Exponential Integrators 
and their application to solving Richards’ Equation

Elliot Carr 
BMaths 

under the supervision of 
Professor Ian Turner

School of Mathematical Sciences 
Faculty of Science 
Queensland University of Technology

November, 2008
Assessment Criteria Weightings

The table below shows the weightings that are to be applied to the assessment criteria for this project.

<table>
<thead>
<tr>
<th>CRITERIA</th>
<th>WEIGHTING (fraction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review and investigation</td>
<td>(W1) 0.35</td>
</tr>
<tr>
<td>Interpretation and synthesis (Must be $\geq 0.3$)</td>
<td>(W2) 0.35</td>
</tr>
<tr>
<td>Written communication</td>
<td>(W3) 0.2</td>
</tr>
<tr>
<td>Oral communication</td>
<td>(W4) 0.1</td>
</tr>
</tbody>
</table>

Note: $W1 + W2 + W3 + W4$ should add up to 1.

I agree that the above weightings are in accordance with those pre-assigned in the unit outline and those chosen by the author of this project in consultation with the principal supervisor for the project.

Signature of author:

Signature of principal supervisor:
Abstract

The ability to predict the movement of water in unsaturated porous media is vital to applications of groundwater flow and is commonly achieved with Richards’ equation. Obtaining a numerical solution to the governing partial differential equation is difficult, not only due to the nonlinearities present in the equation, but also the sharp fronts that feature in the solutions. Typically, spatially discrete solutions are advanced implicitly in time, using backward differentiation approximations to the time derivative. These implementations, however, often encounter mass-balance and convergence problems, motivating the need for improved time integration techniques.

We investigated the use of exponential integrators applied to Richards’ equation in a two-dimensional setting. These schemes are purely explicit and based on the classical Euler and higher-order Rosenbrock integrators. Both formulae require the computation of the matrix function \( \varphi(A) = A^{-1}(e^A - I) \) where \( A \) is large and sparse and we presented two techniques for approximating \( \varphi(A)v \) for a vector \( v \) as required in each integration scheme. The first method is based on best rational approximations to \( \varphi \) and relies on the efficient solution of a number of shifted linear systems of the form \( (A - \zeta I)x = v \) for a complex scalar \( \zeta \). We proposed and compared some GMRES-like strategies to deal with these types of equations. The second method is concerned with projecting the large matrix \( A \) onto a small Krylov subspace so that the only evaluation of \( \varphi \) is performed on a matrix of much smaller dimension.

It was found that using GMRES strategies embedded in best rational approximants yielded methods that were competitive with the established Krylov techniques for computing \( \varphi(A)v \). Our numerical investigations concluded that exponential Euler outperformed the higher-order exponential Rosenbrock scheme for the chosen two-dimensional Richards’ equation problem. In addition, it was shown that both of these exponential schemes convincingly outperformed the established backward Euler method for the test problem.
I hereby declare that this submission is my own work and to the best of my knowledge it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at QUT or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at QUT or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project’s design and conception or in style, presentation and linguistic expression is acknowledged.

Elliot Carr
Acknowledgements

First of all, I wish to thank Professor Ian Turner, my supervisor, for all the help he has provided over the past year. Those Friday afternoons spent talking mathematics have given me great pleasure and made the whole honours experience that much more enjoyable.

I must also thank the School of Mathematical Sciences for their generous support in awarding me an Honours scholarship and to all the staff who have lectured me over the past four years. A special thanks to my high school mathematics teacher Mr Ian Lindsay, who encouraged me to study mathematics at university.

Lastly, but most importantly to my family. Although, you may not understand much of this thesis, without your support none of it would have been possible. Thank you.
## Contents

Assessment Criteria Weightings  
Abstract  
Declaration  
Acknowledgements  

Chapter 1 Introduction  
  1.1 Motivation  
  1.2 Literature Review  
  1.3 Aims and Objectives  
  1.4 Outline of Chapters  

Chapter 2 Modelling flow through porous media using Richards’ equation  
  2.1 Richards’ equation  
  2.2 Test Problem  
  2.3 Finite Volume Method  
    2.3.1 Mesh structure  
    2.3.2 Finite Volume equations  
    2.3.3 Boundary conditions  
  2.4 Summary of the spatial discretisation  

Chapter 3 Time stepping with Exponential Integrators  
  3.1 The exponential Euler method  
  3.2 The exponential Rosenbrock method  
  3.3 Variable stepsize implementation  

Chapter 4 Approximation of the $\varphi$ function for Exponential Integrators  
  4.1 Best rational approximants  
    4.1.1 Constructing approximants  
    4.1.2 Applicability to Richards’ equation  
    4.1.3 GMRES-like methods for the shifted linear systems  
    4.1.4 Practical implementation of GMRES-like methods  
    4.1.5 Investigation into the convergence of GMRES-like methods  
    4.1.6 Strategies using a single Krylov subspace  
  4.2 Krylov subspace approximants  
    4.2.1 Recurrence relations for the $\varphi$ function  
    4.2.2 Padé approximation  
    4.2.3 Practical implementation  
  4.3 Implementation issues  

Chapter 5 Results  
  5.1 Numerical solution to Test problem  

  

5.2 Numerical experiments .............................................. 55
  5.2.1 Exponential Euler method .................................... 56
  5.2.2 Exponential Rosenbrock method ............................ 59
5.3 Comparison with an implicit scheme: the backward Euler method ... 60

Chapter 6 Conclusions ................................................. 65
  6.1 Summary ............................................................... 65
  6.2 Contributions ...................................................... 66
  6.3 Future Work ....................................................... 66

References ............................................................... 68
List of Tables

2.1 Hydraulic properties of sand and clay ........................................... 10
2.2 Midpoint quadrature errors for double integrals ............................. 13
4.1 Best rational approximant to $\varphi(z)$ for $z \leq 0$ ........................... 29
5.1 Spatial discretisations used in the numerical experiments .................. 55
5.2 Performance of the exponential Euler scheme on the test problem with a fixed time step implemented with different strategies for computing the $\varphi$ function ................................................................. 57
5.3 Performance of the exponential Euler scheme on the test problem with adaptive time stepping implemented with different strategies for computing the $\varphi$ function ................................................................. 59
5.4 Performance of the exponential Rosenbrock scheme on the test problem with fixed and adaptive time stepping ................................. 60
5.5 Performance of the exponential Euler, exponential Rosenbrock and backward Euler schemes on the test problem ................................. 63
5.6 Performance of the exponential Euler, exponential Rosenbrock and backward Euler schemes on a stiff variant of the test problem ................. 64
# List of Figures

2.1 Schematic diagram of the test problem ............................................. 9
2.2 Alignment of nodes and control volumes for the test problem ............... 12
2.3 Notation used in the finite volume method ....................................... 13
4.1 Sparsity structure of the Jacobian matrix for the test problem ............... 24
4.2 Applicability of best rational approximants for computing $\varphi$ for the test problem ................................................................. 28
4.3 Example of the convergence rates of different variants of GMRES for the numerical solution of the shifted linear systems that arise in best rational approximants to the $\varphi$ function ............................................. 44
5.1 Numerical solution to the test problem ............................................ 54
5.2 Time step evolution of the exponential Euler, exponential Rosenbrock and backward Euler schemes for the test problem ......................... 63
5.3 Time step evolution of the exponential Euler and exponential Rosenbrock schemes for a stiff variant of the test problem ............................. 64
Chapter 1

Introduction

Exponential integrators are numerical schemes used to approximate the solution of initial value problems governed by systems of ordinary differential equations. In this introductory chapter we provide motivation for applying these methods to the discretised form of Richards’ equation and present a review of their implementation in the literature. The aims and objectives of this thesis are then outlined along with a summary of the contents of each chapter.

1.1 Motivation

In applied mathematics, physical scenarios are often modelled using time and space dependent partial differential equations (PDE). The solution of these equations provides researchers with valuable insight into the underlying process and often theoretically enforces experimentally observed phenomena. However, due to the complexity of these processes and the irregular geometries of some problems, exact analytic solutions to the governing equations are almost always not accessible. Generally in this case, numerical strategies are called upon to provide an approximate solution.

Typically a spatial discretisation is performed over the geometry, producing a discrete analogue of the PDE in the form of a system of ordinary differential equations (ODE). In order, to advance this discrete spatial solution in time, the solution of this ODE system is required. It is at this point Schatzman (2002) claims that with the introduction of better time integration technology for these ODE systems, the largest performance improvement in the solution of PDEs will be observed.

