convergence study, and the adaptive tolerance study. By following this approach, the transient from the initial flow condition is removed from the computations. The vertical velocity and pressure of the reference computation is shown in Figure 2.10.



(a) Computational mesh used for uniform flow around a circular cylinder test case of the complete domain

(b) Computational mesh near the cylinder

Figure 2.9: Computational mesh used for the uniform flow around a circular cylinder case



(a) Reference simulation: vertical velocity (m/s)

(b) Reference simulation: total pressure (Pa)

Figure 2.10: Vertical velocity and total pressure at the end of the simulation of the reference solution.

2.7.2 Performance of the time integration schemes for the flow past a circular cylinder

The performance of the ESDIRK and Rosenbrock time integration schemes is verified by performing a fixed time step study, and an adaptive time step study. Again, the error of one simulation is determined by taking the L_2 norm of the difference between the solution and the reference solution, and scaling with the total number of cells, as shown in Equation (2.52). The reference solution is computed with the multi grid solver combined with ESDIRK5 with time step $\Delta t = 7.8126 \cdot 10^{-6}$ until $t = 0.02$. The tolerance for the multi grid solver is set to the strict value $TOL_{NMG} = 10^{-8}$.

Effect of the Newton-Krylov method and Krylov subspace method on accuracy and efficiency

A fixed time step study has been performed for the ESDIRK and Rosenbrock time integration schemes, in order to compare the accuracy and efficiency of the used schemes. The nonlinear multi grid solver and Jacobian-free Newton-Krylov solver are used to solve the implicit stages of the ESDIRK scheme. GMRES is employed to solve the linear stages of the Rosenbrock time integration schemes. As is also the case for the nonlinear convection-diffusion case, the Eisenstat-Walker method is used to determine the forcing terms of the incomplete Newton-Krylov method, in order to avoid over solving of the linear systems arising from the Newton iterations.

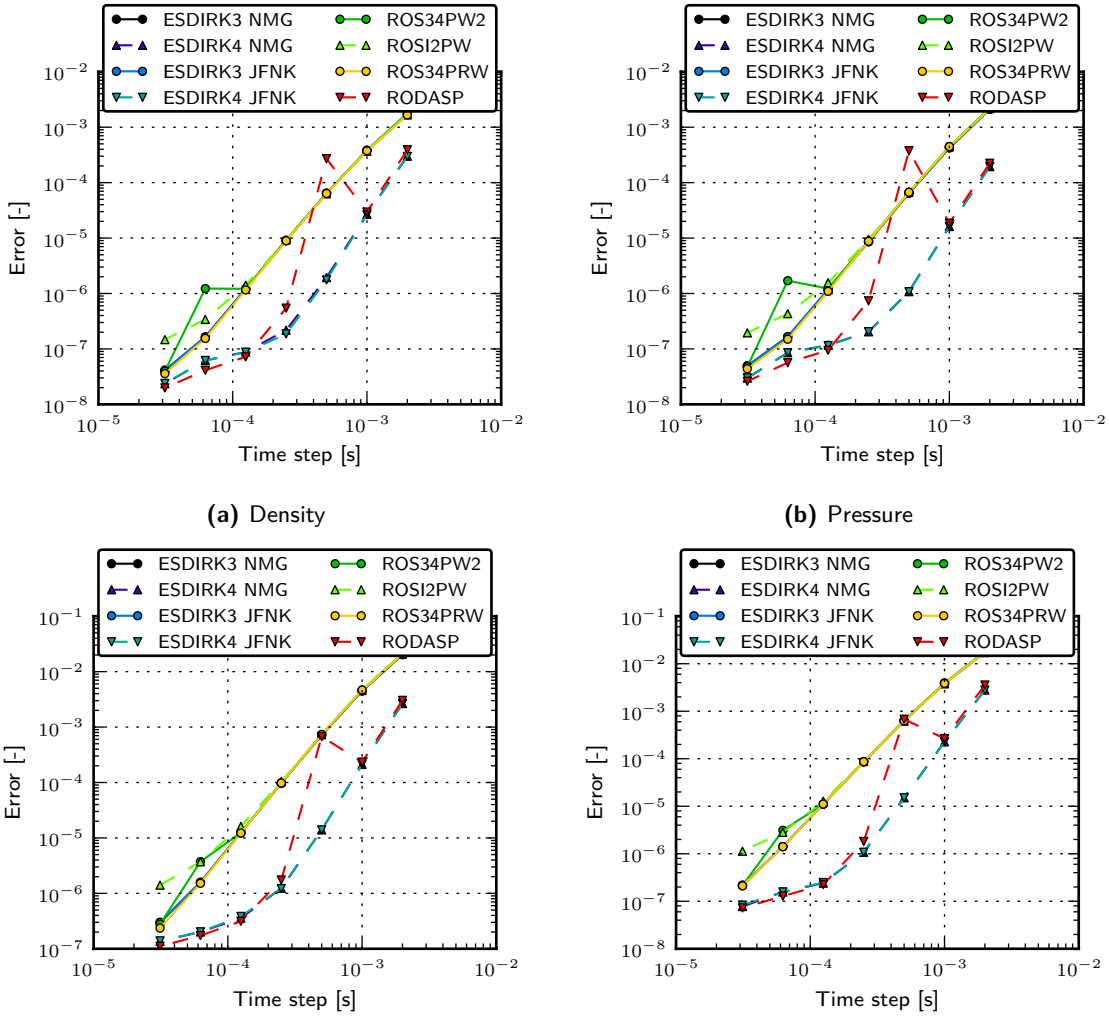
For the first time step study, the tolerance for the multi grid solver, JFNK solver and GMRES solver in case of Rosenbrock is set to the strict value $TOL_{NMG} = TOL_{NK} = TOL_{GMRES} = 10^{-6}$, resulting in a stable time integration. The results for these simulations are shown graphically in Figure 2.11. As can be seen in the four subplots of Figure 2.11, RODASP and ROS34PW2 have trouble converging for the time steps $\Delta t = 5.0 \cdot 10^{-4}$ and $\Delta t = 6.25 \cdot 10^{-5}$, respectively. An explanation for this fact, is that the linear solver is stopped before the solution is converged, since a maximum number of GMRES iterations is imposed. The convergence graph of ROS34PRW lies almost on top of the results for the ESDIRK3 scheme.

The accuracy of the other computations performed with a Rosenbrock scheme lie close to the computations performed with an ESDIRK scheme. The order of the fourth order schemes reduces for small time steps, which is probably caused by the iterative error since a Newton-Krylov or GMRES solver is used. Note that ROSI2PW is slightly less accurate for the two smallest time steps in comparison with ESDIRK3 and ROS34PW2.

Besides these observations, the accuracy of the multi grid and JFNK solvers for the ESDIRK scheme are approximately equal for this tolerance setting ($TOL_{NK} = TOL_{GMRES} = 10^{-6}$). For the tolerances $TOL_{NK} = TOL_{GMRES} = 10^{-1}$ and $TOL_{NK} = TOL_{GMRES} = 10^{-3}$, the order of the time integration schemes reduces for small time steps, as shown in Figure 2.12. From these figures it can be concluded that it is necessary to choose a relatively strict tolerance for

the different solvers. Note that the multi grid approach shows a large reduction of the order for $TOL_{NK} = TOL_{GMRES} = 10^{-1}$, compared to the Jacobian-free Newton-Krylov solver. This can be explained by the fact that one Newton iteration often reduces the residual of the implicit system several orders in magnitude. In other words, the ESDIRK stages are solved to a higher precision than the imposed tolerance setting, whereas this is generally not the case for the multi grid algorithm or the Krylov subspace solver. It is important to note that RODASP is more susceptible to order reduction in comparison with ROS34PW2, ROSI2PW and ROS34PRW.

Concluding, a gain in computational efficiency is observed for the Rosenbrock time integration schemes in comparison with ESDIRK. However, the difference in computational efficiency is not as large as for the convection-diffusion test case. The JFNK solver shows a significant increase in computational efficiency compared to the multi grid solver, confirming the work of Lucas (2010). The RODASP scheme outperforms the other time integration schemes in terms of efficiency, but is also susceptible to order reduction when the tolerance for the GMRES algorithm is chosen too large. ROS34PRW is more efficient that ESDIRK3, and is less sensitive to the GMRES tolerance setting in comparison with RODASP.



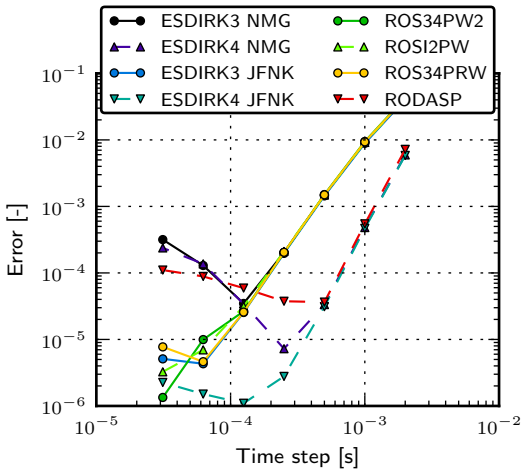
(a) Density

(b) Pressure

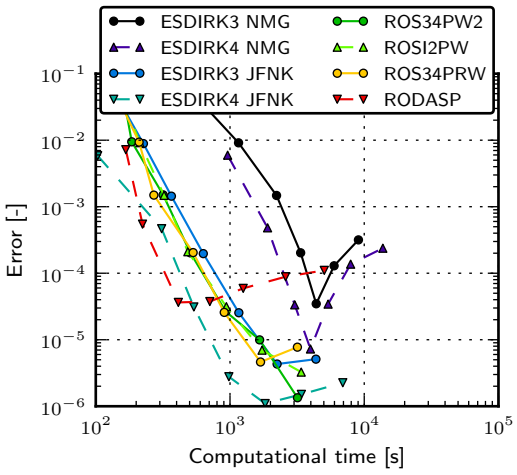
(c) Velocity component in x-direction

(d) Velocity component in y-direction

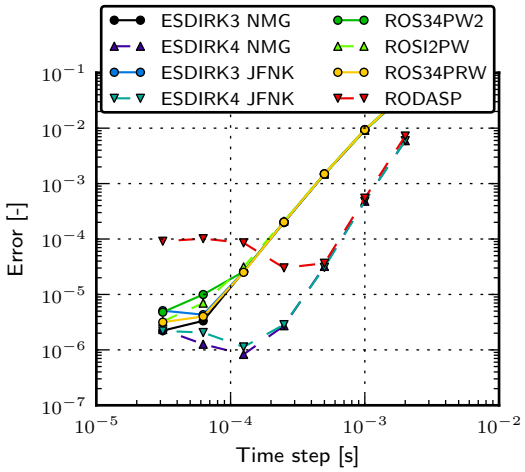
Figure 2.11: Uniform around a cylinder: fixed time step study comparing the accuracy of the ESDIRK and Rosenbrock schemes. The error of the normalised density, pressure and velocity components is shown for a range of time steps. The ESDIRK stages are solved with a nonlinear multi grid algorithm, and with a Jacobian-free Newton-Krylov solver. The stages of the Rosenbrock scheme are solved with the GMRES algorithm. The tolerance setting for these simulations is $TOL_{NK} = TOL_{GMRES} = 10^{-}6$.



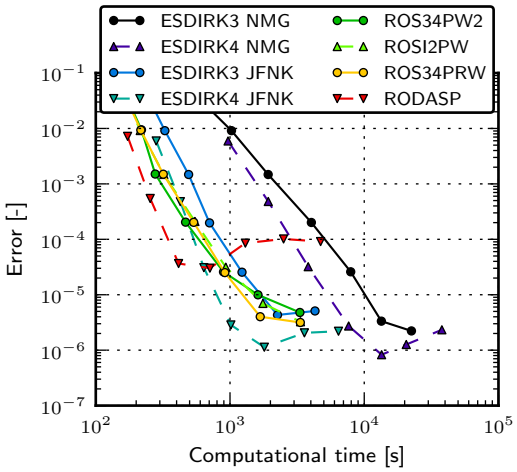
(a) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-1}$



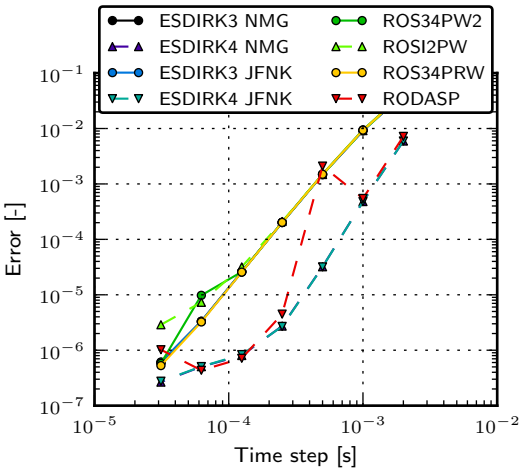
(b) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-1}$



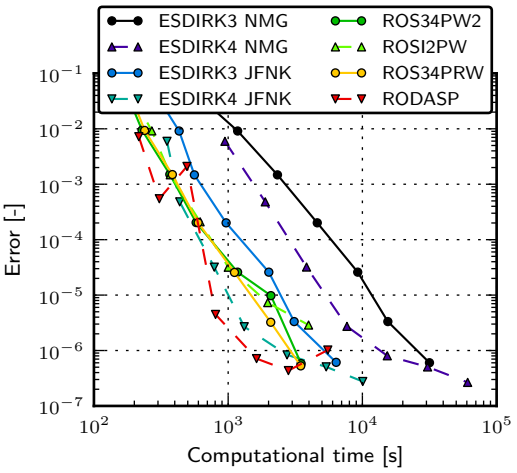
(c) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-3}$



(d) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-3}$



(e) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-5}$



(f) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-5}$

Figure 2.12: Uniform flow around a cylinder: fixed time step study comparing the accuracy and computational efficiency of ESDIRK and Rosenbrock schemes with different tolerance settings applied for the appropriate solvers. The ESDIRK stages are solved with a nonlinear multi grid algorithm (NMG), and with a Jacobian-free Newton-Krylov solver (JFNK). The stages of the Rosenbrock schemes are solved with an ILU preconditioned GMRES method. The error is determined by using the complete solution vector.

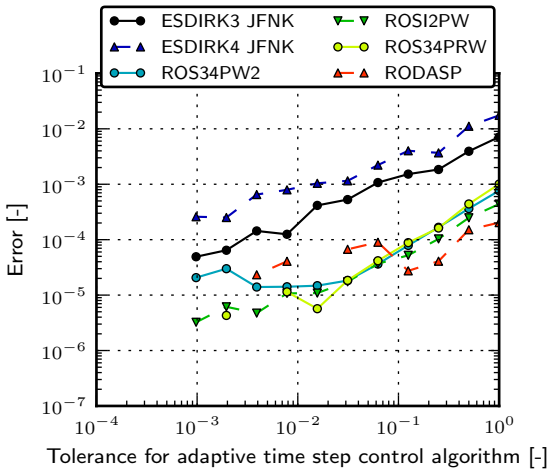
Effect of adaptive time step control on accuracy and computational stability for a uniform flow around a cylinder

The application of the digital filter for the adaptive time step selection has been tested for a range of time steps, in order to study the computational efficiency and accuracy of the Rosenbrock and ESDIRK schemes. The digital filter is combined with the smooth limiter, fixed resolution test and step size rejection. Figure 2.13 shows the results of the simulations performed with this adaptive time step control algorithm. The Jacobian-free Newton Krylov solver is used to solve the implicit stages of the ESDIRK scheme, and the GMRES method solves the linear systems of the Rosenbrock stages. The tolerance for the JFNK solver, and the GMRES algorithm is determined with $TOL_{NK} = TOL_{GMRES} = c \cdot TOL_t$, where c is $c = 10^0$, $c = 10^{-2}$ and 10^{-4}.

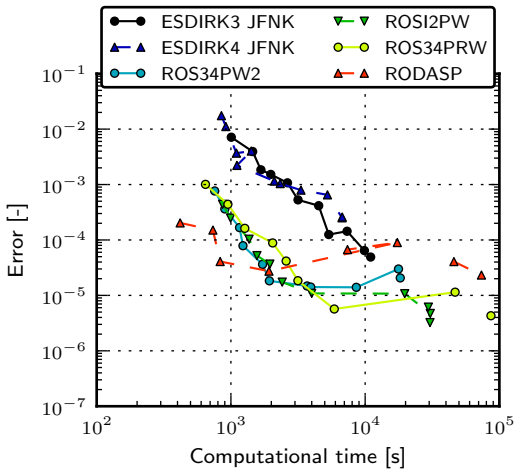
As can be seen in the different graphs, the use of an adaptive time step selection, and notably the use of the digital filter results in good computational stability for the ESDIRK and Rosenbrock solvers. For the RODASP scheme it is necessary to choose a strict tolerance for the GMRES algorithm, which is also observed for the fixed time step study. For loose tolerances, a large number of time steps are rejected, and small time steps are selected as a result. The third order Rosenbrock schemes and the third and fourth order ESDIRK schemes show good computational stability for the different tolerance settings.

Besides, the adaptive time control results in a large difference in accuracy for the same tolerance setting comparing Rosenbrock and ESDIRK. For the Rosenbrock schemes, the accuracy of the solution is more than one magnitude higher compared to ESDIRK3 and ESDIRK4. Therefore, a tolerance calibration procedure has been carried out in order to have a good comparison of the efficiency of the ESDIRK and Rosenbrock schemes.

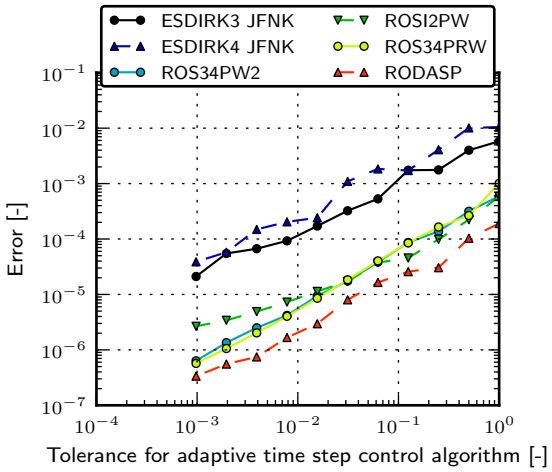
The simulations performed with the adaptive time step selection algorithm show a significant gain in efficiency for the Rosenbrock schemes in comparison with ESDIRK. The computations performed with the calibrated tolerances confirm this result, as shown in Figure 2.14. The coefficients are found via a least squares fit through the data points shown in Figure 2.13, and can be found in Table 2.7. The equivalence point of the calibration is set to $TOL_{t0} = 10^{-2}$. The convergence graphs for the different time integration schemes are now close to each other, but still a difference in accuracy is present between the different schemes at the same tolerance setting TOL_t. For this test case, the third and fourth order Rosenbrock time integration schemes show a significant gain in computational efficiency compared to ESDIRK.



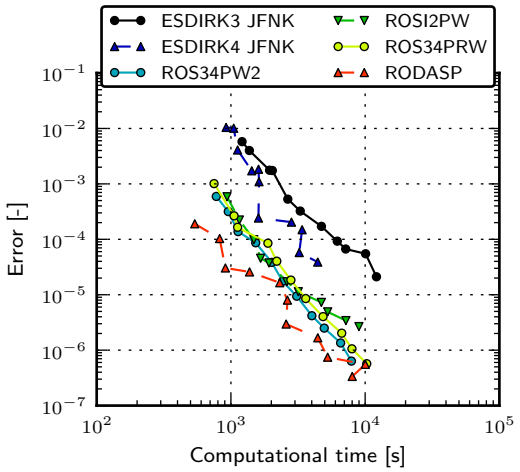
(a) Accuracy for $TOL_{NK} = TOL_{GMRES} = TOL_t$



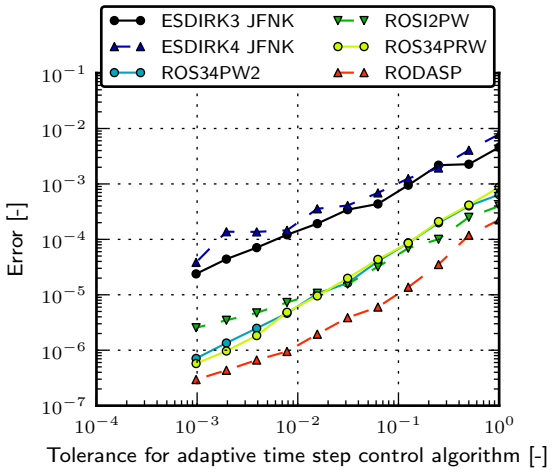
(b) Computational work for $TOL_{NK} = TOL_{GMRES} = TOL_t$



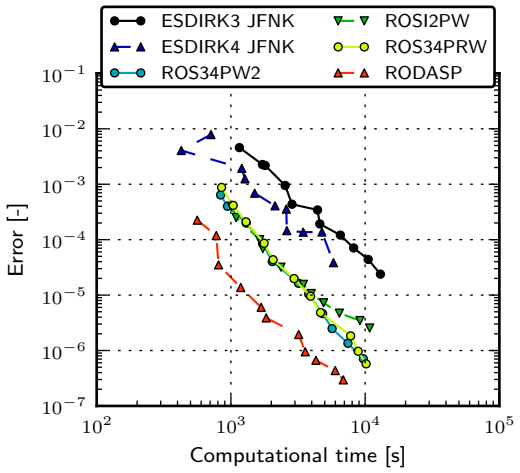
(c) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-2} \cdot TOL_t$



(d) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-2} \cdot TOL_t$

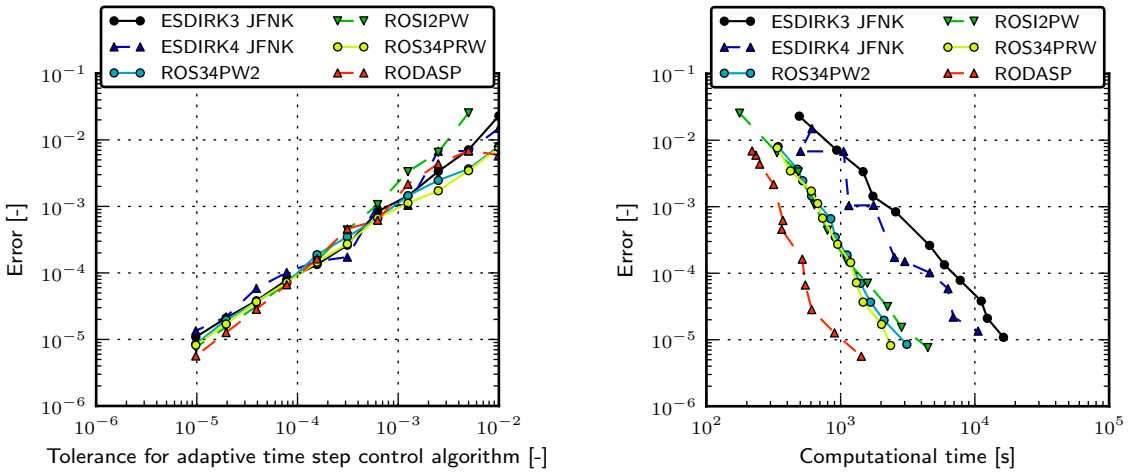


(e) Accuracy for $TOL_{NK} = TOL_{GMRES} = 10^{-4} \cdot TOL_t$



(f) Computational work for $TOL_{NK} = TOL_{GMRES} = 10^{-4} \cdot TOL_t$

Figure 2.13: Uniform around a cylinder: comparison of computational efficiency and accuracy of ESDIRK and Rosenbrock schemes with the digital filter combined with a fixed resolution test used for the adaptive time step selection. The tolerance for the JFNK solver, and the GMRES algorithm is determined with $TOL_{NK} = TOL_{GMRES} = c \cdot TOL_t$ where c is $c = 10^0$, $c = 10^{-2}$ and $c = 10^{-4}$. The error is determined by using the complete solution vector.



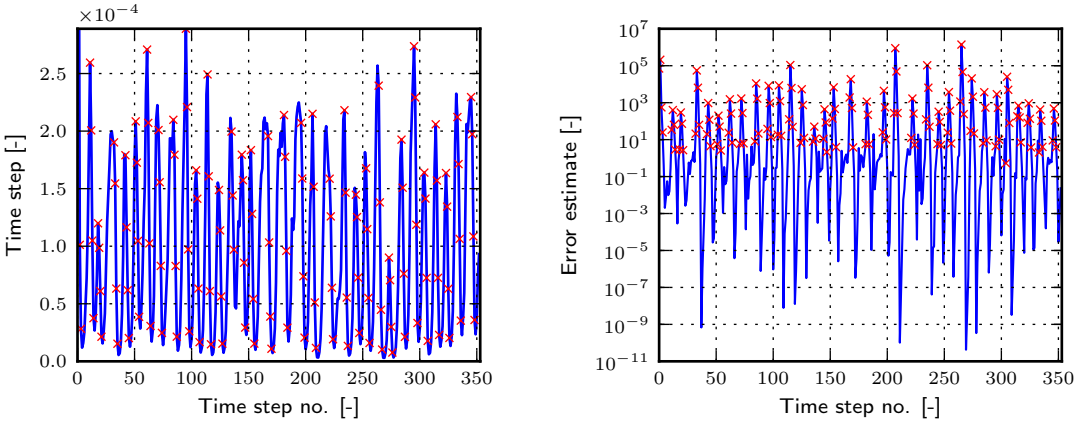
(a) Accuracy **(b)** Computational work

Figure 2.14: Uniform flow around a cylinder: comparison of performance of ESDIRK and Rosenbrock schemes with tolerance scaling and calibration. The digital filter combined with the fixed resolution test, smooth limiter, and time step rejection is used for the adaptive step size selection.

Table 2.7: Calibration coefficients for the different time integration schemes. The coefficients are determined with the results calculated for the uniform flow around a cylinder.

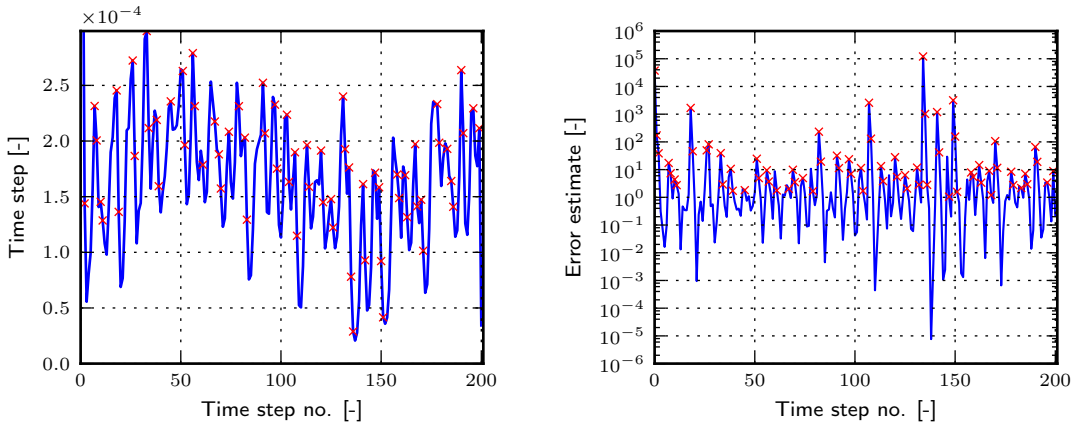
Scheme	α	β
ESDIRK3	0.75	222
ESDIRK4	0.70	171
ROS34PW2	1.01	1434
ROSI2PW	0.75	3192
ROS34PRW	1.08	1155
RODASP	0.97	6951

The time step histories and the error estimator histories for the ESDIRK3 scheme with $TOL_{NK} = 1.6 \cdot 10^{-6}$ and $TOL_t = 1.6 \cdot 10^{-2}$ and for the ROS34PW2 computation with $TOL_{GMRES} = 1.2 \cdot 10^{-5}$ and $TOL_t = 1.3 \cdot 10^{-1}$ are shown graphically in Figures 2.15 and 2.16, respectively. The first fifty time steps of both computations are shown in Figure 2.17. These two simulations have similar accuracies, thus can be used to investigate the cause of the large difference in computational efficiency between the Rosenbrock and ESDIRK time integration schemes. The time step histories show a large difference in the number of time steps between the ROS34PW2 and ESDIRK3 simulation. In case a time step is rejected for the ESDIRK3 scheme, the following two or three steps are also rejected. Thereafter, the time step is slowly increased by the digital filter, and a time step is again rejected. For ROS34PW2 less consecutive time steps are rejected.



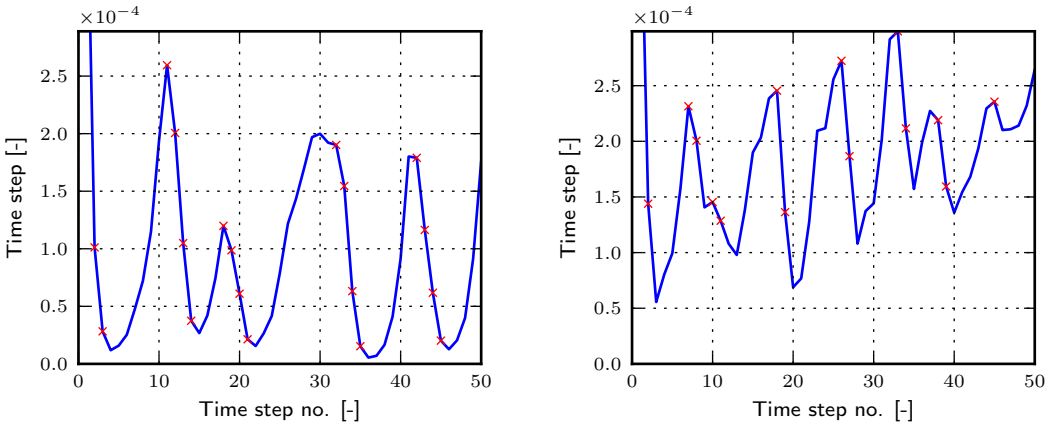
(a) Time step history for ESDIRK3 **(b)** Error estimate history for ESDIRK3

Figure 2.15: Time step history and error estimator history for the ESDIRK3 scheme with $TOL_{NK} = 1.6 \cdot 10^{-6}$ and $TOL_t = 1.6 \cdot 10^{-2}$. Rejected time steps are indicated with red crosses.



(a) Time step history for ROS34PW2 **(b)** Error estimate history for ROS34PW2

Figure 2.16: Time step history and error estimator history for the ROS34PW2 scheme with $TOL_{GMRES} = 1.2 \cdot 10^{-5}$ and $TOL_t = 1.3 \cdot 10^{-1}$.



(a) Time step history for ESDIRK3 **(b)** Time step history for ROS34PW2

Figure 2.17: The first fifty time steps for the ESDIRK3 scheme with with $TOL_{NK} = 1.6 \cdot 10^{-6}$ and $TOL_t = 1.6 \cdot 10^{-2}$, and the ROS34PW2 scheme with $TOL_{GMRES} = 1.2 \cdot 10^{-5}$ and $TOL_t = 1.3 \cdot 10^{-1}$.

2.8 Summary

Implicit Runge-Kutta and Rosenbrock time integration schemes are considered for the use in a flow solver, and possibly also in a fluid-structure interaction solver. Implicit Runge-Kutta methods are preferred over explicit RK methods, since large differences in length and time scales are present when fluid flows are considered. Therefore, the time step is not constrained by stability requirements, but rather by accuracy considerations. For third or higher order accuracy, ESDIRK and Rosenbrock schemes are considered. ESDIRK schemes have advantageous properties, namely for arbitrary high order the method remains L-stable, also s coupled systems of equations need to be solved, versus $s+1$ coupled systems for the IRK, DIRK and SDIRK methods.

Rosenbrock schemes are derived from a DIRK method by following a linearisation procedure. Therefore, stability and accuracy properties of the DIRK method are lost after the linearisation, and a smaller time step is required for Rosenbrock methods compared to ESDIRK. However, Rosenbrock-Wanner methods, which use an approximation for the Jacobian, have less computational costs per time step. Computational costs can further be reduced by reusing the preconditioner for the s stages of the Rosenbrock scheme.

Adaptive time step control algorithms have been discussed shortly. By the using the embedded methods of the Runge-Kutta or Rosenbrock schemes, the error of the solution can be efficiently approximated. The use of a digital filter and limiter ensures computational stability, i.e. a small change of the tolerance parameter only leads to a small change in the accuracy of the solution, as well as on the work required to achieve that accuracy.

A nonlinear convection-diffusion equation is used to compare the performance of the Rosenbrock time integration schemes with the implicit Runge-Kutta schemes. Also, a uniform flow around a circular cylinder is simulated with a RANS solver in order to compare the computational efficiency and accuracy of the Rosenbrock and ESDIRK schemes. Based on the observed order and efficiency of the different time integration schemes, it is believed that the schemes are correctly implemented. A gain in efficiency is observed for the Rosenbrock time integration when fixed and adaptive time steps are used to integrate the fluid dynamics equations, or the nonlinear convection-diffusion equation. A large gain in computational efficiency is observed when adaptive time steps are used. The difference in efficiency is explained by the fact that the Rosenbrock schemes use less time steps to solve the governing equations.

The RODASP scheme outperforms the ESDIRK schemes in terms of computational efficiency, but is also susceptible to order reduction when the tolerance TOL_{GMRES} is chosen rather large. The ROS34PRW scheme is also computationally more efficient in comparison with ESDIRK3, and also has better stability properties in comparison with RODASP.

Combining an adaptive time step selection algorithm with the Rosenbrock time integration schemes is a possible solution for problems caused by instability of the time integration scheme.

Also, inaccuracies arising from slow convergence of the GMRES algorithm can be solved with an adaptive time selection combined with step size rejection. For the test cases studied in this chapter, the performance of the ESDIRK scheme is reduced in terms of computational time when an adaptive time step selection algorithm is used, since a relatively large number of time steps are rejected. Future research can focus on optimising the parameter κ of the adaptive step control algorithm, in order to decrease the number of consecutive rejected time steps for the ESDIRK time integration scheme.

Chapter 3

Krylov subspace enrichment

Several time integration methods have been discussed and studied in the previous chapter. Inexact Newton methods and Rosenbrock-Wanner methods employ iterative methods to solve the linear schemes arising from the two different linearisation steps. The GMRES algorithm which is used in this thesis to solve the linear systems is extensively discussed in this chapter. The chapter starts with some introductory remarks introducing Krylov subspace methods. Section 3.2 discusses the main principles of Krylov subspace methods. Thereafter, restarted and truncated, and augmented and deflated methods are the topics of Sections 3.3 and 3.4, respectively. The GMRES-E algorithm, which reuses Krylov subspace vectors on consecutive GMRES cycles, is discussed in Section 3.5, and thereafter the results of the simulations for the different test cases are reported. The chapter concludes with a short summary explaining the main findings of the performed research.