In many applications the resulting system of ODEs are stiff. Richards’ equation is one such example (Kavetski, Binning & Sloan 2001). Generally such problems require an implicit scheme (Berland, Skaflestad & Wright 2005) as stiffness effectively renders explicit integrators useless, with stability rather than accuracy governing how the integrator performs. However, implicit methods to solve the \( h \)-based form of Richards’ equation

\[
C(h) \frac{\partial h}{\partial t} = \nabla \cdot (K(h) \nabla h) + \frac{\partial K(h)}{\partial z},
\]

(1.1)
can suffer from mass-balance error, convergence problems and poor CPU efficiency (McBride, Cross, Croft, Bennett & Gebhardt 2006). Celia, Bouloutas & Zarba (1990) explain that the problem lies in the approximation of the term \( C(h) \frac{\partial h}{\partial t} \) which is equivalent to the
change in moisture content $\frac{\partial \theta}{\partial t}$. While these terms are equivalent in the continuous sense, due to the highly nonlinear nature of the function $C(h)$ their discrete analogues are not. This leads to significant mass-balance errors using standard time-integration techniques and motivates the search for alternate time integration strategies. In this thesis we investigate integrators for the spatially discrete $h$-based form that avoid using discrete approximations to the time derivative.

1.2 Literature Review

Exponential integrators are explicit time stepping schemes for initial value problems of the form

$$\frac{du}{dt} = G(u) \quad u(0) = u_0,$$

(1.2)

where $u \in \mathbb{R}^N$. These methods use the exponential function (or closely related functions) of the Jacobian inside the numerical scheme (Minchev & Wright 2005). Recent literature by Minchev (2004), Minchev & Wright (2005), Schmelzer & Trefethen (2007) and Hochbruck, Lubich & Selhofer (1996) suggests that exponential integrators have become popular for the integration of stiff spatially discrete forms of PDEs. In fact, these integrators were introduced to overcome the phenomenon of stiffness (Minchev & Wright 2005). Advantageously, they do not experience problems typically associated with standard explicit schemes, where to ensure stability of the solution the time step must be constrained to impractical small values (Minchev 2004).

In this thesis the time integration of Richards’ equation using exponential integrators is investigated. As far as we are aware, the only time this has been attempted in the literature is in the work carried out by Pestana (2007) where an exponentially fitted Euler scheme was shown to outperform the backward Euler method on a one-dimensional problem. Our research focuses on two integrators featured in the literature that are extensions of common standard explicit schemes. We aim to determine their suitability to Richards’ equation in a two-dimensional setting.

The first method has origins in a paper written by Pope (1963). This scheme is the simplest example of an exponential integrator and consequently has appeared repeatedly in the literature since, albeit under various disguises. Common names have included the exponentially-fitted Euler method (Hochbruck et al. 1996), the exponential difference equation (Pope 1963), the local linearisation method (Jimenez, Biscay, Mora & Rodriguez 2002) and the exponential Euler method (Minchev & Wright 2005). Throughout this thesis we refer to this method as the latter, which is in line with the review work by Minchev & Wright (2005). A second method for the integration of (1.2), referred to in this thesis as the exponential Rosenbrock method, is also available (Hochbruck et al. 1996).

In both integrators, evaluation of the following matrix function is required

$$\varphi(A) = A^{-1}(e^A - I),$$

(1.3)
where $A$ is related to the time step and Jacobian matrix of $G$ and is typically large and sparse. According to Minchev & Wright (2005), the main computational challenge in the implementation of any exponential integrator is the need for fast and computationally stable evaluations of $\varphi(A)$ or $\varphi(A)v$ for some suitably defined vector $v$. Numerous methods have been proposed in the literature for this approximation and naturally have been derived from techniques constructed for the matrix exponential. One popular example is the rational Padé approximant with scaling and squaring as featured in the exponential integrators package by Berland et al. (2005). However, the effectiveness of the method depends on the norm of $A$; when $A$ has a large norm, this method becomes expensive and less accurate. This approach is also restricted to matrices of moderate dimension because computing $\varphi(A)v$ requires the explicit computation of $\varphi(A)$ (Schmelzer & Trefethen 2007). It is for this reason that direct use of Padé approximations and any methods that compute $\varphi(A)$ explicitly, should in practice be avoided. Despite this, Padé approximants are very effective on small dense matrices and consequently are used at the core of our sparse Krylov technique, as outlined in §4.2.2.

According to Lu (2003), when using exponential integrators in practice, the action of $\varphi(A)$ on a vector is required. Two methods that approximate $\varphi(A)v$ without explicitly forming $\varphi(A)$ are those based on best rational approximants (Schmelzer & Trefethen 2007, Lu 2003) and Krylov subspaces (Hochbruck et al. 1996). Each will be investigated in this thesis. A third method that is tied with the Cauchy Integral formula is also available, as described by Schmelzer & Trefethen (2007). It will not be discussed in this work due to parallels with best rational approximants and because numerical experiments conducted by Schmelzer & Trefethen (2007) indicated that the rational approximants were typically twice as fast.

Methods based on best rational approximants have featured in papers by Saad (1992), Schmelzer & Trefethen (2007), Lu (2003) and Minchev (2004). Restricted to small dense matrix evaluations in the literature (Sidje 1998), the resulting systems of equations required in the approximation have primarily been solved using direct methods (Sidje 1998, Schmelzer & Trefethen 2007). We aim to extend these ideas to deal with sparse matrices of large dimension and as a consequence GMRES strategies for the iterative solution of the resulting shifted systems will be investigated. The proposed methods outlined in §4.1.3 take advantage of the sparse structure of the coefficient matrix and attempt to avoid unnecessary storage.

Krylov subspace techniques for matrix functions were introduced in the late 80’s in the field of chemical physics by Nauts & Wyatt (1983). More recently, these methods have been used for the integration of large ODE systems, as featured in Sidje (1998) and Hochbruck et al. (1996). These techniques form the sparse routine in the EXPOKIT package by Sidje (1998), the most extensive software for computing the matrix exponential currently available (Moler & van Loan 2003). Their success is due to the underlying Arnoldi process which requires only matrix-vector products with the large, sparse matrix

3
With the introduction of these techniques, exponential integrators which were originally regarded as rather impractical due to the cost of computing $\varphi$, have emerged as a viable explicit alternative to implicit integrators.

To assess the suitability of exponential integrators for spatially discrete forms of Richards’ equation, numerical experiments comparing these methods to an established implicit integrator will be conducted. Next, the main objectives of this thesis are outlined. This is followed by a description of the contents of each chapter.

### 1.3 Aims and Objectives

The primary goal of this thesis is to determine whether exponential integrators provide a practical alternative to implicit methods for solving Richards’ equation.

To achieve this goal the following objectives were set:

1. Conduct a thorough literature review of the integration of Richards’ equation, the implementation of exponential integrators and methods for evaluating the matrix function $\varphi$.
2. Present a two-dimensional Richards’ equation problem from the literature and apply the finite volume method to produce a discrete analogue of the PDE in the form of a system of ODEs.
3. Propose an adaptive time stepping strategy for both the exponential Euler and exponential Rosenbrock schemes.
4. Implement a code that uses best rational approximations to the $\varphi$ function embedded in an exponential Euler framework and investigate some GMRES-like strategies for solving the shifted linear systems $(A - \zeta I)x = v$, that arise in these approximants.
5. Implement a code that uses Krylov subspace approximations to the $\varphi$ function embedded in both an exponential Euler and exponential Rosenbrock framework and employ a Padé approximant for the small, dense evaluation resulting from the projection of the large matrix onto the small Krylov subspace.
6. Compare the $\varphi$ function methods and exponential integrators in terms of accuracy and efficiency to a standard integrator, the backward Euler method, and conclude on the preference of integrator for solving Richards’ equation.

### 1.4 Outline of Chapters

The remaining chapters of this thesis are constructed as follows:

A brief background of Richards’ Equation is given in Chapter 2 and a problem concerning the infiltration into very dry soils is presented. The finite volume method is used to discretise the initial boundary value problem in space.

In Chapter 3 the exponential Euler and exponential Rosenbrock time-stepping formulas are presented for the integration of the initial value problem produced from the discretisation of Richards’ equation. A simple adaptive time stepping strategy is then proposed.
Chapter 4 is concerned with approximating the $\varphi$ function featured in the two schemes presented in Chapter 3. Best rational approximants are discussed and alternate GMRES strategies for the solution of the resulting shifted linear systems are formulated. An outline of the Krylov subspace technique as featured in the literature is then given. Numerical solutions to the test problems are exhibited in Chapter 5 and all methods compared. In Chapter 6 the conclusions of the research are summarised and recommendations for future research given.
Chapter 2

Modelling flow through porous media using Richards’ equation

A brief background on Richards’ equation (Richards 1931) is provided in this chapter, followed by the statement of an initial boundary value problem, previously featured in the literature by McBride et al. (2006), that governs flow into a two-dimensional region of very dry heterogeneous soil. This problem is then discretised in space using the finite volume method. This discretisation produces a system of nonlinear autonomous differential equations, which is the required form for the time integration methods discussed in Chapter 3.