3.1 Introductory remarks

Krylov subspace methods are being applied for the iterative solution of linear systems of equations of the form

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}. \tag{3.1}$$

Krylov subspace methods are extensively used to solve the systems arising from ordinary and partial differential equations, including the Navier-Stokes equations. Several Krylov subspace methods have been developed and extended during the last two decades. These new developments have led to different versions of augmented, restarted, flexible, deflated, nested and inexact methods. Blom (2012) gives an overview of these methods, indicating the applicability of the different methods. The main result of this literature survey was that the GMRES algorithm is currently

the best approach to solve the linear systems arising from the Newton iterations or the stages of a Rosenbrock scheme. The content of this chapter is partly based on this literature survey.

The recently published book by Elman et al. (2005) gives an introduction into iterative linear solvers, and the application of the solvers with finite elements. Saad (2003) gives a thorough overview of iterative methods, covering also preconditioning techniques. Another reference for matrix iterative analysis can be found in Varga (2000). The recent literature survey Simoncini and Szyld (2007) is also a good reference for recent developments of Krylov subspace methods.

Good references are Saad (2003), Kelley (1995), Varga (2000), Greenbaum (1987) and Van der Vorst (2003).

3.2 Krylov subspace methods

Krylov subspace methods build a solution for the linear system of equations given by Equation (3.1). $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\,\mathbf{x}_0$ is the initial residual, and let $\mathcal{K}_m$ be the Krylov subspace with dimension m defined by

$$\mathcal{K}_m\left(\mathbf{A}, \mathbf{r}_0\right) = \text{span}\left\{\mathbf{r}_0, \mathbf{A}\,\mathbf{r}_0, \mathbf{A}^2\,\mathbf{r}_0, \ldots, \mathbf{A}^{m-1}\,\mathbf{r}_0\right\}. \tag{3.2}$$

The m-th step of the iterative method provides the approximation $\mathbf{x}_m$ in $\mathbf{x}_0 + \mathcal{K}_m$. The Krylov subspaces are nested, that is $\mathcal{K}_m \in \mathcal{K}_{m+1}$. The approximation is of the form $\mathbf{x}_m = \mathbf{x}_0 + q_{m-1}\left(\mathbf{A}\right)\mathbf{r}_0$, where q_{m-1} is a polynomial of degree $m-1$.

The residual $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\,\mathbf{x}_m$ is associated with the *residual polynomial* p_m with $p_m\left(0\right) = 1$, as shown in

$$\mathbf{r}_m = \mathbf{b} - \mathbf{A}\,\mathbf{x}_m = \mathbf{r}_0 - \mathbf{A}\,q_{m-1}\left(\mathbf{A}\right)\mathbf{r}_0 = p_m\left(\mathbf{A}\right)\mathbf{r}_0. \tag{3.3}$$

The set of polynomials p of degree at most m such that $p\left(0\right) = 0$ is denoted by $\mathcal{P}_m$. The approximation $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m$ can be found by requiring $\mathbf{x}_m$ to be the minimizer of a certain functional. The GMRES method (Saad and Schultz, 1986) minimizes the 2-norm of the residual, which is discussed in Section 3.2.2.

The procedure starts with the initial vector $\mathbf{x}_0$, the initial residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\,\mathbf{x}_0$, and at the m-th step the vector $\mathbf{x}_m$ in $\mathbf{x}_0 + \mathcal{K}_m\left(\mathbf{A}, \mathbf{r}_0\right)$ is obtained which satisfies a projection or minimizing condition. $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\,\mathbf{x}_m$ is the residual at the m-th iteration step. The *Petrov-Galerkin* condition is defined as follows:

$$\mathbf{r}_m \perp \mathcal{L}_m, \tag{3.4}$$

where $\mathcal{L}_m$ is a m-dimensional subspace. The *Galerkin condition* and the *minimum residual condition* are two Petrov-Galerkin conditions for $\mathcal{L}_m = \mathcal{K}_m$ and $\mathcal{L}_m = \mathbf{A}\,\mathcal{K}_m$, respectively. Thus the

following relation applies for a Galerkin condition:

$$\mathbf{r}_m \perp \mathcal{K}_m. \tag{3.5}$$

The minimum residual condition is defined as shown in the following equation:

$$||\mathbf{r}_m|| = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} ||\,\mathbf{b} - \mathbf{A}\,\mathbf{x}\,||\,. \tag{3.6}$$

Note that any method for which the Petrov-Galerkin, Galerkin or minimum residual condition holds, the method will stop in at most n steps, in exact arithmetic. This is a result of the nested property of the Krylov subspaces. For practical implementations this result has little value, since the storage and computational requirements are prohibitive.

3.2.1 Arnoldi procedure

The Arnoldi procedure constructs an orthonormal basis of the Krylov subspace (Simoncini and Szyld, 2007). For symmetric matrices, the Arnoldi procedure is simplified and is named after Lanczos (Lanczos, 1950).

An orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\}$ of the Krylov subspace $\mathcal{K}_m\,(\mathbf{A}, \mathbf{r}_0)$ is determined by computing one vector during every iteration by evaluating $\mathbf{A}\,\mathbf{v}_k$. This new vector is orthogonalised with respect to the previous vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$, and is also normalised. Therefore,

$$\mathbf{v}_{k+1}\,h_{k+1,k} = \mathbf{A}\,\mathbf{v}_k - \sum_{j=1}^{k} \mathbf{v}_j\,h_{jk}, \tag{3.7}$$

where the coefficients $h_{jk} = \langle \mathbf{v}_j, \mathbf{A}\,\mathbf{v}_k \rangle\,, j \leqslant k$ are constructed such that orthogonality is achieved. $h_{k+1,k}$ is positive, and the new vector $\mathbf{v}_{k+1}$ has the property such that $||\mathbf{v}_{k+1}|| = 1$. These vectors are collected in the matrix $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m]$. The coefficients h_{jk} are collected into the upper Hessenberg matrix $\mathbf{H}_{m+1,m}$ with dimension $(m+1) \times m$. Thus the Arnoldi relation is found:

$$\mathbf{A}\,\mathbf{V}_m = \mathbf{V}_{m+1}\,\mathbf{H}_{m+1,m} = \mathbf{V}_m\,\mathbf{H}_m + h_{m+1,m}\,\mathbf{v}_{m+1}\,{\mathbf{e}_m}^T, \tag{3.8}$$

where ${\mathbf{e}_m}^T$ is the k-th canonical basis vector in $\mathcal{R}^k$. The matrix $\mathbf{H}_m$ contains the first m rows of the upper Hessenberg matrix $\mathbf{H}_{m+1,m}$, which can be written in mathematical terms as

$$\mathbf{H}_{m+1,m} = \begin{bmatrix} \mathbf{H}_m \\ h_{m+1,m}\,{\mathbf{e}_m}^T \end{bmatrix}. \tag{3.9}$$

From Equation (3.8) it follows that the rank of the Hessenberg matrix $\mathbf{H}_{m+1,m}$ is equal to the rank of $\mathbf{A}\,\mathbf{V}_m$. In other words, the upper Hessenberg matrix $\mathbf{H}_{m+1,m}$ has rank m if the new Krylov subspace vector $\mathbf{A}\,\mathbf{V}_m$ is linearly independent of the previous Krylov subspace vectors. Also note

that for the case that $h_{m+1,m} = 0$, the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_m$ form an invariant subspace of the matrix $\mathbf{A}$, thus the solution of Equation (3.1) belongs to this subspace. Also, from Equation (3.8) one obtains

$$\mathbf{V}_m{}^T \mathbf{A} \mathbf{V}_m = \mathbf{H}_m. \tag{3.10}$$

It is important to note that in this description of the Arnoldi procedure, the standard Gram-Schmidt orthogonalisation procedure is applied. In actual implementations, it is advised to apply the modified Gram-Schmidt (MGS) procedure, since this approach is more stable (Golub and van Loan, 1996). Paige et al. (2006) show that the modified Gram-Schmidt GMRES (MGS-GMRES) method is backward stable.

The pseudo code for the Arnoldi-Modified Gram-Schmidt procedure is shown in Algorithm 3.1. In exact arithmetic, the Arnoldi method and the Modified Gram-Schmidt procedure and mathematically equivalent. However, in the presence of round-off the modified Gram-Schmidt procedure is more reliable (Saad, 2003).

Algorithm 3.1 Arnoldi-Modified Gram-Schmidt procedure (Saad, 2003)

Choose a vector v_1 of norm 1
for $j = 1 \rightarrow m$ **do**
    Compute $\mathbf{w}_j = \mathbf{A}\,\mathbf{v}_j$
    for $i = 1 \rightarrow j$ **do**
        $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$
        $\mathbf{w}_j = \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$
    end for
    $h_{j+1,j} = ||\mathbf{w}_j||_2$
    if $h_{j+1,j} = 0$ **then**
        Stop
    end if
    $\mathbf{v}_{j+1} = \dfrac{\mathbf{w}_j}{h_{j+1,j}}$
end for

Saad (2003) discusses two further improvements of the Arnoldi procedure. The first improvement applies double orthogonalisation. When the final vector $\mathbf{w}_j$ is obtained at the end of the main loop in Algorithm 3.1, the norm of this vector is compared with the initial vector $\mathbf{w}_j$, which equals to $||\,\mathbf{A}\,\mathbf{v}_j\,||_2$. A second orthogonalisation is made in case the reduction falls below a beforehand defined threshold.

The second improvement has originally been proposed by Walker (1988). From a numerical point of view, the Householder algorithm has proven to be one of the most reliable orthogonalisation techniques. The Householder algorithm uses reflection matrices of the form $\mathbf{P}_k = \mathbf{I} - 2\,\mathbf{w}_k\,\mathbf{w}_k{}^T$ to transform a matrix into upper triangular form. An orthogonal column $\mathbf{v}_i$ is obtained as $\mathbf{P}_1\,\mathbf{P}_2 ... \mathbf{P}_i\,\mathbf{e}_i$. $\mathbf{P}_1$, $\mathbf{P}_2$, ..., $\mathbf{P}_i$ are the previous Householder matrices. The

resulting vector is multiplied by the matrix $\mathbf{A}$. Thereafter the previous Householder transforms are applied to this result. The next Householder transform is calculated from the resulting vector.

3.2.2 GMRES

The generalised minimal residual (GMRES) method is a Krylov subspace method proposed by Saad and Schultz (1986). The projection condition (3.6) is applied, thus minimizing the residual over all available vectors in the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. Thus, $\mathbf{x}_m$ is found such that

$$||\mathbf{r}_m|| = ||\,\mathbf{b} - \mathbf{A}\,\mathbf{x}_m\,|| = \min_{\mathbf{x}\,\in\,\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)} ||\,\mathbf{b} - \mathbf{A}\,\mathbf{x}\,||\,. \tag{3.11}$$

Usually, the 2-norm is used for the GMRES method. In the literature other norms for the minimisation are also proposed (Simoncini and Szyld, 2007). It is important to note that the solution of the least squares problem (3.11) is unique when $\mathbf{A}$ has full rank (Björck, 1996).

GMRES implements the solution of the least squares problem shown in (3.11) by using a orthonormal basis of the Krylov subspace, which is produced by the Arnoldi process. Thus the approximation $\mathbf{x}_m$ is given by

$$\mathbf{x}_m = \mathbf{V}_m\,\mathbf{y}_m, \tag{3.12}$$

where $\mathbf{y}_m \in \mathcal{R}^m$. By applying Equation (3.8), and with the relation $\mathbf{V}_{m+1}\,\mathbf{e}_1 = \mathbf{v}_1 = \frac{\mathbf{b}}{\beta}$, one obtains

$$\begin{aligned} \mathbf{r}_m &= \mathbf{b} - \mathbf{A}\,\mathbf{x}_m = \mathbf{b} - \mathbf{A}\,\mathbf{V}_m\,\mathbf{y}_m \\ &= \beta\,\mathbf{v}_1 - \mathbf{V}_{m+1}\,\mathbf{H}_{m+1,m}\,\mathbf{y}_m = \mathbf{V}_{m+1}\,(\beta\,\mathbf{e}_1 - \mathbf{H}_{m+1,m}\,\mathbf{y}_m)\,, \end{aligned} \tag{3.13}$$

which can be written as

$$||\mathbf{r}_m|| = \min_{\mathbf{y}\,\in\,\mathcal{R}^m} ||\,\beta\,\mathbf{e}_1 - \mathbf{H}_{m+1,m}\,\mathbf{y}\,||\,, \tag{3.14}$$

since V_{m+1} has orthonormal columns. For the matrix $\mathbf{H}_{m+1,m}$, the following QR decomposition is used:

$$\mathbf{H}_{m+1,m} = \mathbf{Q}_{m+1}\,\mathbf{R}_{m+1,m}. \tag{3.15}$$

The matrix $\mathbf{Q}_{m+1}$ has dimensions $(m+1)\times(m+1)$ and is orthogonal. Also

$$\mathbf{R}_{m+1,m} = \begin{bmatrix} \mathbf{R}_m \\ 0 \end{bmatrix}, \tag{3.16}$$

where the matrix $\mathbf{R}_m$ is upper triangular with dimensions $m \times m$. A common implementation of the QR decomposition shown in Equation (3.15) is with Given rotations such that only two entries per step need to be computed and used to update the matrix $\mathbf{R}_m$ (Saad and Schultz, 1986; Saad, 2003). As a result of the QR decomposition the least squares problem shown in Equation (3.14)

can be written as

$$||\mathbf{r}_m|| = \min_{\mathbf{y} \in \mathcal{R}^m} \left|\left| \mathbf{Q}_{m+1}^T \beta \, \mathbf{e}_1 - \mathbf{R}_{m+1,m} \, \mathbf{y} \right|\right|. \tag{3.17}$$

Now, introduce

$$\mathbf{g}_m = \mathbf{Q}_m \, \beta \, \mathbf{e}_1 = (\gamma_1, \gamma_2, ..., \gamma_m, \gamma_{m+1})^T, \tag{3.18}$$

then it follows from (3.17) that

$$||\mathbf{r}_m|| = \left|\left| \mathbf{Q}_{m+1}^T \beta \, \mathbf{e}_1 - \mathbf{R}_{m+1,m} \, \mathbf{y}_m \right|\right| = |\gamma_{m+1}| \tag{3.19}$$

(Saad, 2003). In the literature, γ_{m+1} is also denoted as ρ_{m+1} (Simoncini and Szyld, 2007). Implementations of the GMRES method use this result to check whether convergence of the solution has been reached. It should be noted that Greenbaum (1997) has shown that the equality $|\gamma_{m+1}| = ||\mathbf{b} - \mathbf{A} \, \mathbf{x}_m||$ may not hold in finite precision arithmetic.

For all methods which satisfy the minimum residual condition on nested subspaces, shown in Equation (3.6), the sequence of residual norms $||\mathbf{r}_m||$ is non-increasing. This is also the case for the GMRES method. A disadvantage of the GMRES method is that is has long recurrences. As the iterations proceed, the storage requirements grow due to the fact that the whole basis of the Krylov subspace is needed for subsequent iterations. Truncated and restarted methods have been proposed in the literature which provide solutions for this behaviour of the GMRES method. These methods are discussed in Section 3.3.

The pseudo code for the GMRES method is shown in Algorithm 3.2, which includes left and right preconditioning via the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$. The only possibility of breakdown in the GMRES method is in the Arnoldi loop. When $\mathbf{w}_j = 0$, or when $h_{m+1,m} = 0$ during iteration step j. For this situation, the procedure stops because the next Arnoldi vector cannot be generated. However, as Saad (2003) shows, for a non singular matrix $\mathbf{A}$, break down occurs for the GMRES algorithm breaks only at step m, which means that $h_{m+1,m} = 0$, if and only if the approximate solution $\mathbf{x}_m$ is exact.

For further information on the GMRES method, see references Saad and Schultz (1986), Saad (2003) and Barrett et al. (1994). Saad (2003) also contains detailed information on preconditioning techniques, which can be used to increase the computational efficiency of the GMRES algorithm.

3.3 Restarted and truncated methods

Krylov subspace methods based on the Arnoldi procedure for non-symmetric matrices are generally expensive. For an accurate solution a large number of iterations may be necessary. As a consequence, the Arnoldi matrix $\mathbf{V}_m$ becomes too large to be stored. The standard procedure consists of restarting the Krylov subspace method when a maximum subspace dimension is reached. After

Algorithm 3.2 Modified Gram-Schmidt GMRES procedure with left and right preconditioning applied (Saad, 2003)

1: Compute $\mathbf{r}_0 = \mathbf{M_1}^{-1}(\mathbf{b} - \mathbf{A}\,\mathbf{x}_0)$, $\beta = ||\mathbf{r}_0||_2$, and $\mathbf{v}_1 = \dfrac{\mathbf{v}_0}{\beta}$
2: **for** $j = 1 \rightarrow m$ **do**
3: Compute $\mathbf{w}_j = \mathbf{M_1}^{-1}\,\mathbf{A}\,\mathbf{M_2}^{-1}\,\mathbf{v}_j$
4: **for** $i = 1 \rightarrow j$ **do**
5: $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$
6: $\mathbf{w}_j = \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$
7: **end for**
8: $h_{j+1,j} = ||\mathbf{w}_j||_2$
9: **if** $h_{j+1,j} = 0$ **then**
10: Set $m = j$ and go to 14
11: **end if**
12: $\mathbf{v}_{j+1} = \dfrac{\mathbf{w}_j}{h_{j+1,j}}$
13: **end for**
14: Define the $(m+1) \times m$ Hessenberg matrix $\overline{\mathbf{H}}_m = \{h_{i,j}\}_{1 \leqslant i \leqslant m+1, 1 \leqslant j \leqslant m}$.
15: Compute $\mathbf{y}_m = \underset{\mathbf{y} \in \mathcal{R}^m}{\min} ||\,\beta\,\mathbf{e}_1 - \mathbf{H}_{m+1,m}\,\mathbf{y}\,||_2$ and $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{M_2}^{-1}\,\mathbf{V}_m\,\mathbf{y}_m$.

m iterations the procedure is stopped, and the current approximation $\mathbf{x}_m$ is used as an initial approximation. The overall procedure of the restarted methods for a maximum 'maxit' number of restarts is shown in Algorithm 3.3.