2.1 Richards’ equation

In an attempt to explain interacting fluid, thermal and chemical reaction processes, mathematical models are becoming increasingly more complex (McBride et al. 2006). The prediction of water movement in porous media, which is commonly based on Richards’ equation (Pan & Wierenga 1995), is often an integral part of such models. This equation captures sharp fronts in fluid pressures and saturations, which are typical physical effects that are present in multiphase flow (Tocci, Miller, Kelley & Kees 1998). Due to the dependence of the hydraulic conductivity and diffusivity on the moisture content, this equation has a highly non-linear nature. The nonlinearity results in a model that is analytically intractable, except for only a few special cases; some such examples are included in Ju & Kung (1997), Arampatzis, Tzimopoulos, Sakellariou-Makrantonaki & Yannopolos (2001) and Kavetski, Binning & Sloan (2002).

The model for flow of water through porous media is formulated by coupling Darcy’s law with a statement of mass conservation (Kavetski et al. 2001), which, when combined with some simplifying assumptions, produces Richards’ equation. Obtaining numerical solutions to Richards’ equation is a non-trivial task, due not only to the physical effects mentioned above, but also the presence of nonlinearities in the governing equation. In an overview of solution methods for Richards’ equation given by Kulabako (1980) it appears that the spatial approximation was usually accomplished using finite element or finite difference methods. More recently the finite volume method is used (McBride et al. 2006).

The mixed form of Richards’ equation takes the form

\[ \frac{\partial \theta}{\partial t} = \nabla \cdot (K(h) \nabla h) + \frac{\partial K(h)}{\partial z}, \]  

(2.1)
where $\theta$ is the moisture content, $h$ is the pressure head ($m^{-1}$), $K(h)$ is the hydraulic conductivity ($ms^{-1}$) and $z$ the vertical dimension ($m$). The moisture content is a dimensionless quantity that measures the fraction of the total volume of the porous medium (soil, water and void space) filled with water. The pressure head is defined as the ratio of the pressure to the specific weight of water (McBride et al. 2006) while hydraulic conductivity expresses the ease with which a fluid is transported through the void space (Bear & Verruijt 1998). The equation is formulated using Darcy’s law, which describes the movement of moisture in saturated porous media. Provided the density of the water and air phase are constant and neglecting the velocity of the solid (Bear & Verruijt 1998), this law states that the flux equals the product of the hydraulic conductivity and the head gradient (Pan & Wierenga 1995) and is given by

$$q = -K(h) \nabla H,$$

(2.2)

where $H = h + z$ is the total hydraulic head. Neglecting sources or sinks, conservation of mass requires that the rate of change of the mass of water in an arbitrary volume of porous medium ($V$) equals the negative mass flow rate out through the surface ($\Omega$). Since the mass of fluid per unit volume of porous medium is given by the product of the density of the fluid ($\rho$) and the moisture content ($\theta$), we have

$$\frac{d}{dt} \int V (\rho \theta) dV = -\int_\Omega (\rho q) \cdot n d\sigma.$$

Using the divergence theorem and arguing for an arbitrary volume $V$ that the resulting integrand must be zero, we obtain the following representation of the mass conservation law:

$$\frac{\partial}{\partial t}(\rho \theta) + \nabla \cdot (\rho q) = 0.$$

Using Darcy’s Law (2.2) and assuming the density is constant yields

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (K(h) \nabla H).$$

(2.3)

Rewriting equation (2.3) in terms of the pressure head ($h$) produces the mixed form of Richards’ equation (2.1). To complete the model, the dependence of the hydraulic conductivity on the pressure head is needed, that is the function $K(h)$ must be specified. Furthermore, to enable the equation to be solved, a function relating the moisture content $\theta$ to the pressure head $h$ is required (Pan & Wierenga 1995). These functions are based on laboratory experiments and numerous relationships have been proposed in the literature. According to McBride et al. (2006), the most common amongst the groundwater modelling community are the Brooks-Corey (Brooks & Corey 1964) and van Genuchten (van Genuchten 1980) relationships. We follow McBride et al. (2006) and use the latter in this work.
The mixed form (2.1) features two dependent variables, the moisture content \( \theta \) and the pressure head \( h \), both functions of space and time. The soil water retention function relates the pressure head to the moisture content of the porous medium, enabling switching between the two variables to be carried out in a simple manner in numerical algorithms. The function describes a highly nonlinear relationship for negative values of the pressure head, while for non-negative values the moisture content takes on a constant value:

\[
\theta = \begin{cases} 
\theta_{\text{res}} + (\theta_{\text{sat}} - \theta_{\text{res}})(1 + (-\alpha h)^n)^{-m} & h < 0 \\
\theta_{\text{sat}} & h \geq 0.
\end{cases}
\] (2.4)

The constants \( \theta_{\text{res}} \) and \( \theta_{\text{sat}} \) are the residual and saturated water contents and respectively provide a minimum and maximum bound for the moisture content with \( \theta \in [\theta_{\text{res}}, \theta_{\text{sat}}] \).

The symbols \( \alpha \) (m\(^{-1}\)) and \( n \) (dimensionless) denote the positive constants relating to the mean and variation of pore sizes and \( m = 1 - 1/n \). These parameters influence the speed and slope of travelling fronts in the solution and influence the difficulty of the problem (Tocci et al. 1998).

The saturated water content \( \theta_{\text{sat}} \) for a given soil can easily be obtained experimentally (van Genuchten 1980). However, while the residual water content \( \theta_{\text{res}} \) can be measured experimentally through determining the water content in very dry soil, often it must be obtained through extrapolation of existing water retention data (van Genuchten 1980).

The remaining parameters \( n \) and \( \alpha \) are then estimated in an attempt to match with an experimental curve.

Expressing \( h \) in terms of \( \theta \) and re-writing we obtain the moisture content (\( \theta \)-based) form

\[
\frac{\partial \theta}{\partial t} = \nabla \cdot (D(\theta) \nabla \theta) + \frac{\partial K(\theta)}{\partial z},
\] (2.5)

where \( D(\theta) \) is the hydraulic diffusivity. This formulation encounters problems in numerical schemes since the inverse transformation of the soil water retention function (2.4) for non-negative values of the pressure head is not unique. Using the function directly we obtain the pressure head (\( h \)-based) form

\[
C(h) \frac{\partial h}{\partial t} = \nabla \cdot (K(h) \nabla h) + \frac{\partial K(h)}{\partial z},
\] (2.6)

where \( C(h) \) is the specific moisture capacity, defined by

\[
C(h) = \frac{\partial \theta}{\partial h} = \begin{cases} 
mn(\theta_{\text{res}} - \theta_{\text{sat}})(-\alpha h)^n (h(1 + (-\alpha h)^n + 1)^{-1} & h < 0 \\
0 & h \geq 0.
\end{cases}
\] (2.7)

Saturation is a dimensionless measure of the volume of void space in the porous medium filled with water. It is easily related to the moisture content, via the relationship

\[
S = \frac{\theta - \theta_{\text{res}}}{\theta_{\text{sat}} - \theta_{\text{res}}},
\]
so that the saturation lies between 0 and 1. Using the soil water retention function (2.4), a relationship relating the saturation to the pressure head is obtained:

\[
S = \begin{cases} 
(1 + (-\alpha h)^n)^{-m} & h < 0 \\
1 & h \geq 0 .
\end{cases}
\] (2.8)

This equation states that for large negative values of the pressure head the saturation is small while for non-negative values the soil is fully saturated. In the van Genuchten model, the dependence of the hydraulic conductivity on the pressure head is given by

\[
K(h) = \begin{cases} 
K_{sat} S^{1/2} \left(1 - S^{1/m}\right)^m & h < 0 \\
K_{sat} & h \geq 0 ,
\end{cases}
\] (2.9)

that is, the conductivity varies rapidly with \( h \) in unsaturated soil but takes on a constant value (\( K_{sat} \)) in saturated regions. It is clear from the expressions (2.7) and (2.9) of the highly nonlinear nature of Richards’ equation and the numerical challenge this equation provides.

### 2.2 Test Problem

All numerical experiments carried out in this thesis are based on the test problem below previously published by McBride et al. (2006), which the authors claim is a good challenge for any numerical algorithm.