Algorithm 3.3 Restarted Krylov subspace method (Simoncini and Szyld, 2007)

1: Given $\mathbf{A}$, ${\mathbf{x}_0}^i$, $\mathbf{b}$, m, maxit
2: **while** $i <$ maxit **do**
3: Run m iterations of the chosen Krylov subspace method and get ${\mathbf{x}_m}^i$
4: Test $||{\mathbf{r}_m}^i|| = ||\,\mathbf{b} - \mathbf{A}\,{\mathbf{x}_m}^i\,||$. If convergence reached then stop
5: Set ${\mathbf{x}_0}^{i+1} = {\mathbf{x}_m}^i$, $i = i + 1$
6: **end while**

The advantage of restarted methods is that at most m iterations of the Arnoldi procedure are performed. As a result, both the computational costs and memory allocations per outer iteration are controlled. A disadvantage of the restarted method is that the optimality properties are lost after the first restart, as is the case for GMRES. The overall process may not converge and stagnate with $||{\mathbf{r}_m}^{i+1}|| \approx ||{\mathbf{r}_m}^i||$ for all i. It should be noted that in GMRES the residual cannot increase in the outer iterations, since the inner GMRES step ensures that $||{\mathbf{r}_m}^i|| \leqslant ||{\mathbf{r}_0}^i||$ for all i. If stagnation occurs, a simple solution would be to enlarge the maximum number of inner GMRES iterations. It should be stressed that enlarging the subspace dimension m does not always ensure faster convergence (Eiermann et al., 2000). Also, for large problems it may not be

possible to choose a larger m, since the used dimension of the Krylov subspace is usually chosen as the maximum dimension affordable. Dynamically selecting the number of inner GMRES iterations may fix this problem (Joubert, 1994; Baker et al., 2009).

When restarting is applied, the GMRES residual is probably not the most efficient vector to carry over to the new restart of the inner Krylov subspace method. The residual ${\mathbf{r}_m}^i$ is a linear combination of the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_m$ which form the basis of the Krylov subspace $\mathcal{K}_m$. If the residual ${\mathbf{r}_m}^i$ is almost a multiple of $\mathbf{v}_1$, then at the following restart the starting vector ${\mathbf{r}_0}^{i+1} \equiv {\mathbf{r}_m}^i$ builds a Krylov subspace that is close to the previously build Krylov subspace. A natural strategy to ensure that the subspace generate after restarting has maximum dimension is to impose that the new direction vector has a non-negligible component onto $\mathbf{v}_{m+1}$. This requirement is actually satisfied by the Full Orthogonalisation Method (FOM), since the residual ${\mathbf{r}_m}^i$ is a multiple of $\mathbf{v}_{m+1}$.

Another approach to enrich the information carried over during the restart is to discard the oldest vectors of the Krylov subspace basis. Thus, only the last j vectors of the basis are kept orthogonal to each other. Then the recurrence is modified as follows:

$$h_{k+1,k}\,\mathbf{v}_{k+1} = \mathbf{A}\,\mathbf{v}_k - \sum_{i\,=\,\max\{1,k-j+1\}}^{k} h_{ik}\,\mathbf{v}_i. \tag{3.20}$$

The basis for the Krylov subspace is updated after the first j steps, and after each iteration the oldest vectors is replaced by last computed vector. Hence, the Arnoldi procedure is truncated. The question remains whether keeping the latest basis vectors is a good choice. Augmented and deflated methods discussed in the following section provide strategies to select 'better' vectors for the basis of the Krylov subspace.

3.4 Augmented and deflated methods

Augmented and deflated methods have been developed to reduce the computational cost of solving a linear system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$. The main idea of augmented and deflated methods is to determine an approximation space of dimension m as the sum of two spaces with a smaller dimension, thus the approximation space is spanned by the basis $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{m-k}\}$. The standard Arnoldi procedure is used to determine the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_k$, and the remaining vectors $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_{m-k}$ contain information saved from previous iterations, or are chosen beforehand as is the case for the GMRES-E algorithm (Carpenter et al., 2010).

The algorithms show differences in the manner the vectors $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_{m-k}$ are determined, and also in the way the vectors are included in the approximation. Eiermann et al. (2000) perform an analysis of the different techniques proposed in the literature. Note that Eiermann concludes with the statement that the presented techniques cannot replace an effective preconditioning

strategy. However, the algorithms can dramatically improve the performance of restarted GMRES when the method is applied to a properly preconditioned linear system.

Morgan (1995) has proposed to compute the vectors $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_{m-k}$ as approximate eigenvectors in the current approximation space. Thus, Ritz vectors are generated in the case of FOM, and harmonic Ritz vectors when GMRES is applied. This method is useful for a problem with small eigenvalues. Also, if GMRES is used for a problem with multiple right-hand sides, then the harmonic Ritz vectors can be computed once and used for all of the right-hand sides.

Note that if the Ritz vectors or the harmonic Ritz vectors are used to approximate the eigenvectors of the coefficient matrix $\mathbf{A}$, then the subspace spanned by span $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{m-k}\}$ is still a Krylov subspace generated by the matrix $\mathbf{A}$, but a different starting vector is used. This can be illustrated with a simple example. Let θ_1 and θ_2 be two harmonic Ritz values, with associated harmonic Ritz vectors $\mathbf{w}_1$ and $\mathbf{w}_2$. Therefore, $\mathbf{A}\,\mathbf{w}_j - \theta_j\,\mathbf{w}_j = \gamma_j\,\mathbf{r}_m$ for $j = 1, 2$ for a certain γ_j. Here $\mathbf{r}_m$ is the GMRES residual of the current cycle. By setting $\mathbf{s} = \gamma_2\,\mathbf{w}_1 - \gamma_1\,\mathbf{w}_2$, and $\mathbf{v}_1 = \dfrac{\mathbf{r}_m}{||\mathbf{r}_m||}$ is the initial vector for the new GMRES cycle, the following relation is found:

$$\text{span}\left\{\mathbf{s}, \mathbf{A}\,\mathbf{s}, ..., \mathbf{A}^{m-1}\,\mathbf{s}\right\} = \text{span}\left\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{v}_1, \mathbf{A}\,\mathbf{v}_1, ..., \mathbf{A}^{m-3}\,\mathbf{v}_1\right\}. \tag{3.21}$$

Morgan (2000, 2002) has introduced the concept of implicitly including the eigenvectors in restarted GMRES. The enrichment vectors are included within the Krylov subspace during each cycle. The success of an augmented approach depends on the matrix being not too far from normal.

Besides the augmented approach which includes enrichment vectors into the approximation space, another approach would be to explicitly deflate the eigenvector components and solve the deflated problem. The problem to be solved for a non-symmetric matrix $\mathbf{A}$ is

$$(\mathbf{I} - \mathbf{U}\,\mathbf{S}\,\mathbf{W}^\star)\,\mathbf{A}\,\mathbf{x} = (\mathbf{I} - \mathbf{U}\,\mathbf{S}\,\mathbf{W}^\star)\,\mathbf{b}, \tag{3.22}$$

where $\mathbf{S}$ is determined with the relation $\mathbf{S} = (\mathbf{W}^\star\,\mathbf{A}\,\mathbf{U})^{-1}$, and the columns of the matrices $\mathbf{U}$ and $\mathbf{W}$ span the approximate right and left invariant subspaces associated with a group of 'undesired' eigenvalues. However, if an exact invariant subspace of $\mathbf{A}$ is available, the residual norm obtained with this approach is not smaller than that obtained by using the corresponding augmented technique.

Another adaptive approach is based on the idea of translating a group of small eigenvalues by means of a series of low-rank projections of the coefficient matrix of the form

$$\tilde{\mathbf{A}} = \mathbf{A}\,(\mathbf{I} + \mathbf{u}_1\,\mathbf{w}_1^\star) \cdots (\mathbf{I} + \mathbf{u}_k\,\mathbf{w}_k^\star)\,, \tag{3.23}$$

where $\mathbf{u}_j$, $\mathbf{w}_j$ are the right and left eigenvectors associated with the eigenvalues which are trans-

lated. Hence, the Krylov subspace method is applied on $\tilde{\mathbf{A}}$. For a restarted method, $\tilde{\mathbf{A}}$ is updated at each restart. This can be seen as a preconditioning strategy.

Thus augmented and deflated methods show potential for acceleration of the convergence of Krylov subspace methods when restarts are necessary. The GMRES-E algorithm also shows promising results, and has been implemented and tested for the different test cases.

3.5 Generalised Krylov subspace method

As mentioned in the previous section, augmented and deflated methods show a good potential for acceleration of the convergence of Krylov subspace methods in case restarts are necessary. The GMRES-E algorithm designed by Carpenter et al. (2010) uses augmentation of the Krylov subspace, and reuses information from previous GMRES(m) cycles. This approach is the subject of this section, and has been implemented and tested for the different test cases discussed in Chapter 2. The theory behind the algorithm is discussed in depth, and the results of the simulations are discussed.

3.5.1 GMRES with enrichment

The GMRES-E algorithm is designed to solve large, slowly varying linear systems:

$$\mathbf{A}^i \mathbf{x} = \mathbf{b}^i. \tag{3.24}$$

The method consists of the conventional restarted GMRES algorithm where k enrichment vectors are prepended to the Krylov subspace. The algorithm consists of four main elements:

1. Preprocessing of previous solutions: an optimal starting solution $\overline{\mathbf{x}}_0$ is determined based on previous solutions under the condition that the system matrix $\mathbf{A}$ is constant, i.e. $\mathbf{A}^i = \mathbf{A}^{i-1}$.
2. Selection of enrichment vectors: this element is performed for each consecutive GMRES cycle, or when the current GMRES cycle is restarted. The eigenvalues which are inadequately clustered by the preconditioner are selected and prepended to the Krylov subspace. These eigenvalues contribute to poor convergence of the GMRES solver.
3. Data rotation and compression: the Arnoldi relationship $\mathbf{A}^i \mathbf{S}_k = \mathbf{V}_k \overline{\mathbf{H}}_k$ is constructed for the selected enrichment vectors. In case the system matrix $\mathbf{A}$ is unchanged, an existing Arnoldi relationship of the previous GMRES cycle $\mathbf{A}^i \mathbf{S}_m = \mathbf{V}_{m+1} \overline{\mathbf{H}}_m$ is rotated and compressed.
4. Initialisation with Galerkin projection: the new starting residual ${\mathbf{r}_0}^i$ is preprocessed in order to ensure consistency with the current Arnoldi relation.

Note that the convention is followed that matrices written with an over bar are non squares matrices, and matrices without an over bar are square matrices.

3.5.2 Preprocessing of previous solutions

The first enhancement to the GMRES algorithm consists of the reuse of previous solutions for linear systems where the system matrix $\mathbf{A}$ is equal to the previous systems. However, the right hand side vector $\mathbf{b}$ is allowed to change for a small amount. By reusing the solutions from previous linear systems, presumably an optimal starting solution and starting residual is chosen, thus hereby improving the convergence of the GMRES algorithm.

Hence, it is assumed that $i-1$ linear problems have been solved, where the system matrix $\mathbf{A}$ remained unchanged. The current right hand side vector $\mathbf{b}^i$ is projected onto the data space $span\left\{\mathbf{b}^1, ..., \mathbf{b}^{i-1}\right\}$ in order to determine ${\mathbf{b}_\perp}^i$. Thus the vector $\mathbf{b}^i$ is decomposed into $\mathbf{b}^i = {\mathbf{b}_\perp}^i + \left(\mathbf{b}^i - {\mathbf{b}_\perp}^i\right)$.

Practical implementations solve the system $\mathbf{A}^i\,\mathbf{x}^i = \mathbf{b}^i - \mathbf{r}^i$, instead of $\mathbf{A}^i\,\mathbf{x}^i = \mathbf{b}^i$. Therefore, the vector $\mathbf{b}^i$ is projected onto the data space $span\left\{\mathbf{b}^1 - \mathbf{r}^1, ..., \mathbf{b}^{i-1} - \mathbf{r}^{i-1}\right\}$. Summarising, two vector spaces are defined:

$$\mathbf{\Gamma}^{i-1} = \left[\mathbf{x}^1, ..., \mathbf{x}^{i-1}\right] \tag{3.25}$$

and

$$\mathbf{\Omega}^{i-1} = \left[\mathbf{b}^1 - \mathbf{r}^1, ..., \mathbf{b}^{i-1} - \mathbf{r}^{i-1}\right]. \tag{3.26}$$

Ω^{i-1} is orthogonalised such that

$$\mathbf{\Omega}^{i-1} = \mathbf{\Theta}^{i-1}\,\mathbf{H}_{br} \tag{3.27}$$

and

$$\left[\mathbf{\Theta}^{i-1}\right]^T\,\mathbf{\Theta}^{i-1} = \mathcal{I}. \tag{3.28}$$

With these definitions for $\mathbf{\Omega}^{i-1}$ and $\mathbf{\Theta}^{i-1}$, the data from the previous problems can be described as

$$\mathbf{A}\,\mathbf{\Gamma}^{i-1} = \mathbf{\Theta}^{i-1}\,\mathbf{H}_{br}. \tag{3.29}$$

Hence the L_2 optimal starting guess ${\mathbf{x}_0}^i$ can be computed with

$${\mathbf{x}_0}^i = {\overline{\mathbf{x}}_0}^i + \mathbf{\Gamma}^{i-1}\,{\mathbf{H}_{br}}^{-1}\left[\mathbf{\Theta}^{i-1}\right]^T\,\overline{\mathbf{r}}_0, \tag{3.30}$$

given the initial guess ${\overline{\mathbf{x}}_0}^i$, and initial residual $\overline{\mathbf{r}}_0$. The starting residual $\mathbf{r}_0$ can be determined with

$$\mathbf{r}_0 = \left(\mathcal{I} - \mathbf{\Theta}^{i-1}\,\left[\mathbf{\Theta}^{i-1}\right]^T\right)\overline{\mathbf{r}}_0. \tag{3.31}$$

The proof for these equations can be found in Carpenter et al. (2010).

The advantage of this preprocessing step is that the solution to the problem $\mathbf{A}\,\mathbf{x}^i = \left(\mathbf{b}^i - \mathbf{b}_\perp{}^i\right)$ is already available from the solution space $span\left\{\mathbf{x}^1, ..., \mathbf{x}^{i-1}\right\}$. The remaining problem which need to be solved is $\mathbf{A}\,\mathbf{x}^i = \mathbf{b}_\perp{}^i$. Thus information from previous GMRES cycles is efficiently reused to start the next GMRES solve. However, as mentioned before, this step can only be used in case the system matrix $\mathbf{A}$ is constant. Therefore, Rosenbrock time integration schemes can use this step to speed up the convergence of the GMRES algorithm, since the system matrix is constant for the consecutive stages of the Rosenbrock scheme. Contrary to Rosenbrock schemes, ESDIRK schemes combined with a nonlinear Newton solver cannot take advantage of this step, since the Newton iterations change the system matrix.

3.5.3 Selection of enrichment vectors

Various different approaches can be used for the selection of the enrichment vectors. One approach would be to use physical arguments to construct approximate eigenvectors. Nicolaides (1987) derives algebraic deflation vectors which are used to deflate coarse-grid information. Another approach would be to reuse the vectors which contributed significantly to the convergence of previous iterations (De Sturler, 1999). The approach followed by Carpenter et al. (2010) and which is also used in this thesis, is to identify the eigenvalues which are inadequately clustered by the preconditioner. These eigenvalues contribute to poor convergence of the GMRES algorithm, and are therefore used as enrichment vectors.