![Schematic diagram of the test problem with clay and sand](image)

**Figure 2.1: Schematic diagram of the test problem with clay ■ and sand □.**

**Unsaturated flow into heterogeneous soil:** The problem is flow into a rectangular region of heterogeneous soil of width 5 m and depth 3 m. The region is divided into nine alternating blocks of clay and sand (see Figure 2.1). Zero flux boundary conditions are applied to all boundaries, apart from a 1 m strip across the top block of sand, where a prescribed constant influx of water of 5 cm/day is applied. We take the initial value of \( h = -500 \) m throughout the region, which corresponds to very dry initial conditions. The simulation is run for 12.5 days. The hydraulic properties of sand and clay are provided in Table 2.1.
Numerical solutions are obtained by McBride et al. (2006) to the test problem using a method that switches between the \( h \)-based (2.6) and mixed form (2.1) of Richards’ equation. They employ the finite volume method to discretise in space and implicit backward Euler approximations are used for advancing the solution in time. The \( h \)-based form achieves good mass balance provided the change in \( h \) is not too small over a time step, whereas the mixed form improves mass balance with a sharp wetting front. The authors argue that combining these makes a more efficient procedure for long time simulations of water flow in soils with frequent infiltration and deep drainage processes.

All numerical solutions obtained in this thesis use the two-dimensional \( h \)-based form of Richards’ equation

\[
C(h) \frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left( K(h) \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial z} \left( K(h) \frac{\partial h}{\partial z} + K(h) \right),
\]

where

\[
C^*(h) \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( K^*(h) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial z} \left( K^*(h) \frac{\partial u}{\partial z} + K(h) \right),
\]

where \( C^*(h) \) is the transformed specific water capacity

\[
C^*(h) = \left\{ \begin{array}{ll}
C(h)(1 + \xi h) \quad &h < 0 \\
C(h) \quad &h \geq 0,
\end{array} \right.
\]

where \( \xi < 0 \) (\( m^{-1} \)) maps negative values of the pressure head to the open interval \((1/\xi, 0)\).

The transformation reduces the nonlinearity of the functions relating the moisture content (2.4) and hydraulic conductivity (2.9) to the pressure head (McBride et al. 2006). Setting \( \xi = 0 \) recovers the identity transformation, which allows the effect of no transformation to be investigated. Without the use of the transformation the methods used by McBride et al. (2006) failed, while the best performing value identified by the authors was \( \xi = -4m^{-1} \). Applying the transformation (2.11) to (2.10) the result is an equation in the new dependent value \( u \)

\[
\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( K^*(h) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial z} \left( K^*(h) \frac{\partial u}{\partial z} + K(h) \right),
\]

where

<table>
<thead>
<tr>
<th>Material</th>
<th>( \theta_{res} )</th>
<th>( \theta_{sat} )</th>
<th>( K_{sat} ) (m/s)</th>
<th>( \alpha ) (m/s)</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clay</td>
<td>0.1060</td>
<td>0.4686</td>
<td>1.516 × 10^{-6}</td>
<td>1.04</td>
<td>1.3954</td>
</tr>
<tr>
<td>Sand</td>
<td>0.0286</td>
<td>0.3658</td>
<td>6.262 × 10^{-5}</td>
<td>2.80</td>
<td>2.2390</td>
</tr>
</tbody>
</table>

Table 2.1: Hydraulic properties of sand and clay including residual moisture content \((\theta_{res})\), saturated moisture content \((\theta_{sat})\), saturated hydraulic conductivity \((K_{sat})\) and parameters relating to the mean pore size \((\alpha)\) and the variation of pore sizes \((n)\).
and $K^*(h)$ is the transformed hydraulic conductivity

$$K^*(h) = \begin{cases} K(h)(1 + \xi h)^2 & h < 0 \\ K(h) & h \geq 0 \end{cases}.$$ 

The boundary conditions for the test problem are now summarised. A prescribed influx of water at a rate of 5 cm/day is applied to a section of the top boundary, while on the remaining boundary sections a no flux condition is imposed, that is

$$K^*(h) \frac{\partial u}{\partial z} + K(h) \approx 5.787 \times 10^{-7} \text{ ms}^{-1} \quad \text{for } 2 < x < 3 \text{ at } z = 3 \text{m},$$

$$K^*(h) \frac{\partial u}{\partial x} = 0 \quad \text{at } x = 0 \text{ and } x = 5 \text{m}, \quad (2.13)$$

$$K^*(h) \frac{\partial u}{\partial z} + K(h) = 0 \quad \text{at } z = 0 \text{ and } x = 3 \text{m}.$$ 

Next, a spatial discretisation of the test problem is performed.

### 2.3 Finite Volume Method

Obtaining analytic solutions to the Richards’ equation problem posed in the previous section is not possible. Typically, to numerically solve such problems we discretise, that is, approximate (2.12) by equations that involve a finite number of unknowns (Saad 2003). To perform this spatial discretisation we have used the Finite Volume Method (FVM) as it is specifically geared towards the solution of conservation equations (Saad 2003). The reason for this is that the fluxes are conserved on a discrete level, with the flux entering a given control volume identical to the one leaving the adjacent volume. This makes it very appealing to conservation problems, like flow through porous media, where the flux is of importance.

### 2.3.1 Mesh structure

The FVM may be used on many different types of geometries, using structured or unstructured meshes (Eymard, Gallouet & Herbin 2006). The technique requires a mesh to be defined over the domain; composed of a number of node points and associated control volumes. The main focus of this thesis is the solution of the system of differential equations obtained as a result of the spatial discretisation process. Here we use a rectangular mesh to illustrate the main ideas, due to its inherent simplicity and because this is the approach taken by McBride et al. (2006). It is worth mentioning, however, that the time integration methods proposed in this thesis are by no means restricted to this mesh structure.

The heterogeneous nature of the test problem, shown in Figure 2.1 raises the question of where to position the nodes and control volumes at the sand-clay interfaces. Our preference was to align adjacent control volume faces with the interfaces rather than the nodes themselves. The reason for this is that the latter requires the interfaces to be prescribed as sand or clay in order to compute quantities such as the specific moisture capacity (2.7),
saturation (2.8) and hydraulic conductivity (2.9). Furthermore, the treatment of fluxes is made more difficult when control volumes are spread across an interface due to the change in hydraulic conductivity moving from one soil to the next. In Figure 2.2 an example of the configuration of nodes and control volumes around an interface is given.

![Figure 2.2: Alignment of nodes and control volumes around the sand-clay interfaces. Nodes are situated on the boundaries while control volume faces are aligned with the sand-clay interfaces.](image)

We note that requiring nodes on the boundary of the domain in addition to forcing that the control volume faces align with the sand-clay interfaces means that it is impossible to construct a uniform mesh.

### 2.3.2 Finite Volume equations

The proceeding formulation will detail the use of the FVM to derive the general Finite Volume Equation (FVE) for an internal node for the partial differential equation (2.12). The resulting FVE is then easily modified to accommodate the boundary conditions. The goal is simply to convert the continuous partial differential equation problem into a discrete problem in space leading to a system of ordinary differential equations, which is the required form to employ the exponential integrator methods presented in Chapter 3. The right hand side of (2.12) can be expressed as the divergence of the vector flux $Q \in \mathbb{R}^2$:

$$Q = -K^*(h) \nabla u - K(h)e_z.$$

This allows the equation to be expressed in the significantly more compact form

$$C^*(h) \frac{\partial u}{\partial t} + \nabla \cdot Q = 0. \quad (2.14)$$

Consider the control volume depicted in Figure 2.3, that consists of a central node P located in a region of “volume” $V_P$ and is connected to four neighbouring nodes, which are directly North, South, East and West of P respectively. The control volume has width $\Delta x_P$ in the $x$-direction and height $\Delta z_P$ in the $z$-direction. The distances between each of the surrounding nodes and P along with the distances between nodes and control volume faces are labelled in the figure.