Determination of eigenvectors

Ritz-values and Ritz-vectors are commonly used to approximate the eigenvalues and eigenvectors of the system matrix $\mathbf{A}$ (Morgan and Zeng, 1998). When the iterations of the GMRES algorithm progresses, the accuracy of the approximation for the eigenvalues and eigenvectors of the system matrix increases. The derivation for these Ritz-values and vectors starts with the non-symmetric eigenvalue problem

$$\left(\mathbf{A} - \theta\,\mathcal{I}\right)\zeta = 0, \tag{3.32}$$

where $\mathbf{A}$ is the system matrix, θ is an eigenvalue and ζ is the corresponding eigenvector. With a Petrov-Galerkin projection technique, an approximate solution to the eigenvalue problem given by Equation (3.32) can be found. Thus, Equation (3.32) is approximated with

$$\left[\mathbf{A} - \breve{\theta}\,\mathcal{I}\right]\mathbf{S}_m\,\breve{\xi}_m = 0, \tag{3.33}$$

where $\mathbf{S}_m$ is the search space for an approximate eigenvector, and $\breve{\xi}_m$ is an arbitrary vector which combines the basis vectors in the search space $\mathbf{S}_m$. Following the Petrov-Galerkin projection technique, Equation (3.33) is constrained such that an approximation error that exists is orthogonal

to the subspace $\mathbf{A}\,\mathbf{S}_m$. With the eigenpair $\left(\tilde{\theta}, \tilde{\xi}\right)$, the resulting equation is

$$\left(\mathbf{A}\,\mathbf{S}_m\right)^T \left[\mathbf{A} - \tilde{\theta}\,\mathcal{I}\right] \mathbf{S}_m\,\tilde{\xi}_m = 0. \tag{3.34}$$

And with the definition $\mathbf{\Gamma} = \left(\mathbf{V}_{m+1}\right)^T \mathbf{S}_m$, the $m \times m$ generalised eigenvalue problem can be described as

$$\left\{\mathbf{A}_1 - \tilde{\theta}\,\mathbf{A}_2\right\} \tilde{\xi}_m = 0, \tag{3.35}$$

where $\mathbf{A}_1$ is given by $\mathbf{A}_1 = \overline{\mathbf{H}}_m{}^T\,\overline{\mathbf{H}}_m$ and $\mathbf{A}_2$ is determined with $\mathbf{A}_2 = \overline{\mathbf{H}}_m{}^T\,\mathbf{\Gamma}$. This generalised eigenvalue problem be easily solved with the routine GGEVX of the software package LAPACK (Netlib, 2012).

An equivalent harmonic Ritz eigenvalue problem can be formulated which has a decreased condition number. By introducing $\mathbf{h}_e = \mathbf{H}_m{}^{-T}\,\mathbf{e}_m$, the following formulation is found:

$$\left\{\overline{\mathbf{A}}_1 - \tilde{\theta}\,\overline{\mathbf{A}}_2\right\} \tilde{\xi}_m = 0, \tag{3.36}$$

where $\overline{\mathbf{A}}_1$ is given by $\overline{\mathbf{A}}_1 = \mathbf{H}_m + \left(h_{m+1,m}\right)^2 \mathbf{h}_e\,\mathbf{e}_m{}^T$ and $\overline{\mathbf{A}}_2$ is determined by evaluating $\overline{\mathbf{A}}_2 = \left(\mathbf{V}_m\right)^T \mathbf{S}_m + \left(h_{m+1,m}\right) \mathbf{h}_e\,\mathbf{e}_{m+1}{}^T\,\mathbf{\Gamma}$.

Selection of eigenvectors

With the harmonic Ritz-values and Ritz-vectors known, the optimal choice would be to select the k smallest harmonic Ritz values in magnitude. The harmonic Ritz procedure is known to accurately predict the small eigenvalues of the matrix $\mathbf{A}$ (Carpenter et al., 2010). For most cases, this is the best choice, but exceptions do exist. When a preconditioner is used to speed up the convergence of the iterative solver, the eigenvalues of the resulting system are clustered within the unit circle with the centre at $(1.0, 0.0)$ in the complex plane. In case eigenvalues are close to the origin, the selection procedure regards them as problematic. Therefore, another selection criterion is needed in case the eigenvalues are far removed from the origin.

Carpenter et al. (2010) propose four different selection criteria, by assigning a merit to each eigenvector and eigenvalue pair. The values with the smallest merit are used for the Krylov subspace enrichment. The merit functions are:

1. $|\zeta_j| = \sqrt{\left(\theta_r\right)_j^2 + \left(\theta_i\right)_j^2}$
2. $|\zeta_j| = \dfrac{1}{\sqrt{\left(1 - \theta_r\right)_j^2 + \left(\theta_i\right)_j^2}}$
3. $|\zeta_j| = \dfrac{-\left(\theta_r\right)_j}{\sqrt{\left(1 - \theta_r\right)_j^2 + \left(\theta_i\right)_j^2}}$

4. $|\zeta_j| = \dfrac{\sqrt{(-0.25-\theta_r)_j^2 + (\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}$

θ_r and θ_i represent the real and imaginary part of the eigenvalue θ.

Unfortunately, the choice of the merit function has a substantial influence on the performance of the GMRES-E algorithm (Carpenter et al., 2010). The first function selects the Ritz values based on the distance from the origin. The inverse distance from the point $(1.0, 0.0)$ is used by the second function. The third function selects the Ritz values based on their inverse distance from the point $(1.0, 0.0)$, and assigns a higher merit to the eigenvalues located in the left half-plane of the complex plane. The last function selects Ritz values located outside of the unit circle, and gives a higher merit to values located close to the point $(-0.25, 0.0)$.

Note that the selection algorithm should be modified when a complex conjugate pair is selected. In case the conjugate pair is assigned the ranks k and $k+1$, then $k-1$ vectors should be used for the enrichment instead of k vectors.

The selected vectors are assembled in the matrix $\tilde{\mathbf{P}}_k$. For the complex conjugate pairs, Schur vectors are used instead of the eigenvectors. When the Schur vectors are used, the column span of $\tilde{\mathbf{P}}_k$ is retained, and the matrix $\tilde{\mathbf{P}}_k$ consists of real entries. The Schur vectors are determined from the conjugate eigenvector pairs $\tilde{\xi}_{n\,j}$, $\tilde{\xi}_{n\,j+1}$ via the rotation

$$\begin{bmatrix} \xi'_{n\,j} \\ \xi'_{n\,j+1} \end{bmatrix} = \tfrac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{n\,j} \\ \tilde{\xi}_{n\,j+1} \end{bmatrix}. \tag{3.37}$$

As a last final step, the modified Gram-Schmidt algorithm is used on $\tilde{\mathbf{P}}_k$ to produce the orthogonal matrix $\overline{\mathbf{P}}_k$. The columns of $\overline{\mathbf{P}}_k$ are no longer the eigenvectors of the harmonic Ritz problem, but the column span of $\tilde{\mathbf{P}}_k$ has not been changed.

Summarising, the selection of the enrichment vectors consists of a procedure to approximate the eigenvalues and eigenvectors of the system matrix $\mathbf{A}$ with harmonic Ritz values and Ritz vectors. The standard approach would be to select the harmonic Ritz values smallest in magnitude. Other approaches are to select Ritz values based on their inverse distance to the point $(1.0, 0.0)$ in the complex plane. Finally, the orthogonal matrix $\overline{\mathbf{P}}_k$ is calculated with the modified Gram-Schmidt procedure applied on the selected harmonic Ritz vectors.

3.5.4 Data rotation and compression

The existing Arnoldi relation is rotated or compressed in preparation for the next GMRES cycle. This step is performed in order to construct the Arnoldi relationship $\mathbf{A}^i\,\mathbf{S}_k = \mathbf{V}_k\,\overline{\mathbf{H}}_k$ for the selected enrichment vectors. The matrix $\overline{\mathbf{P}}_k$ contains the Ritz vectors which follow from the

selection of the enrichment vectors. This rotation and compression step can only be used in case the system matrix $\mathbf{A}$ is unchanged. When the system matrix $\mathbf{A}$ has been changed, for example when Newton iterations are carried out to solve the root problem, then the selected enrichment vectors need to be multiplied with the system matrix $\mathbf{A}$ via the Arnoldi procedure.

The previous Arnoldi $\mathbf{A}^{i-1}\,\mathbf{S}_n = \mathbf{V}_{n+1}\,\overline{\mathbf{H}}_n$ is right multiplied by the matrix $\overline{\mathbf{P}}_k$ forming the relation

$$\mathbf{A}\,\mathbf{S}_n\,\overline{\mathbf{P}}_k = \mathbf{V}_{n+1}\,\overline{\mathbf{H}}_n\,\overline{\mathbf{P}}_k. \tag{3.38}$$

Thereafter, QR factorisations are used to decompose the matrix product $\overline{\mathbf{H}}_n\,\overline{\mathbf{P}}_k$ into

$$\overline{\mathbf{H}}_n\,\overline{\mathbf{P}}_k = \overline{\mathbf{Q}}_n\,\mathbf{R}_k. \tag{3.39}$$

Now, the matrices $\mathbf{S}_k$ and $\mathbf{V}_k$ are defined as

$$\mathbf{S}_k = \mathbf{S}_n\,\overline{\mathbf{P}}_k \tag{3.40}$$

and

$$\mathbf{V}_k = \mathbf{V}_{n+1}\,\overline{\mathbf{Q}}_n. \tag{3.41}$$

Hence, the compressed system is described with the relation

$$\mathbf{A}\,\mathbf{S}_k = \mathbf{V}_k\,\mathbf{R}_k, \tag{3.42}$$

where the matrix $\mathbf{V}_k$ is orthogonal (Carpenter et al., 2010).

As mentioned, when the system matrix $\mathbf{A}$ has been changed, then the selected enrichment vectors need to be left-multiplied with $\mathbf{A}$ to form the Arnoldi relationship $\mathbf{A}^i\,\mathbf{S}_k = \mathbf{V}_k\,\mathbf{H}_k$. The standard Arnoldi procedure is used to construct the Arnoldi relationship $\mathbf{A}^i\,\mathbf{S}_k = \mathbf{V}_k\,\mathbf{H}_k$, as shown in Algorithm 3.4.

3.5.5 Initialisation with Galerkin projection

The current residual needs to be included in the Krylov subspace, and the next iteration needs to be started. The Galerkin projection technique is used to ensure consistency with the current Arnoldi iteration.

Carpenter et al. (2010) show that the L_2 optimal starting solution $\mathbf{x}_0$ can be constructed from the vectors $\mathbf{S}_k$ and a starting guess $\overline{\mathbf{x}}_0$ with

$$\mathbf{x}_0 = \overline{\mathbf{x}}_0 + {\mathbf{M}_2}^{-1}\,\mathbf{S}_k\,{\mathbf{R}_k}^{-1}\,{\mathbf{V}_k}^T\,\overline{\mathbf{r}}_0. \tag{3.43}$$

Algorithm 3.4 Left multiplication of selected enrichment vectors with the system matrix $\mathbf{A}$ to form the Arnoldi relationship $\mathbf{A}^i\,\mathbf{S}_k = \mathbf{V}_k\,\mathbf{H}_k$, with left and right preconditioning.

for $j = 1 \rightarrow k$ **do**
  Compute $\mathbf{w}_j = {\mathbf{M}_1}^{-1}\,\mathbf{A}\,{\mathbf{M}_2}^{-1}\,(\mathbf{S}_k)_j$
  for $i = 1 \rightarrow j$ **do**
    $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$
    $\mathbf{w}_j = \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$
  end for
  $h_{j+1,j} = ||\mathbf{w}_j||_2$
  if $h_{j+1,j} = 0$ **then**
    Stop
  end if
  $\mathbf{v}_{j+1} = \dfrac{\mathbf{w}_j}{h_{j+1,j}}$
end for
return $\mathbf{V}_k$, $\mathbf{H}_k$

The starting residual $\mathbf{r}_0$ can be determined with

$$\mathbf{r}_0 = \left(\mathbf{I} - \mathbf{V}_k\,{\mathbf{V}_k}^T\right)\bar{\mathbf{r}}_0. \tag{3.44}$$

The new system $\mathbf{A}\,\mathbf{S}_k = \mathbf{V}_{k+1}\,\overline{\mathbf{R}}_k$ is now ready to start the conventional GMRES algorithm to generate a $m-k$ dimensional Krylov subspace. Algorithm 3.5 shows the pseudo code for the complete GMRES-E algorithm.

3.5.6 Implementation details of GMRES-E for ESDIRK and Rosenbrock time integration schemes

The application of the GMRES-E algorithm differs per used time integration scheme. The most important difference between the Rosenbrock and ESDIRK time integration schemes, is that the system matrix for the Rosenbrock time integration scheme is constant for all stages, whereas the system matrix for ESDIRK combined with a Newton-Krylov solver is not constant.

For the Rosenbrock time integration schemes, the four main elements of the GMRES-E method can be used to speed up the convergence of the GMRES algorithm for the consecutive stages of the Rosenbrock method. The system matrix of the Rosenbrock scheme is constant for all stages. As a result, the solutions of the previous stages can be reused to speed up the convergence of a new GMRES cycle. Also, data rotation and compression can be performed efficiently, since the Arnoldi relationship of the previous GMRES-E cycle is still available from memory and can be reused.

Algorithm 3.5 GMRES(m) with q enrichment vectors to solve $\mathbf{A}^\kappa \mathbf{x} = \mathbf{b}^\kappa$ with left and right preconditioning applied (Carpenter et al., 2010)

while Nonlinear convergence criterion **do**
  $\kappa = \kappa + 1$
  Construct $\mathbf{A}^\kappa$ and $\mathbf{b}^\kappa$, choose $\overline{\mathbf{x}}_0$ and compute $\overline{\mathbf{r}}_0 = {\mathbf{M}_1}^{-1} (\mathbf{b} - \mathbf{A}^\kappa \overline{\mathbf{x}}_0)$
  if $\mathbf{A}^\tau = ... = \mathbf{A}^{\kappa-1} = \mathbf{A}^\kappa$ **then**
    Project $\overline{\mathbf{x}}$ and $\overline{\mathbf{r}}_0$ onto $\mathbf{A}\,\mathbf{x}^{[\tau,\dots,\kappa-1]} = \mathbf{b}^{[\tau,\dots,\kappa-1]}$ for optimal initial $\mathbf{x}_0$ and $\mathbf{r}_0$
  end if
  while Linear convergence criterion **do**
    Initialise $\mathbf{S}_q^\kappa$ with q enrichment vectors
    if Restart **then**
      Use problematic eigenvectors from $\mathbf{A}^\kappa$
    end if
    if Same $\mathbf{A}$, new $\mathbf{b}^\kappa$, and available memory **then**
      $\mathbf{S}_q^\kappa = \mathbf{S}_m^{\kappa-1}$
    end if
    Construct the square Arnoldi relation $\mathbf{A}^\kappa \mathbf{S}_q = \mathbf{V}_q \mathbf{H}_q$
    Orthogonalise $\mathbf{r}_0$ against $\mathbf{V}_q$ such that $\tilde{\mathbf{r}}_0 \perp \mathbf{V}_q$
    Compute $\beta = ||\tilde{\mathbf{r}}_0||_2$; $\mathbf{v}_{q+1} = \frac{\tilde{\mathbf{r}}_0}{\beta}$; $\overline{H}_{q+1,j} = \mathbf{0}^T$, $j = 1 \rightarrow q$
    Set $j = q + 1$
    while Convergence criterion and $j < m$ **do**
      $j = j + 1$
      $\mathbf{v}_{j+1} = {\mathbf{M}_1}^{-1} \mathbf{A}^\kappa {\mathbf{M}_2}^{-1} \mathbf{v}_j$
      for $i = 0 \rightarrow j$ **do**
        $\overline{H}_{i,j} = (\mathbf{v}_i,\, \mathbf{v}_{j+1})$
        $\mathbf{v}_{j+1} = \mathbf{v}_{j+1} - \overline{H}_{i,j}\, \mathbf{v}_i$
      end for
      $\overline{H}_{j+1,j} = ||\mathbf{v}_{j+1}||_2$; $\mathbf{v}_{j+1} = \frac{\mathbf{v}_{j+1}}{\overline{H}_{j+1,j}}$
    end while
    Define $\mathbf{S} = [\mathbf{S}_q,\, \mathbf{V}_{q+1\rightarrow j}]$
    Compute $\mathbf{y} = \min_{\mathbf{y} \in \mathcal{R}^q} \left|\left| \beta\, \mathbf{e}_q - \overline{\mathbf{H}}_j\, \mathbf{y} \right|\right|$; $\mathbf{r}_j = \mathbf{V}_{j+1} \left[\beta\, \mathbf{e} - \overline{\mathbf{H}}_j\, \mathbf{y} \right]$
    $\mathbf{x}_j = \mathbf{x}_0 + {\mathbf{M}_2}^{-1} \mathbf{S}\, \mathbf{y}$
    Estimate problematic eigenvectors from $\mathbf{A}^\kappa \mathbf{S}_j = \mathbf{V}_{j+1} \overline{\mathbf{H}}_j$
    if Linear convergence **then**
      Save $\mathbf{x}^\kappa$ and $\mathbf{b}^\kappa$ as $\mathbf{A}\,\mathbf{x}^{[\tau,\dots,\kappa]} = \mathbf{b}^{[\tau,\dots,\kappa]}$
      Exit to nonlinear loop
    else
      Restart: $\mathbf{x}_0 = \mathbf{x}_j$ and $\mathbf{r}_0 = \mathbf{r}_j$
    end if
  end while
end while

In contrast to the Rosenbrock scheme, the ESDIRK time integration method uses an inexact Newton-Krylov solver to solve the implicit stages. Since the system matrix A is changed at every stage and at every Newton iteration, it is not possible to reuse the solutions of the previous stages. Data rotation and compression can also not be performed efficiently if a new Newton iteration is started, since the selected enrichment vectors need to be multiplied with the system matrix A via the Arnoldi algorithm. As a consequence, the computational costs of including one enrichment vector of a previous GMRES-E cycle into the Krylov subspace is equal to the computational cost of one GMRES iteration, under the assumption that the computational costs of the remaining steps of the GMRES-E method are negligible. However, in case the GMRES algorithm is restarted during one Newton iteration, the selected enrichment vectors can efficiently be included into the new Krylov subspace.

3.6 Nonlinear convection-diffusion equation

The same test cases which were used to judge the performance of the ESDIRK and Rosenbrock time integration schemes are also used to study the effect of the enrichment of the Krylov subspace on computational efficiency. In this section the results for the nonlinear convection-diffusion test case are shown and discussed. Thereafter, Section 3.7 discusses the results for the uniform flow past a circular cylinder case. The governing equation and further details on the nonlinear convection-diffusion test case can be found in Section 2.6.

3.6.1 Influence of the number of Ritz vectors used for the Krylov subspace enrichment

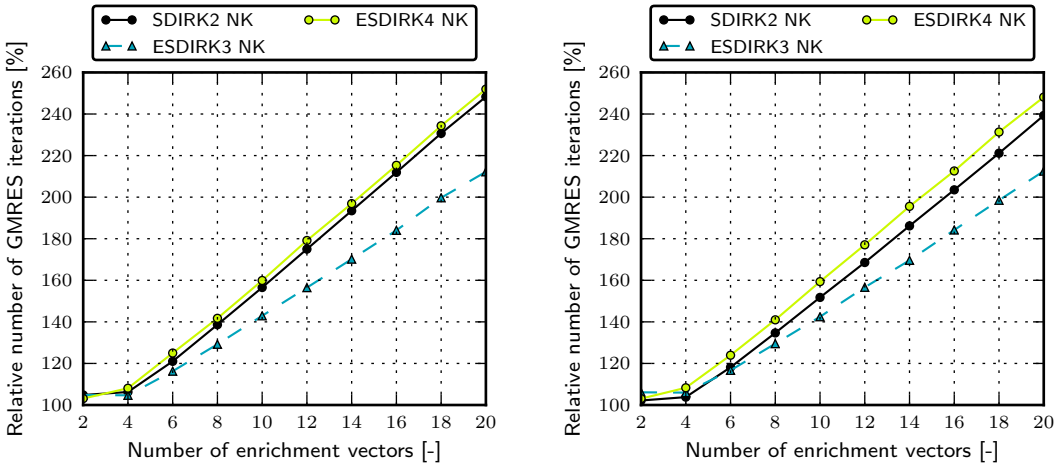
The influence of the number of enrichment vectors on the computational efficiency of the solver is studied for several nonlinear convection-diffusion cases. The time integration is performed with the fixed time step $\Delta t = 1.25 \cdot 10^{-4}$ and tolerance $TOL_{NK} = TOL_{GMRES} = 1.0 \cdot 10^{-6}$ for the Newton iterations and the GMRES algorithm. The results of the simulations are shown in Figures 3.1 and 3.2 for the ESDIRK schemes and Rosenbrock schemes, respectively. The total number of GMRES iterations is scaled with the number of GMRES iterations with the standard GMRES algorithm applied. The first selection criterion or classical selection criterion is used to select the enrichment vectors for the simulations shown in this section, as shown in Section 3.5.3.

Judging from Figure 3.1, the use of the GMRES-E algorithm does not accelerate the convergence of the GMRES algorithm in case the ESDIRK time integration scheme is used in combination with an incomplete Newton-Krylov solver. For two and four enrichment vectors, the number of GMRES iterations is approximately equal to the reference computations. For more than four enrichment vectors, the number of GMRES iterations increases linearly indicating that the reuse of Krylov subspace vectors does not accelerate the convergence of the GMRES algorithm, but rather

has a negative effect on the performance of GMRES.

In contrast to ESDIRK, the number of GMRES iterations is decreased by at least $35\,\%$ for the Rosenbrock time integration schemes, as shown in Figure 3.2. This indicates that the reuse of Krylov subspace vectors does lead to an acceleration of the GMRES algorithm. The efficiency of the algorithm increases for an increasing number of enrichment vectors. The efficiency of the GMRES-E algorithm is reduced when the amount of non linearity is increased for this problem. Also, the effect of the GMRES-E method depends on the used time integration scheme. The inclusion of the enrichment vectors is the most effective in case the ROSI2PW scheme is used.

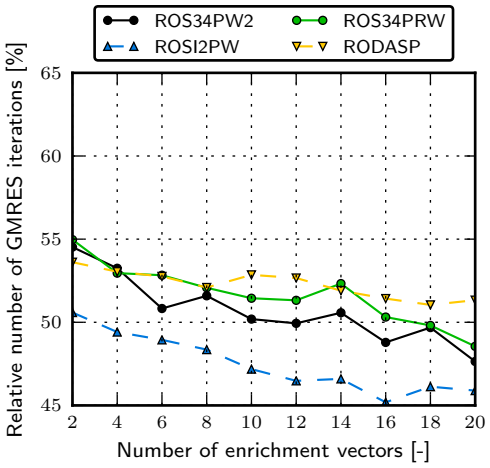
Summarising, the reuse of Krylov subspace vectors accelerates the convergence of the GMRES algorithm in case the Rosenbrock time integration scheme is used. For an incomplete Newton-Krylov solver this is not the case however. The inclusion of the enrichment vectors in the Krylov subspace cause an increase of GMRES iterations for the ESDIRK schemes. For the Rosenbrock schemes, an increasing number of enrichment vectors causes the number of GMRES iterations to decrease.



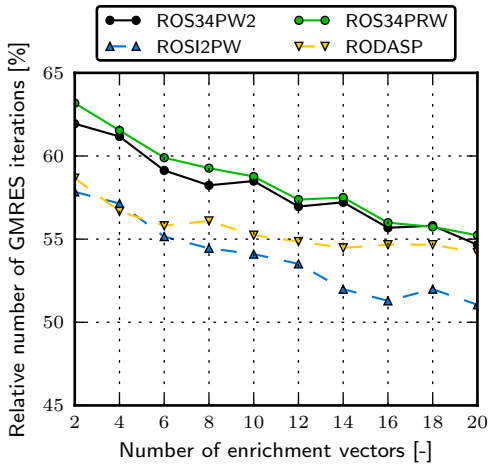
(a) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = 3,\ m = 1$)

(b) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 3$)

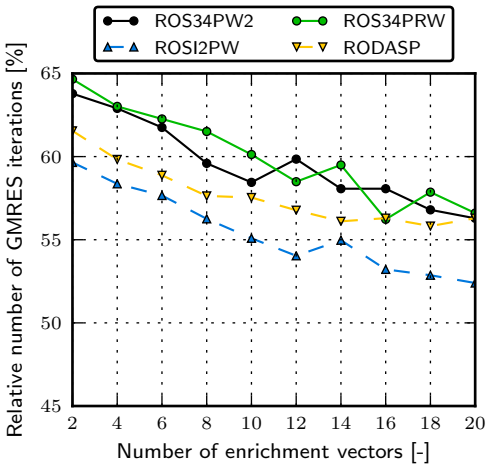
Figure 3.1: Nonlinear convection-diffusion case: influence of the number of Ritz vectors used for the Krylov subspace enrichment on the number of GMRES iterations for the ESDIRK time integration schemes. The total number of GMRES iterations is scaled with the number of GMRES iterations of the standard GMRES algorithm. The simulations are run for two non linearity settings.



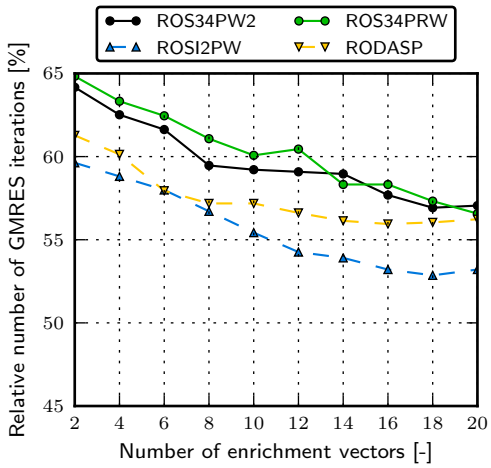
(a) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 0$)



(b) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 1$)



(c) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = 3$, $m = 1$)



(d) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 3$)

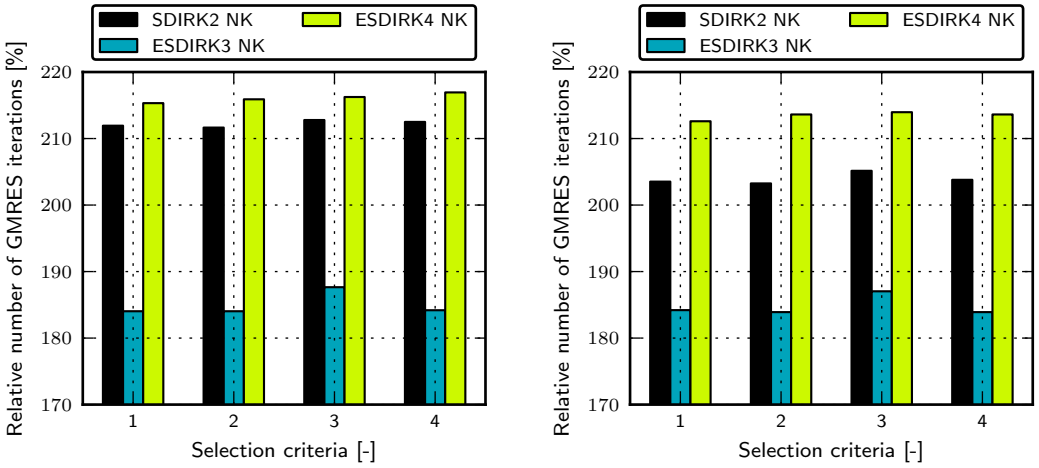
Figure 3.2: Nonlinear convection-diffusion case: influence of the number of Ritz vectors used for the Krylov subspace enrichment on the number of GMRES iterations for the Rosenbrock time integration schemes. The total number of GMRES iterations is scaled with the number of GMRES iterations of the standard GMRES algorithm. The simulations are run for various non linearity settings.

3.6.2 Influence of selection criteria of the enrichment vectors on computational efficiency

The influence of the used selection criteria is also studied for several nonlinear convection-diffusion cases. The same settings for the time integration were used as for the research on the number of enrichment vectors, namely $\Delta t = 1.25 \cdot 10^{-4}$ and $TOL_{NK} = TOL_{GMRES} = 1.0 \cdot 10^{-6}$. Sixteen enrichment vectors are included in the Krylov subspace between restarts of the GMRES-E algorithm. The results of the simulations are shown in Figure 3.3 and 3.4 for ESDIRK and Rosenbrock, respectively.

It is apparent from Figure 3.3 that the selection criteria does have an influence on the convergence behaviour of the GMRES algorithm. However, the efficiency of the GMRES algorithm is not increased when an ESDIRK time integration scheme is used for every selection criterion.

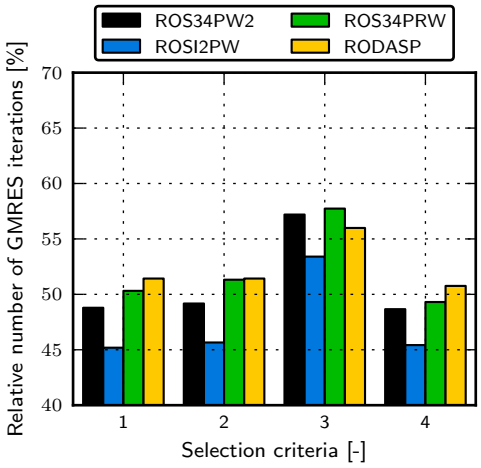
For the Rosenbrock time integration schemes, the performance of the GMRES-E algorithm with different selection criteria is similar, except for the third selection criterion, as shown in Figure 3.4. In case Ritz vectors are selected based on their inverse distance from the point $(1.0, 0.0)$ in the complex plane, the performance of GMRES is less in comparison with the other three selection criteria.



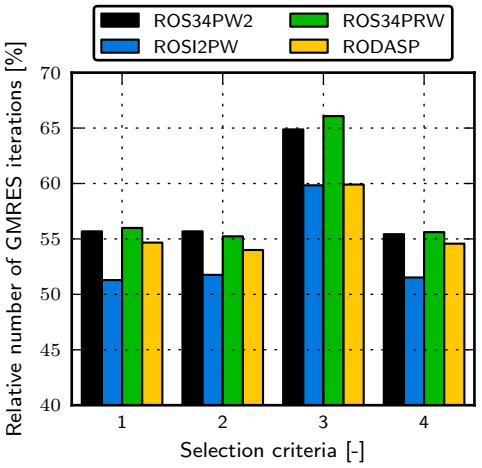
(a) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = 3$, $m = 1$)

(b) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 3$)

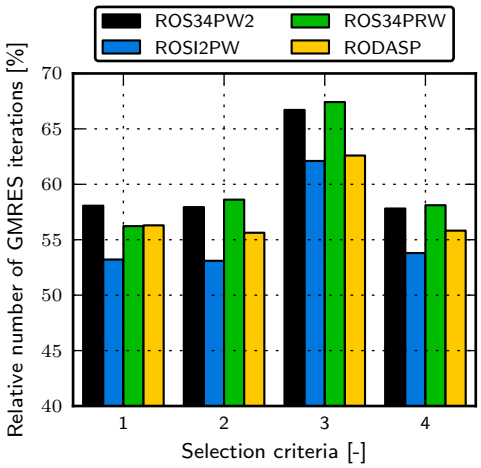
Figure 3.3: Nonlinear convection-diffusion case: influence of the selection criteria used for the Krylov subspace enrichment on the number of GMRES iterations for the ESDIRK time integration schemes. The total number of GMRES iterations is scaled with the number of GMRES iterations of the standard GMRES algorithm. The simulations are run for various non linearity settings.



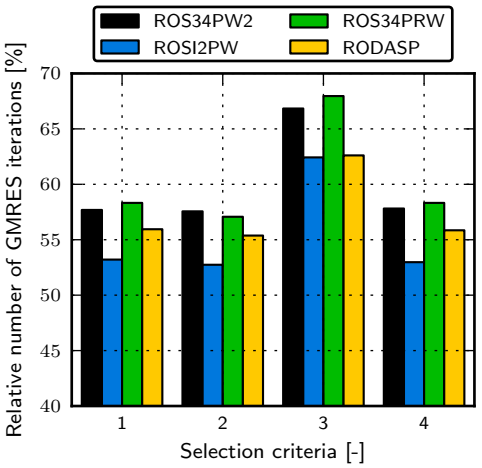
(a) Relative number of GMRES iterations for the linear convection-diffusion equation ($n = m = 0$)



(b) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 1$)



(c) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = 3$, $m = 1$)



(d) Relative number of GMRES iterations for the non-linear convection-diffusion equation ($n = m = 3$)

Figure 3.4: Nonlinear convection-diffusion case: influence of the selection criteria used for the Krylov subspace enrichment on the number of GMRES iterations for the Rosenbrock time integration schemes. The total number of GMRES iterations is scaled with the number of GMRES iterations of the standard GMRES algorithm. The simulations are run for various non linearity settings.

3.7 Uniform flow past a circular cylinder

The uniform flow around a cylinder test case has also been used to verify the performance of the GMRES-E algorithm. A complete description of the mesh and computational settings can be found in Section 2.7. The influence of the number of enrichment vectors and the used selection criteria is studied for this test case. The ESDIRK time integration scheme is not considered in this section, since preliminary computations showed that the use of the GMRES-E algorithm caused an increase of GMRES iterations, as is also the case for the convection-diffusion test case.

The time integration is performed with the fixed time step $\Delta t = 2.5 \cdot 10^{-4}$ and tolerance $TOL_{GMRES} = 1.0 \cdot 10^{-6}$ for the GMRES algorithm. The results of the computations are shown in Figure 3.5 and 3.6. Figure 3.5 shows the amount of computational time, where the computational time is scaled with the computational time for the standard GMRES algorithm. The number of enrichment vectors is varied, and the used selection criterion is also varied as shown in the subplots of Figure 3.5. The total number of GMRES iterations are shown in Figure 3.6.

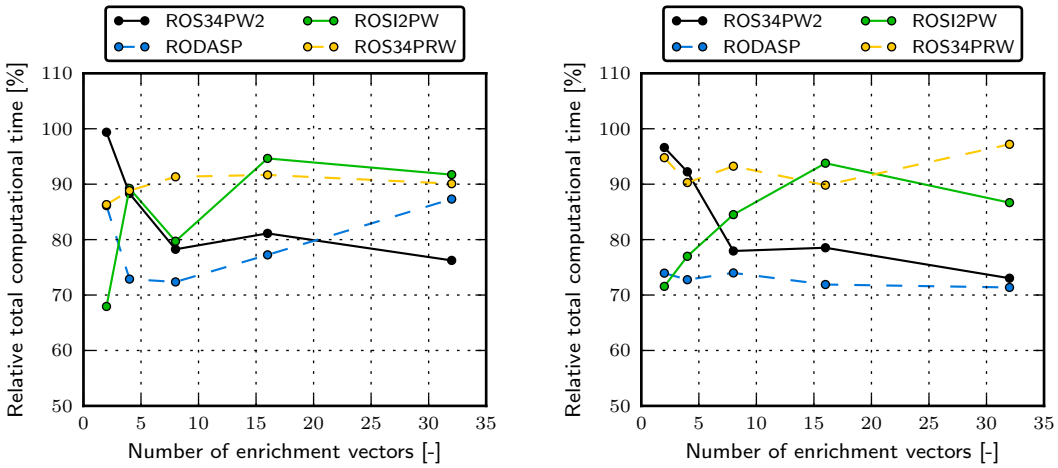
As shown in Figure 3.5, the use of the GMRES-E algorithm leads to a decrease in computational time for the Rosenbrock time integration schemes. The preprocessing step has the biggest impact on the total computational time. In case the fourth selection criterion is used, increasing the number of enrichment vectors leads to a decrease in computational time except for ROSI2PW. However, judging from the results for the other selection criteria, increasing the number of enrichment vectors may also lead to an increase in computational time. For a large number of enrichment vectors, the gain in computational efficiency seems to flatten out. This is caused by the fact that the GMRES algorithm is already converged in less iterations than the number of used enrichment vectors. Thus all the Krylov subspace vectors are reused by the algorithm.