To represent the value of the unknown quantities $h$ and $u$ at a node point, the label given to the node will be subscripted; for example $h_E$ and $u_E$ are used respectively to represent the untransformed and transformed pressure head value at the east node E. Lower case will be used to denote the control volume face; for example $K_e$ denotes the conductivity.
at the east control volume face. The FVM begins by integrating (2.14) over the entire control volume as follows:

$$\int\int_{V_P} C^*(h) \frac{\partial u}{\partial t} \, dV + \int\int_{V_P} \nabla \cdot \mathbf{Q} \, dV = 0. \tag{2.15}$$

Each of the integrals in (2.15) are intractable in their current form and some approximation is required. To simplify the time dependent integral we make use of the midpoint rule of integration in two dimensions. The approximation takes the form

$$\int\int_{V_P} C^*(h) \frac{\partial u}{\partial t} \, dV \approx \Delta x_P \Delta z_P C^*(h_P) \frac{du_P}{dt}. \tag{2.16}$$

This approximation is most accurate when the node P is located at the centre of its corresponding control volume $V_P$. This is not the case for boundary nodes and hence the error associated with the above approximation is greater there. We have summarised the errors induced by the approximation performed in (2.16) in Table 2.2.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>$O(\Delta x_P \Delta z_P)$ if $\Delta z_P &gt; \Delta x_P$, else $O(\Delta x_P^2 \Delta z_P)$</td>
</tr>
<tr>
<td>East &amp; West</td>
<td>$O(\Delta x_P \Delta z_P^3)$ if $\Delta z_P^2 &gt; \Delta x_P$, else $O(\Delta x_P^2 \Delta z_P)$</td>
</tr>
<tr>
<td>North &amp; South</td>
<td>$O(\Delta x_P^3 \Delta z_P)$ if $\Delta x_P^2 &gt; \Delta z_P$, else $O(\Delta x_P \Delta z_P^2)$</td>
</tr>
<tr>
<td>Corner</td>
<td>$O(\Delta x_P^2 \Delta z_P)$ if $\Delta x_P &gt; \Delta z_P$, else $O(\Delta x_P \Delta z_P^3)$</td>
</tr>
</tbody>
</table>

Table 2.2: Errors associated with the midpoint approximation performed in equation (2.16) for internal nodes and nodes situated on the boundaries and corners of the solution domain depicted in Figure 2.1.

To approximate the integral containing the divergence of the vector flux $\mathbf{Q}$ in (2.15) we use the two-dimensional analogue of the divergence theorem, the result known as Green’s theorem. Due to the strictly rectangular control volumes the single line integral can be
separated into four components, one for each of the four distinct faces. If \( D = \{ n, e, s, w \} \) is the set of faces, then

\[
\int\int_{V_p} \nabla \cdot Q \, dV = \oint_{C} Q \cdot n \, d\sigma = \sum_{j \in D} \int_{j} Q \cdot n \, d\sigma.
\] (2.17)

Each line integral cannot be computed analytically and a numerical integration technique must therefore be employed. We have used a one-point midpoint quadrature approximation. If \( m_j \) is defined to be the coordinates of the midpoint of the cell face of interest and \( s_j \) the corresponding cell face length, the approximation is given by

\[
\int_{j} (Q \cdot n) \, d\sigma \approx (Q \cdot n)_{m_j} s_j.
\] (2.18)

Using (2.17) and (2.18) we hence have that

\[
\int\int_{V_p} \nabla \cdot Q \, dV \approx \sum_{j \in D} (Q \cdot n)_{m_j} s_j.
\] (2.19)

The unit normals for the east and west control volume faces are \( n = i \) and \( n = -i \) respectively with the cell face lengths given by \( \Delta z_p \). Similarly, the unit normals for the north and south cell faces are \( n = k \) and \( n = -k \) with the length of the north and south cell faces given by \( \Delta x_p \). As a result, the contribution from each of the cell faces is greatly simplified with \((Q \cdot n)\) comprising only the first component of the vector flux \( Q \) for the faces in the \( x \)-direction and the second component in the \( z \)-direction. This means that we can write (2.19) as

\[
\int\int_{V_p} \nabla \cdot Q \, dV \approx \Delta z_p (Q_e - Q_w) + \Delta x_p (Q_n - Q_s),
\] (2.20)

where the terms are listed below with a central difference approximation used for the derivatives

\[
Q_n = - \left(K_n^* \left[ \frac{\partial u}{\partial z} \right]_n + K_n \right) = - \left(K_n^* \left[ \frac{u_S - u_P}{\delta z_n} \right] + K_n \right),
\]

\[
Q_s = - \left(K_s^* \left[ \frac{\partial u}{\partial z} \right]_s + K_s \right) = - \left(K_s^* \left[ \frac{u_P - u_S}{\delta z_s} \right] + K_s \right),
\]

\[
Q_e = -K_e^* \left[ \frac{\partial u}{\partial x} \right]_e = -K_e^* \left[ \frac{u_E - u_P}{\delta x_e} \right],
\]

\[
Q_w = -K_w^* \left[ \frac{\partial u}{\partial x} \right]_w = -K_w^* \left[ \frac{u_P - u_W}{\delta x_w} \right].
\]

The transformed and untransformed conductivities in each of the above terms must be evaluated at the midpoint of the control volume faces. This value of \( h \), required to compute \( K(h) \), is not known nor is it represented in our unknowns as it is not located at a pre-allocated node point. As a result, we have implemented an arithmetic weighted average of the conductivities at the nodes either side of the control volume face. Evaluating a
linear interpolant at the face gives the weighted average approximation. As an example to approximate the conductivities at the east control volume face the expressions

\[ K_e \approx \delta x^+ \delta x^+ K_P + \delta x^- \delta x^- K_E \]

\[ K^* e \approx \delta x^+ K^* P + \delta x^- K^* E \]

were used where the constants featured in the ratios are defined in Figure 2.3(b). Combining (2.16) and (2.20) the general FVE for an internal node \( P \) can be expressed as:

\[ C^*(h_P) \frac{du_P}{dt} \approx \left[ \frac{1}{\Delta z_P} (Q_s - Q_n) + \frac{1}{\Delta x_P} (Q_w - Q_e) \right]. \tag{2.21} \]

Division by \( C^*(h_P) \) can be carried out provided \( h_P < 0 \) otherwise \( C^*(h_P) = C(h_P) = 0 \). Fortunately the pressure head remains negative in the test problem proposed in §2.2 and we write:

\[ \frac{du_P}{dt} = \frac{1}{C^*(h_P)} \left[ \frac{1}{\Delta z_P} (Q_s - Q_n) + \frac{1}{\Delta x_P} (Q_w - Q_e) \right]. \tag{2.22} \]

Next, we modify this equation to accommodate the boundary conditions (2.13).

### 2.3.3 Boundary conditions

The formulation of the FVE for a boundary node is identical to that for an interior node. Where the difference lies, is in the treatment of the flux through the control volumes adjacent to the boundary. The test problem exhibits a no-flux condition on the boundary apart from a section at the top of the domain where an inward flux is specified. On each of the no-flux boundaries the flux corresponding to the control volume face lying on the boundary is set to zero. Thus for the test problem, on the west, east and south boundaries respectively the finite volume equation is given by

\[ \frac{du_P}{dt} = \frac{1}{C^*(h_P)} \left[ \frac{1}{\Delta z_P} (Q_s - Q_n) - \frac{1}{\Delta x_P} Q_e \right], \]

\[ \frac{du_P}{dt} = \frac{1}{C^*(h_P)} \left[ \frac{1}{\Delta z_P} (Q_s - Q_n) + \frac{1}{\Delta x_P} Q_w \right], \]

\[ \frac{du_P}{dt} = \frac{1}{C^*(h_P)} \left[ -\frac{1}{\Delta z_P} Q_n + \frac{1}{\Delta x_P} (Q_w - Q_e) \right]. \]

On the north boundary where there is no prescribed influx of water, the equation takes the form:

\[ \frac{du_P}{dt} = \frac{1}{C^*(h_P)} \left[ \frac{1}{\Delta z_P} Q_s + \frac{1}{\Delta x_P} (Q_w - Q_e) \right]. \]

For nodes situated on the strip with the prescribed flux, the FVE is of the form (2.22) with the north flux specified as:

\[ Q_n = 5.787 \times 10^{-7} \text{ ms}^{-1}. \]
2.4 Summary of the spatial discretisation

Once all the $N$ control volumes are visited in the mesh the result is a system of differential equations of the form

$$\frac{du}{dt} = G(u),$$

(2.23)

where the vector valued function $G(u) \in \mathbb{R}^N$ and the vector $u \in \mathbb{R}^N$ containing the discrete nodal transformed pressure head values are defined respectively by

$$G = \begin{pmatrix} g_1(u) \\ \vdots \\ g_N(u) \end{pmatrix} \quad \text{and} \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix},$$

where for example the coordinate function for a general internal node $i$ is given by

$$g_i(u) = \frac{1}{C^*(h_i)} \left[ \frac{1}{\Delta z_i} (Q_s - Q_n) + \frac{1}{\Delta x_i} (Q_w - Q_e) \right].$$

This chapter, through the use of the finite volume method, has produced a discrete analogue of Richards’ equation in the form of a system of ordinary differential equations. In the next chapter, exponential time integration techniques are presented to advance the spatially discrete solution governed by the system of differential equations (2.23) in time.
Chapter 3

Time stepping with Exponential Integrators

In Chapter 2, the initial boundary value problem governing the movement of water in a block of sand and clay was transformed into a system of nonlinear differential equations, each governing the change in the transformed pressure head at a discrete point in space. In order to advance the initial transformed pressure head distribution in time a time integration method must be employed. In this chapter, we present two exponential time integrators for this purpose, the exponential Euler method (Pope 1963) and the exponential Rosenbrock method (Hochbruck et al. 1996).