When comparing the computational time with the number of GMRES iterations, it is unclear which number of enrichment vectors and which selection criterion provides the best performance improvement. Increasing the number of enrichment vectors may lead to a large increase in GMRES iterations. The fourth selection criterion seems to produce the best results. However, it is unclear whether this criterion should be used for other test cases.

3.8 Summary

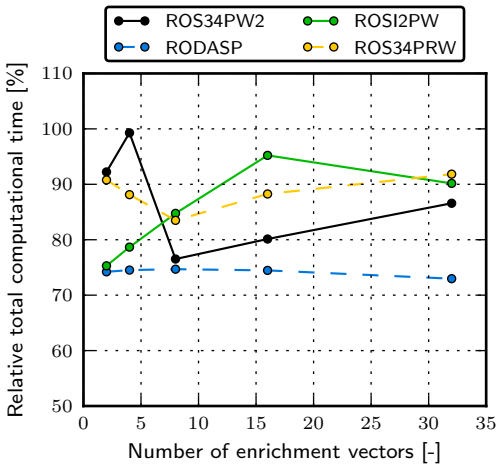
The GMRES algorithm has been discussed to solve the linear system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ for application in a flow solver. Besides the conventional GMRES algorithm, several approaches are discussed which supposedly reduce the computational costs of the method, and reduce the number of iterations needed to satisfy the convergence criteria.

GMRES is one of the most popular methods for solving non-symmetric systems. The method

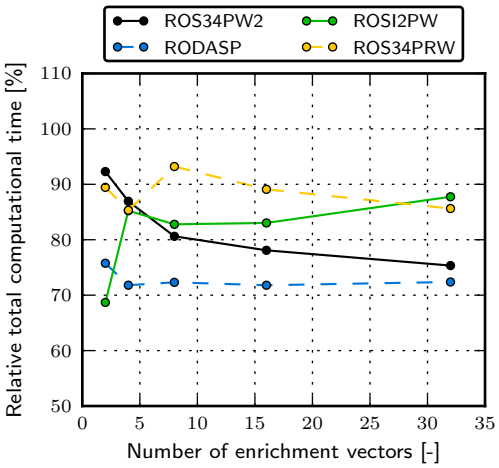


(a) Selection criteria 1

(b) Selection criteria 2

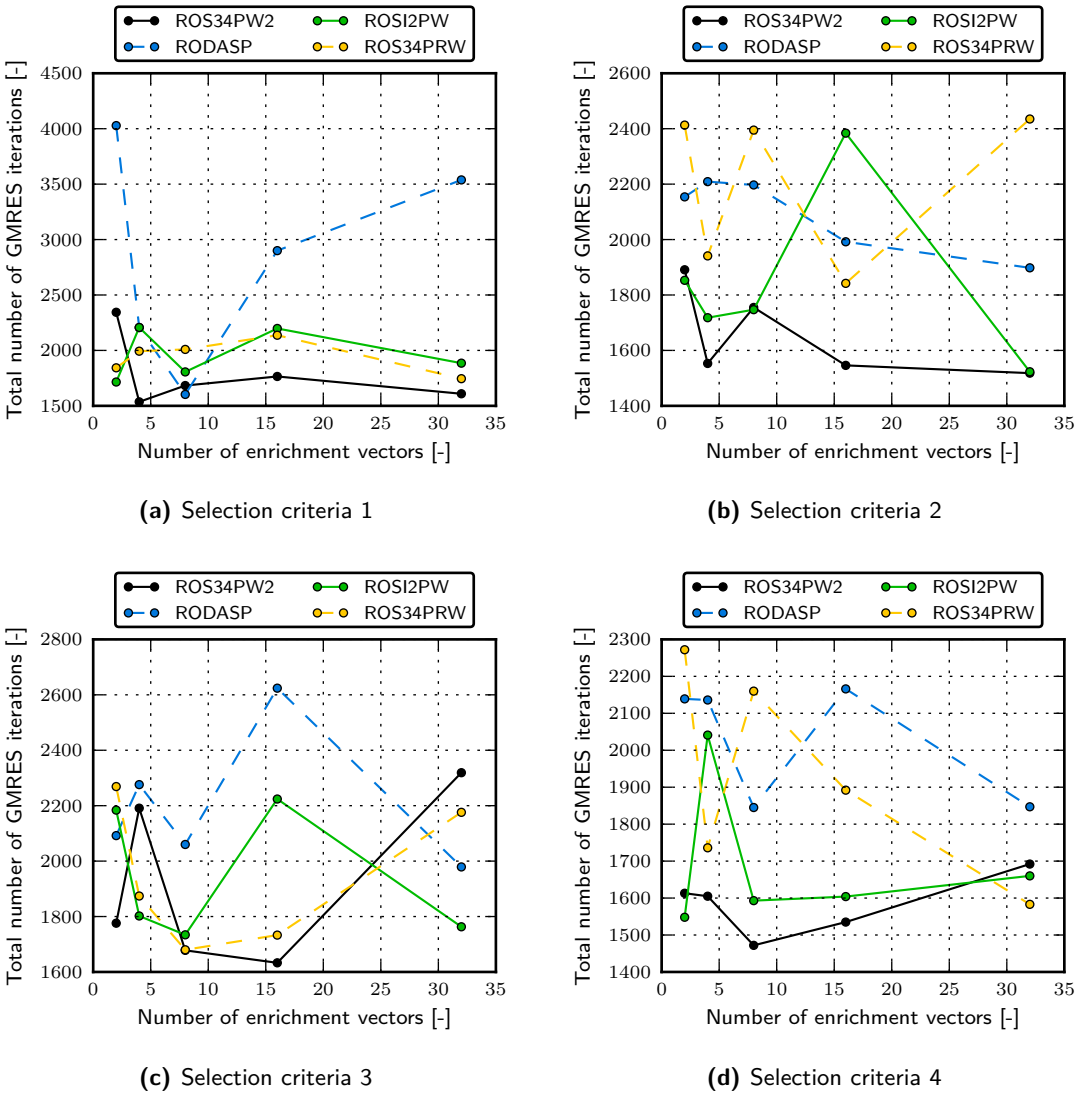


(c) Selection criteria 3



(d) Selection criteria 4

Figure 3.5: Uniform flow around a cylinder case: influence of the number of enrichment vectors and the used selection criteria on the total computational time for the Rosenbrock time integration schemes. The total CPU time is scaled with the total CPU of the standard GMRES algorithm. The plots show the results for the four selection criteria.



(a) Selection criteria 1

(b) Selection criteria 2

(c) Selection criteria 3

(d) Selection criteria 4

Figure 3.6: Uniform flow around a cylinder case: influence of the number of enrichment vectors and the used selection criteria on the total number of GMRES iterations for the Rosenbrock time integration schemes. The plots show the results for the four selection criteria.

is robust, i.e. the solution of the least squares problem always exists. GMRES does have full recurrences since the basis of the Krylov subspace needs to be stored completely. Also, orthogonalisation of new basis vectors becomes increasingly more expensive when the iterations proceed. GMRES minimizes the residual norm, as a result the residual norm decreases monotonically and the convergence is smooth.

The GMRES-E method uses augmentation of the Krylov subspace, and reuses information from previous GMRES cycles in order to speed up the convergence of the GMRES solver. This method is suitable for Rosenbrock time integration schemes, since the system matrix stays constant for the consecutive stages of the ROW-method. Rosenbrock schemes utilise the complete GMRES-E method, whereas ESDIRK schemes can only reuse Krylov subspace vectors of previous GMRES cycles by performing a matrix vector multiplication increasing the computational costs significantly.

The nonlinear convection-diffusion problem and the uniform flow around a cylinder test case are used to compare the performance of the GMRES-E algorithm with GMRES. The use of the GMRES-E algorithm reduces the number of GMRES iterations and total computational time significantly for the Rosenbrock time integration schemes. With an increasing number of enrichment vectors, the efficiency of GMRES-E is increased as well for the nonlinear convection-diffusion test case. The used selection criteria does not have a significant influence on the number of GMRES iterations, except when the third criterion is used which has a negative influence on the performance of GMRES. For the viscous test case, it is unclear which settings are optimal. The use of the GMRES-E algorithm for ESDIRK time integration schemes is not advised, since the number of GMRES iterations increased with an increasing number of enrichment vectors.

Chapter 4

Conclusions and recommendations

4.1 Conclusions

The objective of the research project is to improve state-of-the-art unsteady flow solvers by comparing high order Rosenbrock-Wanner time integration schemes with implicit Runge-Kutta schemes for non linear convection-diffusion problems and viscous flows, as mentioned in the first chapter. The main conclusions of the comparison of the different time integration schemes are summarised. Thereafter, conclusions are given for the use of Krylov subspace enrichment in a flow solver.

4.1.1 Time integration schemes for fluid dynamics

A gain in efficiency is observed for the Rosenbrock-Wanner time integration schemes when fixed and adaptive time steps are used to integrate the fluid dynamics equations, or the nonlinear convection-diffusion equation. A large gain in computational efficiency is observed in case an adaptive time step control algorithm is used. The ROW-schemes use less time steps in comparison with ESDIRK to solve the governing equations. The Rosenbrock schemes introduce a decrease in accuracy as is clearly observed for the convection-diffusion test case. This is also observed for the viscous flow problem.

The RODASP scheme outperforms the ESDIRK schemes in terms of computational efficiency, but is also susceptible to order reduction for loose tolerance settings for the linear solver. ROS34PRW is also computationally more efficient than ESDIRK3, and is less susceptible to order reduction compared with RODASP. However, it was necessary for the computations performed in this thesis to use a second order finite difference quotient for the matrix vector product used

by the Jacobian-free Newton-Krylov method when a Rosenbrock time integration scheme is used. With a first order approximation, instabilities in the matrix vector product caused the GMRES algorithm to diverge.

Combining an adaptive time step selection algorithm with the Rosenbrock time integration schemes is a possible solution for problems caused by instability of the time integration scheme, especially when relatively large time steps are desired for the time integration. However, for the two test cases studied in this thesis, the computational efficiency of the Rosenbrock time integration scheme is not improved. The computational efficiency of the ESDIRK scheme reduces with the use of an adaptive time step control algorithm, due to a large number of time steps being rejected.

4.1.2 Krylov subspace enrichment

The reuse of the Krylov subspace vectors for the subsequent stages of the Rosenbrock time integration scheme accelerates the converge of the GMRES solver for the non linear convection-diffusion test case, and also for the uniform flow around a cylinder test case. The preprocessing step of the GMRES-E algorithm has the biggest impact on the computational efficiency. By increasing the number of enrichment vectors, the computational efficiency is further increased for the convection-diffusion test case. The used selection criteria does not have a significant influence on the convergence of the GMRES algorithm, except for the third criterion which is less efficient than criteria 1, 2 and 4. It is unclear which number of enrichment vectors and selection criterion results in the best performance for the viscous flow case.

The use of the GMRES-E algorithm for ESDIRK time integration schemes is not advised, since the number of GMRES iterations increased with an increasing number of enrichment vectors. This is caused by the fact that the system matrix $\mathbf{A}$ changes at every Newton iteration, and also at every stage of the ESDIRK scheme.

4.2 Recommendations

Based on the outcome of the performed research some recommendations are given for further research. The Rosenbrock-Wanner time integration schemes proved to be computationally more efficient with ESDIRK time integration schemes for the cases considered in this thesis. So far only laminar flows are studied. Therefore, the comparison of the different schemes in terms of efficiency and accuracy for turbulent flows is a further step. Also, a grid convergence study needs to performed since the test cases studied in this thesis are exactly solved in time, but inaccuracies exist in space.

Rosenbrock time integration schemes combined with moving meshes is also not considered in

this thesis. This is the next step if it is desired to employ a ROW-scheme in a fluid-structure interaction simulation. It is advised to follow the same approach used in this thesis, namely to first apply the method on a relatively simple test case such as a diffusion-convection problem.

Regarding the inclusion of enrichment vectors in the Krylov subspace, further tests are also necessary to confirm the gain in efficiency for the GMRES-E algorithm. As mentioned, turbulent flows are not considered in this thesis, and only one grid is used for the viscous test case. Also, it may be possible to use the GMRES-E algorithm for the ESDIRK in case the Jacobian used by the Newton-Krylov method is kept constant for the consecutive stages of the scheme. In this way, it is possible to utilise all the elements of the GMRES-E algorithm.

Further research is also necessary for the selection of the enrichment vectors of the GMRES-E algorithm. Probably with the use of an error estimator for the computed Ritz vectors, the performance of GMRES-E can be improved.

Bibliography

Baker, A. H., Jessup, E. R., and Kolev, T. V. (2009). 'A simple strategy for varying the restart parameter in GMRES(m)'. *Journal of Computional and Applied Mathematics*, **230**(2), 751–761.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Van der Vorst, H. A. (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM.

Bijl, H. and Carpenter, M. H. (2005). 'Iterative solution techniques for unsteady flow computations using higher order time integration schemes'. *International Journal for numerical methods in fluids*, **47**, 857–862.

Bijl, H., Carpenter, M. H., Vatsa, V. N., and Kennedy, C. A. (2002). 'Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow'. *Journal of Computational Physics*, **179**(1), 313–329.

Björck, A. (1996). *Numerical Methods for Least Squares Problems*. SIAM: Philadelphia, PA.

Blom, D. S. (2012). 'Rosenbrock time integration in combination with fluid-structure interaction - Unsteady aerodynamics'. Literature Study, Delft University of Technology.

Butcher, J. C. (2003). *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons.

Calahan, D. (1968). 'A stable, accurate method of numerical integration for nonlinear systems'. *Proceedings of the IEEE*, **56**(4), 744. ISSN 0018-9219.

Carpenter, M. H., Vuik, C., Lucas, P., van Gijzen, M., and Bijl, H. (2010). 'A General Algorithm for Reusing Krylov Subspace Information. I. Unsteady Navier-Stokes'. Technical report, National Aeronautics and Space Administration, Langley Research Center, Delft University of Technology.

Eiermann, M., Ernst, O. G., and Schneider, O. (2000). 'Analysis of acceleration strategies for restarted minimal residual methods'. *Journal of Computional and Applied Mathematics*, **123**(1-2), 261–292.

Eisenstat, S. C. and Walker, H. F. (1994). 'Choosing the Forcing Terms in an Inexact Newton Method'. *SIAM Journal on Scientific Computing*, **17**, 16–32.

Ellsiepen, P. (1999). *Zeits- und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poröser Medien*. Ph.D. thesis, University of Stuttgart, Institute of Mechanics II.

Elman, H. C., Silvester, D. J., and Wathen, A. J. (2005). *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics (Numerical Mathematics and Scientific Computat)*. Oxford University Press.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, 3rd edition.

Greenbaum, A. (1987). *Iterative Methods for Solving Linear Systems (Frontiers in Applied Mathematics)*. Society for Industrial and Applied Mathematics.

Greenbaum, A. (1997). 'Estimating the attainable accuracy of recursively computed residual methods'. *SIAM Journal on Matrix Analysis and Applications*, **18**(3), 535–551.

Gustafsson, K. (1991). 'Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods'. *ACM Transactions on Mathematical Software*, **17**(14), 533–554.

Gustafsson, K. (1994). 'Control-Theoretic Techniques for Stepsize Selection in Implicit Runge-Kutta Methods'. *ACM Transactions on Mathematical Software*, **20**(4), 496–517.

Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II Stiff and Differential-Algebraic Problems*. Springler-Verlag, 2nd edition.

John, V. and Rang, J. (2010). 'Adaptive time step control for the incompressible Navier-Stokes equations'. *Computer Methods in Applied Mechanics and Engineering*, **199**(9-12), 514–524.

Jothiprasad, G., Mavriplis, D. J., and Caughey, D. A. (2003). 'Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes'. *Journal of Computational Physics*, **191**(2), 542–566.

Joubert, W. D. (1994). 'On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems'. *Numerical Linear Algebra with Applications*, **1**, 427–447.

Kelley, C. T. (1995). *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics.

Kennedy, C. A. and Carpenter, M. H. (2003). 'Additive Runge-Kutta schemes for convection-diffusion-reaction equations'. *Applied Numerical Mathematics*, **44**(1-2), 139–181.

Knoll, D. A. and Keyes, D. E. (2004). 'Jacobian-free Newton-Krylov methods: a survey of approaches and applications'. *Journal of Computational Physics*, **193**, 357–397.

Lanczos, C. (1950). 'An iteration method for the solution of the eigenvalue problem of linear differential and integral operators'. *Journal of Research of the National Bureau of Standards*, **45**(4), 255–282.

Lang, J. (1999). 'Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems. Theory, Algorithm, and Applications'. Technical Report SC-99-20, ZIB, Takustr.7, 14195 Berlin.

Lucas, P. (2010). *Efficient numerical methods to compute unsteady subsonic flows on unstructured grids*. Ph.D. thesis, Delft University of Technology.

Morgan, R. B. (1995). 'A restarted GMRES method augmented with eigenvectors'. *SIAM Journal on Matrix Analysis and Applications*, **16**(4), 1154–1171.

Morgan, R. B. (2000). 'Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations'. *SIAM Journal on Matrix Analysis and Applications*, **21**(4), 1112–1135.

Morgan, R. B. (2002). 'GMRES with deflated restarting'. *SIAM Journal on Scientific Computing*, **24**(1), 20–37.