3.1 The exponential Euler method

Recall from Chapter 2 the system of differential equations, which when paired with the initial transformed pressure head distribution $u_0$

$$u_0 = \left(\frac{500}{500\xi-1}, \ldots, \frac{500}{500\xi-1}\right)^T,$$  

produces the initial value problem

$$\frac{du}{dt} = G(u); \quad u(0) = u_0.$$  

(3.2)

The simplest example of an exponential integrator for (3.2) is the exponential Euler method. It provides a one-step explicit formula for advancing the solution in time, which is similar to the forward Euler method, but tries to overcome the stability problems experienced by its close relative for stiff problems (Schmelzer & Trefethen 2007). In some sections of the literature, this scheme is referred to as the local linearisation method (Jimenez et al. 2002), as it proceeds by linearising the nonlinear vector valued function $G(u)$ and solving the resulting linear problem exactly. By Taylor’s theorem the component scalar valued functions $g_i(u), i = 1, \ldots, N$, of $G(u)$ have the linear approximation

$$g_i(u) \approx g_i(u_n) + \nabla g_i(u_n)^T(u - u_n); \quad i = 1, \ldots, N,$$

where the linearisation is performed at time $t = t_n$ with $u_n$ denoting the current known solution vector at the point $t_n$ in time. Using each of these approximations, $G(u)$ may
be approximated by

\[ G(u) \approx \begin{pmatrix} g_1(u_n) \\ g_2(u_n) \\ \vdots \\ g_N(u_n) \end{pmatrix} + \begin{pmatrix} \nabla g_1(u_n)^T \\ \nabla g_2(u_n)^T \\ \vdots \\ \nabla g_N(u_n)^T \end{pmatrix} (u - u_n), \] (3.3)

where the matrix of gradient operators is the Jacobian matrix of \( G \) evaluated at the known solution \( J_G(u_n) \). For simplicity from this point forth the notation \( J_n = J_G(u_n) \) and \( G_n = G(u_n) \) will be utilised. Using (3.3), we can approximate the original system (3.2) by

\[ \frac{du}{dt} \approx G_n + J_n (u - u_n). \] (3.4)

The result is a system of first order linear nonhomogeneous ordinary differential equations, which can be solved exactly to provide an approximate solution to the original nonlinear problem (3.2). For convenience the approximation in (3.4) will be dropped. Multiplying (3.4) by the integrating factor \( e^{-J_n t} \) yields

\[ \frac{d}{dt} \left( e^{-J_n t} u \right) = e^{-J_n t} \left( G_n - J_n u_n \right). \] (3.5)

The solution is advanced in time by integrating (3.5) over a single time step of length \( \tau \) from \( t = t_n \) to \( t = t_{n+1} \), resulting in dependence of \( u_{n+1} \) on the most recent solution \( u_n \) only and hence the classification as a one-step method. Integrating (3.5) and arranging the resulting equation for \( u_{n+1} \) yields the time-stepping formula

\[ u_{n+1} = e^{\tau J_n} u_n + J_n^{-1} (e^{\tau J_n} - I) \left( G_n - J_n u_n \right). \] (3.6)

Following recent convention (Hochbruck et al. 1996, Schmelzer & Trefethen 2007, Berland et al. 2005) we express the result in terms of the \( \varphi \) function. This is achieved by adding and subtracting \( u_n \) to the right hand side of (3.6), producing

\[ u_{n+1} = u_n + \tau \varphi(\tau J_n) G_n, \] (3.7)

the exponential Euler time-stepping formula (Minchev & Wright 2005) where we recall \( \varphi(A) = A^{-1}(e^A - I) \). It should be stressed that (3.7) is the exact solution of the linear problem (3.4) while it produces an approximate solution to the original nonlinear problem (3.2). In fact, the exponential Euler time-stepping formula produces the exact solution for any constant coefficient linear problem of the form

\[ \frac{du}{dt} = Au + b, \]

where \( A \) and \( b \) do not depend on either \( u \) or \( t \). In general, due to the linearisation performed in (3.4), the approximate solution obtained via (3.7) is second order accurate in
time when used with the exact Jacobian (Hochbruck et al. 1996). Provided the linearisation performed in (3.4) is valid over the time step $\tau$, then the scheme provides an accurate approximation to the solution of the original system of differential equations.

Using the initial discrete transformed pressure head distribution $u_0$, equation (3.7) defines a simple recursive procedure to march the solution in time. From the form of the formula (3.7), we observe that the solution at time $t_{n+1}$ is given by the solution at time $t_n$ plus some correction. This correction requires the computation of the $\varphi$ function and thus will contribute the bulk of the computation of the scheme.

The next section discusses a higher-order exponential integrator for the initial value problem (3.2).

### 3.2 The exponential Rosenbrock method

Rosenbrock methods are linearly implicit numerical schemes for (3.2). This means that the solution of a number of linear systems must be found per predicted solution. The scheme is derived from the diagonally implicit $s$-stage Runge-Kutta method, as featured in Hairer, Norsett & Wanner (1993), which requires $s$ nonlinear systems of equations of $N$ unknowns to be solved at each time step. Rosenbrock methods reduce the computational work by linearising these nonlinear equations. Indeed, these methods can be interpreted as applying one Newton iteration, as opposed to the Runge-Kutta method which continues the iterations until convergence.

In general an $s$-stage Rosenbrock method is given by the formula (Hairer et al. 1993)

$$u_{n+1} = u_n + \tau \sum_{i=1}^{s} b_i k_i,$$

where,

$$k_i = G \left( u_n + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \tau J_n \sum_{j=1}^{i-1} \gamma_{ij} k_j; \quad i = 1, \ldots, s. \quad (3.9)$$

The choice of the parameters $b_i$, $\alpha_{ij}$ and $\gamma_{ij}$ where $\alpha_{i,j} = \gamma_{i,j} = 0$ for $i \leq j$ specify the order of the method. Due to the dependence of $k_i$ on itself, we must solve the linear system (3.9) $s$ times for an $s$-stage method in order to obtain the vectors $k_i, i = 1, \ldots, s$, needed for (3.8). These systems of equations can be expressed as

$$(I - \gamma_{ii} \tau J_n) k_i = G \left( u_n + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \tau J_n \sum_{j=1}^{i-1} \gamma_{ij} k_j,$$

and hence have solution

$$k_i = (I - \gamma_{ii} \tau J_n)^{-1} \left[ G \left( u_n + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \tau J_n \sum_{j=1}^{i-1} \gamma_{ij} k_j \right]. \quad (3.10)$$
Hochbruck et al. (1996) extend the general Rosenbrock method to an exponential setting, adapting the scheme to contain the $\varphi$ function. The authors propose to replace the inverse matrix in (3.10) with the matrix function $\varphi(\gamma_{ii}^n J_n)$

$$k_i = \varphi(\gamma_{ii}^n J_n) \left[ G \left( u_n + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \tau J_n \sum_{j=1}^{i-1} \gamma_{ij} k_j \right],$$

which when paired with (3.8) form a general class of exponential integrators for the initial value problem (3.2). These methods can be interpreted as an extension of the exponential Euler method to multiple stages (Hochbruck et al. 1996), with (3.7) an example of a one stage method ($s = 1$)

$$u_{n+1} = u_n + \tau b_1 k_1,$$

where

$$k_1 = \varphi(\gamma_{11}^n J_n) G_n,$$

with $b_1 = 1$ and $\gamma_{11} = 1$. Constraints on the free parameters are formulated by investigating the order conditions of the scheme. Hochbruck et al. (1996) construct a 7-stage method which is fourth order when used with the exact Jacobian. We provide the scheme below which advances the solution from $t = t_n$ to $t = t_{n+1}$ via the sequential computation of the vectors

$$
\begin{align*}
    k_1 &= \varphi(\frac{1}{3} \tau J_n) G_n, \\
    k_2 &= \varphi(\frac{2}{3} \tau J_n) G_n, \\
    k_3 &= \varphi(\tau J_n) G_n, \\
    w_4 &= \frac{7}{300} k_1 + \frac{97}{150} k_2 - \frac{37}{300} k_3, \\
    u_4 &= u_n + \tau w_4, \\
    d_4 &= G(u_4) - G(u_n) - \tau J_n w_4, \\
    k_4 &= \varphi(\frac{1}{5} \tau J_n) d_4, \\
    k_5 &= \varphi(\frac{2}{5} \tau J_n) d_4, \\
    k_6 &= \varphi(\tau J_n) d_4, \\
    w_7 &= \frac{59}{300} k_1 - \frac{7}{75} k_2 + \frac{269}{300} k_3 + \frac{3}{5} (k_4 + k_5 + k_6), \\
    u_7 &= u_n + \tau w_7, \\
    d_7 &= G(u_7) - G(u_n) - \tau J_n w_7, \\
    k_7 &= \varphi(\frac{1}{7} \tau J_n) d_7, \\
    u_{n+1} &= u_n + \tau (k_3 + k_4 - \frac{4}{3} k_5 + k_6 + \frac{1}{5} k_7).
\end{align*}
$$

The scheme defined by (3.11) will be referred to as the exponential Rosenbrock method throughout this thesis. At first we note the increase in computation required for this method compared with the exponential Euler scheme with seven occurrences of the $\varphi$
function compared with a single occurrence in (3.7). However we show in §4.2.1, that when using Krylov subspace methods to evaluate the \( \varphi \) function, each of \( k_1 \) through \( k_7 \) can be obtained by computing the matrix function three times only, with \( k_2 \) and \( k_3 \) computed using \( k_1 \); and \( k_5 \) and \( k_6 \) computed using \( k_4 \). As a result, the majority of the computation in the scheme is experienced at the stages of computing \( k_1, k_4 \) and \( k_7 \).

The accuracy and stability of the approximate solution of (3.2) obtained using the exponential Euler and exponential Rosenbrock time-stepping formulas rely heavily on the choice of the step size \( \tau \) at each stage in the scheme. If \( \tau \) is too large then inaccuracies can occur and the solution can become physically invalid while if \( \tau \) is too small then the scheme will suffer from slow computation times. The following section provides a strategy to guard against these occurrences by adaptively controlling the time step.

### 3.3 Variable stepsize implementation

Here an adaptive time-stepping strategy is presented, which is commonly used in numerical schemes for differential equations (Papakostas, Tsitouras & Papageorgiou 1996). This strategy has also been paired with standard integrators for Richards’ equation by Kavetski et al. (2001) and D’Haese, Putti, Paniconi & Verhoest (2007). The strategy is based on the local error \( \ell_{n+1} \), defined as the amount by which the numerical solution \( u_{n+1} \) at each step differs from the solution \( \bar{u}_{n+1} \) of the initial value problem

\[
\frac{d\bar{u}}{dt} = G(\bar{u}); \quad \bar{u}(t_n) = u_n,
\]

that is, the local error is given as \( \ell_{n+1} = \bar{u}_{n+1} - u_{n+1} \). The adaptive process proposes a change in the time step based on the formula

\[
\tau = \gamma \left( \frac{\text{tol}}{\text{err}} \right)^{1/p} \tau,
\]

where \( \gamma \) is a safeguard factor, \( \text{err} \) is an approximation to \( \| \ell_{n+1} \|_{\infty} \), \( \text{tol} \) is a user specified measure of the maximum allowable local error induced by the time step from \( t_n \) to \( t_{n+1} \) and \( p \) is related to the order of the integration scheme. The value \( \gamma = 0.9 \) is often used in the literature (Kavetski et al. 2001, Papakostas et al. 1996, Hull, Enright, Fellen & Sedgwick 1972). If \( \text{err} < \text{tol} \) the current time step is accepted and an enlargement in the next time step is made according to

\[
\tau_{n+1} = \gamma \left( \frac{\text{tol}}{\text{err}} \right)^{1/p} \tau_n,
\]

while if \( \text{err} > \text{tol} \) the current step size is rejected and reduced until the error tolerance is met via

\[
\tau_n = \gamma \left( \frac{\text{tol}}{\text{err}} \right)^{1/p} \tau_n.
\]
Typically, to prevent one very accurate time step from producing a large increase in the step size, as well as to prevent one very inaccurate time step from producing a large decrease in the step size, the change in $\tau$ is restricted over a single time step (Bradie 2006). We chose the bounds

$$0.1 \leq \gamma \left( \frac{tol}{err} \right)^{1/p} \leq 1.2.$$ 

Since exact solutions to (3.12) are inaccessible the local error must be approximated. We follow Papakostas et al. (1996) and use the difference between $u_{n+1}$, the solution obtained using the underlying method, and another approximate solution $\tilde{u}_{n+1}$ obtained using a different method.

$$\ell_{n+1} \approx u_{n+1} - \tilde{u}_{n+1}.$$ 

Our adaptive time stepping strategy for both the exponential Euler and exponential Rosenbrock schemes is outlined in Algorithm 1.

**Algorithm 1: Adaptive time stepping**

1. **Start**
   - Specify initial solution $u_0$ and set the error tolerance $tol$
   - Initialise time $t = 0$ and time step $\tau_0 = 0$
2. **Local error approximation**
   - Compute approximation to local error $err = \|u_{n+1} - \tilde{u}_{n+1}\|_{\infty}$
3. **Time step selection**
   - if $err > tol$
     - Reduce current time step $\tau_n$ by a factor $\max(0.1, \gamma (tol/err)^{1/p})$ and go back to 2.
   - else
     - Accept current time step $\tau_n$ and propose new time step $\tau_{n+1} = \min(\gamma (tol/err)^{1/p}, 1.2)\tau_n$
   - end if
4. **Solution**
   - Store solution $u_{n+1}$ and update time $t = t + \tau_n$
   - Compute and store $u_{n+2}$ and update time $t = t + \tau_{n+1}$
   - If $t < t_{end}$ set $u_n = u_{n+2}$ and go to 2.

Preferably the additional computation introduced by computing $\tilde{u}_{n+1}$ is minimal and the cost of implementing adaptive time stepping is small. As a result, typically the second approximation uses computation already performed in the underlying scheme; a famous example is the Runge-Kutta-Fehlberg (RKF45) scheme (Bradie 2006). Fortunately for the exponential Rosenbrock method the formula

$$\tilde{u}_{n+1} = u_n + \tau(-k_1 + 2k_2 - k_4 + k_7),$$  

(3.14)

where $k_1$, $k_2$, $k_4$ and $k_7$ are as given in (3.11) is a second order method for (3.2) (Hochbruck et al. 1996). We have approximated the infinity norm of the local error for the exponential Rosenbrock method by

$$err = \|u_{n+1} - \tilde{u}_{n+1}\|_{\infty}$$ 

$$= \tau\|k_3 + 2k_4 - \frac{2}{3}k_5 + k_6 - \frac{5}{6}k_7\|_{\infty},$$

22
and used $p = 3$ which was found to be a reasonable value in the numerical experiments conducted by Hochbruck et al. (1996). For the exponential Euler method we proposed to obtain the second approximate solution $\tilde{u}_{n+1}$ by performing two time steps of length $\frac{\tau}{2}$ and compute

$$u_{n+1/2} = u_n + \frac{\tau}{2} \varphi \left( \frac{\tau}{2} J_n \right) G_n,$$
$$\tilde{u}_{n+1} = u_{n+1/2} + \frac{\tau}{2} \varphi \left( \frac{\tau}{2} J_{n+1/2} \right) G(u_{n+1/2}).$$

Hence, the error for the exponential Euler method is estimated by

$$err = \| u_{n+1} - \tilde{u}_{n+1} \|_{\infty}$$
$$= \tau \| \varphi(\tau J_n)G_n - \frac{1}{2} \varphi(\frac{\tau}{2} J_n)G_n - \frac{1}{2} \varphi(\frac{\tau}{2} J_{n+1/2})G(u_{n+1/2}) \|_{\infty}$$

where we found the value $p = 2$ produced reasonable results for the test problem. Unfortunately this approximation introduces two additional computations of the matrix function on a vector as well as an additional evaluation of $G$. The increased computation is not ideal, compared with the exponential Rosenbrock scheme where minimal additional computation is introduced.

This chapter has provided two numerical schemes for approximating the solution of the initial value problem (3.2). The following chapter is devoted to methods for approximating the $\varphi$ function required in both the exponential Euler and Rosenbrock schemes.
Approximation of the $\varphi$ function for Exponential Integrators

In Chapter 2 the finite volume method when applied to Richards’ equation produced a system of nonlinear first order differential equations. In Chapter 3, two formulae from the class of exponential integrators were presented for advancing the discrete spatial solution of Richards’ equation in time. The question that has yet to be answered concerns the evaluation of the matrix function $\varphi$ that appears in both schemes. The efficient and accurate evaluation of this function is vital to the overall performance of the exponential integrator (Minchev & Wright 2005).

![Figure 4.1: Sparsity structure of the Jacobian matrix for the test problem outlined in §2.2.](image)

Strategies for approximating $\varphi(\tau J)v$ for the large, sparse Jacobian matrices $J$ produced from the spatial discretisation of Richards’ equation, as shown in Figure 4.1, are described in this chapter.