Morgan, R. B. and Zeng, M. (1998). 'Harmonic projection methods for large non-symmetric eigenvalue problems'. *Numerical Linear Algebra with Applications*, **5**(1), 33–55.

Netlib (2012). 'LAPACK - Linear Algebra PACKage'. `http://www.netlib.org/lapack/` (last checked November 16, 2012).

Nicolaides, R. A. (1987). 'Deflation of conjugate gradients with applications to boundary value problems'. *SIAM Journal on Numerical Analysis*, **24**(2), 355–365.

Paige, C. C., Rozložník, M., and Strakos, Z. (2006). 'Modified Gram-Schmidt (MGS), Least Squares and Backward Stability of MGS-GMRES'. *SIAM Journal on Matrix Analysis and Applications*, **28**(1), 264–284.

Qin, N., Ludlow, D. K., and Shaw, S. T. (2000). 'A matrix-free preconditioned newton/gmres method for unsteady navier-stokes solutions'. *International Journal for numerical methods in fluids*, **33**, 223–248.

Rang, J. (2004). *Stability estimates and numerical methods for degenerate parabolic differential equations*. Ph.D. thesis, Mathematisch-Naturwissenschaftlichen Fakultät der Technischen Universität Clausthal.

Rang, J. (2013). 'A new stiffly accurate Rosenbrock-Wanner method for solving the incompressible Navier-Stokes equations'. In 'Recent Developments in the Numerics of Nonlinear Hyperbolic Conservation Laws', volume 120, pages 301–315. Springer Berlin Heidelberg.

Rang, J. and Angermann, L. (2005). 'New Rosenbrock W-Methods of Order 3 for Partial Differential Algebraic Equations of Index 1'. *BIT Numerical Mathematics*, **45**(4), 761–787.

Rang, J. and Angermann, L. (2006). 'New Rosenbrock methods of order 3 for PDAEs of index 2'. *BIT Numerical Mathematics*.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2nd edition.

Saad, Y. and Schultz, M. H. (1986). 'GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems'. *SIAM Journal on Scientific Computing*, **7**(3), 856–869.

Simoncini, V. and Szyld, D. B. (2007). 'Recent computational developments in Krylov subspace methods for linear systems'. *Numerical Linear Algebra with Applications*, **14**, 1–59.

Söderlind, G. (2002). 'Automatic control and adaptive time-stepping'. *Numerical Algorithms*, **31**(1-4), 281–310.

Söderlind, G. (2003). 'Digital filters in adaptive time-stepping'. *ACM Transactions on Mathematical Software*, **29**(1), 1–26.

Söderlind, G. and Wang, L. (2006a). 'Adaptive time-stepping and computational stability'. *Journal of Computational and Applied Mathematics*, **185**(2), 225–246.

Söderlind, G. and Wang, L. (2006b). 'Evaluating numerical ODE/DAE methods, algorithms and software'. *Journal of Computational and Applied Mathematics*, **185**(2), 244–260.

De Sturler, E. (1999). 'Truncation strategies for optimal krylov subspace methods'. *SIAM Journal on Numerical Analysis*, **36**(3), 864–889.

University of Washington (1940a). 'Opening day of the Tacoma Narrows Bridge, July 1, 1940'. **URL:** `http://content.lib.washington.edu/cdm4/item_viewer.php?CISOROOT=/farquharson&CISOPTR=13` (last checked April 4, 2012).

University of Washington (1940b). 'Tacoma Narrows Bridge midsection collapsing into the waters of the Tacoma Narrows, November 7, 1940'. **URL:** `http://content.lib.washington.edu/cdm-ayp/item_viewer.php?CISOROOT=%2Ffarquharson&CISOPTR=19&DMSCALE=100&DMWIDTH=802&DMHEIGHT=652.66927083333&DMMODE=viewer&DMFULL=1&DMX=0&DMY=0&DMTEXT=%2520FAR175%2520FAR017&DMTHUMB=0&REC=1&DMROTATE=0&x=414&y=312` (last checked April 4, 2012).

Van E. Henson, editor (2003). *Multigrid methods for nonlinear problems: an overview*, volume 5016. Lawrence Livermore National Lab. (USA), SPIE.

Varga, R. S. (2000). *Matrix Iterative Analysis*. Springler-Verlag Berlin Heidelberg, 2nd edition.

Van der Vorst, H. A. (2003). *Iterative Krylov Methods for Large Linear Systems (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press.

Walker, H. F. (1988). 'Implementation of the gmres method using householder transformations'. *SIAM Journal on Scientific Computing*, **9**(1), 152–163.

Wang, L. and Mavriplis, D. J. (2007). 'Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations'. *Journal of Computational Physics*, **225**(2), 1994–2015.

Widlund, O. B. (1967). 'A note on unconditionally stable linear multistep methods'. *BIT Numerical Mathematics*, **7**(1), 65–70.

Van Zuijlen, A. H. (2006). *Fluid-Structure Interaction Simulations - Efficient Higher Order Time Integration of Partitioned Systems*. Ph.D. thesis, Delft University of Technology.

Appendix A

Butcher tableaus for the SDIRK, ESDIRK and ROW schemes

The used coefficients for the SDIRK2, ESDIRK3, ESDIRK4, ESDIRK5, ROS34PW2, ROSI2PW2, ROS34PRW and RODASP schemes can be found in Tables A.1 - A.8. The tables include the embedded schemes of the different methods.

Table A.1: Butcher tableau for the method of Ellsiepen (SDIRK2), where $\alpha = 1 - \sqrt{2}/2$, $\hat{\alpha} = 2 - \frac{5}{4}\sqrt{2}$ and $\alpha - \hat{\alpha} = -1 + \frac{3}{4}\sqrt{2}$ (Ellsiepen, 1999).

α	α	0
1	$1-\alpha$	α
b_i	$1-\alpha$	α
$\hat{b_i}$	$1-\hat{\alpha}$	$\hat{\alpha}$
$b_i - \hat{b_i}$	$\hat{\alpha} - \alpha$	$\alpha - \hat{\alpha}$

Table A.2: Butcher tableau for ESDIRK3 (Kennedy and Carpenter, 2003)

c_1	0	0	0	0
c_2	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
c_3	$\frac{2746238789719}{10658868560708}$	$-\frac{640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
c_4	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
b_i	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
$\hat{b}_i$	$\frac{2756255671327}{12835298489170}$	$-\frac{10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

Table A.3: Butcher tableau for ESDIRK4 (Kennedy and Carpenter, 2003)

c_1	0	0	0	0	0	0
c_2	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
c_3	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
c_4	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
c_5	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
c_6	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
$\hat{b}_i$	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

Table A.4: Butcher tableau for ESDIRK5 (Kennedy and Carpenter, 2003)

c_1	0	0	0	0	0	0	0	0
c_2	$\frac{41}{200}$	$\frac{41}{200}$	0	0	0	0	0	0
c_3	$\frac{41}{200}$	$-\frac{567603406766}{11931857230679}$	$\frac{41}{200}$	0	0	0	0	0
c_4	$\frac{683785636431}{9252920307686}$	0	$-\frac{110385047103}{1367015193373}$	$\frac{41}{200}$	0	0	0	0
c_5	$\frac{3016520224154}{10081342136671}$	0	$\frac{30586259806659}{12414158314087}$	$-\frac{22760509404356}{11113319521817}$	$\frac{41}{200}$	0	0	0
c_6	$\frac{218866479029}{1489978393911}$	0	$\frac{638256894668}{5436446318841}$	$-\frac{1179710474555}{5321154724896}$	$-\frac{60928119172}{8023461067671}$	$\frac{41}{200}$	0	0
c_7	$\frac{1020004230633}{5715676835656}$	0	$\frac{25762820946817}{25263940353407}$	$-\frac{2161375909145}{9755907335909}$	$-\frac{211217309593}{5846859502534}$	$-\frac{4269925059573}{7827059040749}$	$\frac{41}{200}$	0
c_8	$-\frac{872700587467}{9133579230613}$	0	0	$\frac{22348218063261}{9555858737531}$	$-\frac{1143369518992}{8141816002931}$	$-\frac{39379526789629}{19018526304540}$	$\frac{32727382324388}{42900044865799}$	$\frac{41}{200}$
b_i	$-\frac{872700587467}{9133579230613}$	0	0	$\frac{22348218063261}{9555858737531}$	$-\frac{1143369518992}{8141816002931}$	$-\frac{39379526789629}{19018526304540}$	$\frac{32727382324388}{42900044865799}$	$\frac{41}{200}$
$\hat{b}_i$	$-\frac{975461918565}{9796059967033}$	0	0	$\frac{78070527104295}{32432590147079}$	$-\frac{548382580838}{3424219808633}$	$-\frac{33438840321285}{15594753105479}$	$\frac{3629800801594}{4656183773603}$	$\frac{4035322873751}{18575991585200}$

Table A.5: Set of coefficients for Rosenbrock ROS34PW2, a stiffly accurate W-method of order three for PDAES of index one (Rang and Angermann, 2005).

γ	$=$	$4.3586652150845900 \times 10^{-1}$			
α_{21}	$=$	$8.7173304301691801 \times 10^{-1}$	γ_{21}	$=$	$-8.7173304301691801 \times 10^{-1}$
α_{31}	$=$	$8.4457060015369423 \times 10^{-1}$	γ_{31}	$=$	$-9.0338057013044082 \times 10^{-1}$
α_{32}	$=$	$-1.1299064236484185 \times 10^{-1}$	γ_{32}	$=$	$5.4180672388095326 \times 10^{-2}$
α_{41}	$=$	$0.0000000000000000 \times 10^{+0}$	γ_{41}	$=$	$2.4212380706095346 \times 10^{-1}$
α_{42}	$=$	$0.0000000000000000 \times 10^{+0}$	γ_{42}	$=$	$-1.2232505839045147 \times 10^{+0}$
α_{43}	$=$	$1.0000000000000000 \times 10^{+0}$	γ_{43}	$=$	$5.4526025533510214 \times 10^{-1}$
b_1	$=$	$2.4212380706095346 \times 10^{-1}$	$\hat{b}_1$	$=$	$3.7810903145819369 \times 10^{-1}$
b_2	$=$	$-1.2232505839045147 \times 10^{+0}$	$\hat{b}_2$	$=$	$-9.6042292212423178 \times 10^{-2}$
b_3	$=$	$1.5452602553351020 \times 10^{+0}$	$\hat{b}_3$	$=$	$5.0000000000000000 \times 10^{-1}$
b_4	$=$	$4.3586652150845900 \times 10^{-1}$	$\hat{b}_4$	$=$	$2.1793326075422950 \times 10^{-1}$

Table A.6: Set of coefficients for Rosenbrock ROSI2PW and its embedded method, a stiffly accurate W-method of order three for PDAES of index two (Rang and Angermann, 2006).

γ	$=$	$4.3586652150845900 \times 10^{-1}$			
α_{21}	$=$	$8.7173304301691801 \times 10^{-1}$	γ_{21}	$=$	$-8.7173304301691801 \times 10^{-1}$
α_{31}	$=$	$-7.9937335839852708 \times 10^{-1}$	γ_{31}	$=$	$3.0647867418622479 \times 10^{+0}$
α_{32}	$=$	$-7.9937335839852708 \times 10^{-1}$	γ_{32}	$=$	$3.0647867418622479 \times 10^{+0}$
α_{41}	$=$	$7.0849664917601007 \times 10^{-1}$	γ_{41}	$=$	$-1.0424832458800504 \times 10^{-1}$
α_{42}	$=$	$3.1746327955312481 \times 10^{-1}$	γ_{42}	$=$	$-3.1746327955312481 \times 10^{-1}$
α_{43}	$=$	$-2.5959928729134892 \times 10^{-2}$	γ_{43}	$=$	$-1.4154917367329144 \times 10^{-2}$
b_1	$=$	$6.0424832458800504 \times 10^{-1}$	$\hat{b}_1$	$=$	$4.4315753191688778 \times 10^{-1}$
b_2	$=$	$-3.6210810811598324 \times 10^{-32}$	$\hat{b}_2$	$=$	$4.4315753191688778 \times 10^{-1}$
b_3	$=$	$-4.0114846096464034 \times 10^{-2}$	$\hat{b}_3$	$=$	$0.0000000000000000 \times 10^{+0}$
b_4	$=$	$4.3586652150845900 \times 10^{-1}$	$\hat{b}_4$	$=$	$1.1368493616622447 \times 10^{-1}$

Table A.7: Set of coefficients for Rosenbrock ROS34PRW and its embedded method, a stiffly accurate W-method of order three for PDAES of index two (Rang, 2013).

γ	$=$	$4.3586652150845900 \times 10^{-1}$			
α_{21}	$=$	$8.7173304301691801 \times 10^{-1}$	γ_{21}	$=$	$-8.7173304301691801 \times 10^{-1}$
α_{31}	$=$	$1.4722022879435914 \times 10^{+0}$	γ_{31}	$=$	$-1.2855347382089872 \times 10^{+0}$
α_{32}	$=$	$-3.1840250568090289 \times 10^{-1}$	γ_{32}	$=$	$5.0507005541550687 \times 10^{-1}$
α_{41}	$=$	$8.1505192016694938 \times 10^{-1}$	γ_{41}	$=$	$-4.8201449182864348 \times 10^{-1}$
α_{42}	$=$	$5.0000000000000000 \times 10^{-1}$	γ_{42}	$=$	$2.1793326075422950 \times 10^{-1}$
α_{43}	$=$	$-3.1505192016694938 \times 10^{-1}$	γ_{43}	$=$	$-1.7178529043404503 \times 10^{-1}$
b_1	$=$	$3.3303742833830591 \times 10^{-1}$	$\hat{b}_1$	$=$	$2.5000000000000000 \times 10^{-1}$
b_2	$=$	$7.1793326075422947 \times 10^{-1}$	$\hat{b}_2$	$=$	$7.4276119608319180 \times 10^{-1}$
b_3	$=$	$-4.8683721060099439 \times 10^{-1}$	$\hat{b}_3$	$=$	$-3.1472922970066219 \times 10^{-1}$
b_4	$=$	$4.3586652150845900 \times 10^{-1}$	$\hat{b}_4$	$=$	$3.2196803361747034 \times 10^{-1}$

Table A.8: Set of coefficients for Rosenbrock RODASP and its embedded method, a stiffly accurate method of order four for PDAES of index one (John and Rang, 2010).

γ	$=$	$2.5000000000 \times 10^{-1}$			
α_{21}	$=$	$7.5000000000 \times 10^{-1}$	γ_{21}	$=$	$-7.5000000000 \times 10^{-1}$
α_{31}	$=$	$8.6120400814 \times 10^{-2}$	γ_{31}	$=$	$-1.3551200000 \times 10^{-1}$
α_{32}	$=$	$1.2387959919 \times 10^{-1}$	γ_{32}	$=$	$-1.3799200000 \times 10^{-1}$
α_{41}	$=$	$7.7403453551 \times 10^{-1}$	γ_{41}	$=$	$-1.2560800000 \times 10^{+0}$
α_{42}	$=$	$1.4926515495 \times 10^{-1}$	γ_{42}	$=$	$-2.5014500000 \times 10^{-1}$
α_{43}	$=$	$-2.9419969046 \times 10^{-1}$	γ_{43}	$=$	$1.2209300000 \times 10^{+0}$
α_{51}	$=$	$5.3087466826 \times 10^{+0}$	γ_{51}	$=$	$-7.0731800000 \times 10^{+0}$
α_{52}	$=$	$1.3308921400 \times 10^{+0}$	γ_{52}	$=$	$-1.8056500000 \times 10^{+0}$
α_{53}	$=$	$-5.3741378117 \times 10^{+0}$	γ_{53}	$=$	$7.7438300000 \times 10^{+0}$
α_{54}	$=$	$-2.6550101103 \times 10^{-1}$	γ_{54}	$=$	$8.8500300000 \times 10^{-1}$
α_{61}	$=$	$-1.7644376488 \times 10^{+0}$	γ_{61}	$=$	$1.6840700000 \times 10^{+0}$
α_{62}	$=$	$-4.7475655721 \times 10^{-1}$	γ_{62}	$=$	$4.1826600000 \times 10^{-1}$
α_{63}	$=$	$2.3696918469 \times 10^{+0}$	γ_{63}	$=$	$-1.8814100000 \times 10^{+0}$
α_{64}	$=$	$6.1950235906 \times 10^{-1}$	γ_{64}	$=$	$-1.1378600000 \times 10^{-1}$
α_{65}	$=$	$2.5000000000 \times 10^{-1}$	γ_{65}	$=$	$-3.5714300000 \times 10^{-1}$
b_1	$=$	$-8.0368370789 \times 10^{-2}$	$\hat{b}_1$	$=$	$-1.7644376488 \times 10^{+0}$
b_2	$=$	$-5.6490613592 \times 10^{-2}$	$\hat{b}_2$	$=$	$-4.7475655721 \times 10^{-1}$
b_3	$=$	$4.8828563004 \times 10^{-1}$	$\hat{b}_3$	$=$	$2.3696918469 \times 10^{+0}$
b_4	$=$	$5.0571621148 \times 10^{-1}$	$\hat{b}_4$	$=$	$6.1950235906 \times 10^{-1}$
b_5	$=$	$-1.0714285714 \times 10^{-1}$	$\hat{b}_5$	$=$	$2.5000000000 \times 10^{-1}$
b_6	$=$	$2.5000000000 \times 10^{-1}$	$\hat{b}_5$	$=$	$0.0000000000 \times 10^{+0}$