### 4.1 Best rational approximants

Recall from Chapter 3 the exponential Euler time stepping formula

$$u_{n+1} = u_n + \tau \varphi(\tau J_n)G_n,$$

where $J_n$ is the Jacobian matrix at the current time level $t_n$, $G_n = G(u_n)$ and $\tau$ is the time step. Here, a strategy is presented for the fast evaluation of $\varphi(\tau J)v$; hence allowing the term $\varphi(\tau J_n)G_n$ to be computed in (4.1). The technique is concerned with a rational
approximation to the matrix function, constructed based on the scalar analogue of \( \varphi \), namely

\[
\varphi(z) = \frac{e^z - 1}{z},
\]

(4.2)
on the nonpositive real line \((-\infty, 0]\). As a consequence, the approximation remains valid for the matrix analogue of a nonpositive scalar when extended to \( \varphi(\tau J) \). The requirement here is that \( J \) be negative semidefinite (Schmelzer & Trefethen 2007). In practice, however, symmetry and a nonpositive spectrum of the Jacobian matrix for Richards’ equation problems is not guaranteed. In this case, we propose to still use the rational approximation provided the eigenvalues do not drift “too far” from the nonpositive real line. Further details are provided in §4.1.2.

In principle, the technique relies on extending the Chebyshev polynomial approximation theory to rational functions (Sidje 1998) thus introducing the notion of a best rational approximant. The best rational approximant of \( \varphi(z) \) on \((-\infty, 0]\) minimises the maximum absolute difference between the rational function and (4.2) over the entire interval. The problem of finding the approximant is summarised as follows (Schmelzer & Trefethen 2007, Sidje 1998).

Let \( P_n \) denote the set of all polynomials of degree at most \( n \in \mathbb{N} \) with real coefficients. Let \( R_{mn} \) denote the set of all rational functions with numerator degree at most \( m \in \mathbb{N} \) and denominator degree at most \( n \in \mathbb{N} \),

\[
R_{mn} = \left\{ \frac{q_1(z)}{q_2(z)} \mid q_1(z) \in P_m, q_2(z) \in P_n, q_2(z) \neq 0 \right\}.
\]

The best rational approximant \( r_{mn}(z) \) to \( \varphi(z) \) on the interval \((-\infty, 0]\) is defined by

\[
\min_{r_{mn} \in R_{mn}} \max_{z \in (-\infty, 0]} |r_{mn}(z) - \varphi(z)|.
\]

The function \( r_{mn}(z) \) is unique and exists (Meinardus 1967).

In this thesis we restrict the discussion to diagonal approximants, where the polynomials \( q_1(z) \) and \( q_2(z) \) have the same maximum degree \( (m = n = p) \). These feature more prominently in the literature, with each of Saad (1992), Lu (2003), Minchev (2004) and Sidje (1998) favouring this form. The strength in using rational approximations for exponential integrators, as is demonstrated later, lies in their partial fraction representation (Schmelzer & Trefethen 2007). For the \((p, p)\) diagonal approximant, this representation has the form

\[
r_p(z) = \alpha^{(0)} + \sum_{j=1}^{p} \frac{\alpha^{(j)}}{z - \zeta^{(j)}},
\]

(4.3)

where \( \alpha^{(0)} = p_0/q_0 \) is the ratio of the constant terms of the numerator and denominator polynomials and \( \zeta^{(j)}, \alpha^{(j)} \in \mathbb{C}, j = 1, \ldots, p \), are respectively the poles and residues of the
rational function, satisfying (Gallopoulos & Saad 1992)

\[ q(\zeta^{(j)}) = 0, \quad \alpha^{(j)} = \frac{p(\zeta^{(j)})}{q'(\zeta^{(j)})}; \quad j = 1, \ldots, p. \]  \hspace{1cm} (4.4)

Details of techniques used to compute these best rational approximants are presented in the next section, followed by a discussion of the applicability of these methods in the context of Richards’ equation.

4.1.1 Constructing approximants

In general determining the best diagonal rational approximant for a given \( p \) is non-trivial (Lu 2003). However, a simple strategy to obtain a useful approximant is to use the corresponding best rational approximation of \( e^z \) on \( (-\infty, 0] \). This problem is a famous one and was stated and solved by Cody, Meinardus & Varga (1969). In a subsequent paper (Carpenter, Ruttan & Varga 1984) the coefficients of the numerator and denominator polynomials were computed and listed up to and including \( p = 30 \). The corresponding poles and residues could then be computed using (4.4) at the expense of nonlinear iterations to determine the poles. Using these values it is then possible to extract the poles and residues for the approximation to \( \varphi(z) \). The idea has been described by Lu (2003) and considers the diagonalisable matrix

\[ B = \begin{pmatrix} z & 1 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{2 \times 2}. \]  \hspace{1cm} (4.5)

The exponential of \( B \) has the exact representation

\[ e^B = \begin{pmatrix} 1 & -1 \\ 0 & z \end{pmatrix} \begin{pmatrix} e^z & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & z \end{pmatrix}^{-1} = \begin{pmatrix} e^z \varphi(z) \\ 0 & 1 \end{pmatrix}, \]

for \( z \neq 0 \), with the \((1, 2)\) entry producing \( \varphi(z) \). By using the best rational approximation to \( e^B \)

\[ e^B \approx \alpha^{(0)} I + \sum_{j=1}^{p} \alpha^{(j)} (B - \zeta^{(j)} I)^{-1}, \]  \hspace{1cm} (4.6)

and by noting that the \((1, 2)\) entry of the matrix \( (B - \zeta^{(j)} I)^{-1} \) is \( \zeta^{(j)}(z - \zeta^{(j)})^{-1} \) we obtain the approximation

\[ \varphi(z) \approx \alpha^{(0)} + \sum_{j=1}^{p} \frac{\alpha^{(j)}}{\zeta^{(j)}(z - \zeta^{(j)})}. \]

Although approximations to \( \varphi(z) \) obtained in this way are not optimal, they provide reasonable quality (Schmelzer & Trefethen 2007) and are very useful for constructing arbitrary degree approximations. Alternatively, the standard procedure for constructing best rational approximations is the Remes Algorithm (Braess 1986). However, the
most efficient way to compute near-best approximations seems to be the Carathéodory-
Fejér procedure as applied to the $\varphi$ function by Schmelzer & Trefethen (2007). We provide
no further detail, and instead use the poles and residues given by Lu (2003).

A further point is that the matrix (4.5) is also important in constructing approximations when the largest eigenvalue of $\tau J$ is positive. Suppose $\lambda_1 > 0$ denotes the largest eigenvalue of $\tau J$ and hence the largest eigenvalue of

$$B = \begin{pmatrix} \tau J & \mathbf{I} \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{2N \times 2N},$$

(4.7)

where

$$e^B = \begin{pmatrix} e^{\tau J} & \varphi(\tau J) \\ 0 & \mathbf{I} \end{pmatrix}. \quad (4.8)$$

Lu (2003) suggests splitting the exponential $e^B = e^{\lambda_1 \mathbf{I}} e^{B - \lambda_1 \mathbf{I}}$. With the eigenvalues of the matrix $B - \lambda_1 \mathbf{I}$ guaranteed to be non-positive (Minchev 2004) the rational approximant (4.6) is invoked. Extracting the (1,2) entry of the approximation, the following approximation was obtained by Lu (2003):

$$\varphi(\tau J) \approx e^{\lambda_1} \sum_{j=1}^{p} \frac{a_j^{(j)}}{\lambda_1 + \zeta_j} (\tau J - (\lambda_1 + \zeta_j) \mathbf{I})^{-1}. \quad (4.9)$$

Note however, that the use of the approximation (4.9) requires the largest eigenvalue of $\tau J$, which comes with unwanted overhead.

4.1.2 Applicability to Richards’ equation

For the test problem stated in §2.2, our numerical experiments show the resulting Jacobian matrix is not semidefinite. This presents a problem with the rational function no longer guaranteed to be an accurate approximation to $\varphi$. Nevertheless, Figures 4.2(a) and 4.2(b) indicate that due to the small magnitude of the imaginary components of the eigenvalues the approximation should remain accurate. A similar diagram to Figure 4.2(b) for the exponential has previously been published by Sidje (1998). The eigenvalue distribution of $\tau J$, obtained from a typical simulation of the test problem, indicates that the error induced by the rational approximant is at worst $10^{-10}$ for scalars in the range of the eigenvalues of $\tau J$.

Indeed, Gallopoulos & Saad (1992) who were interested in approximating the matrix exponential, experimentally verified in their problems that the approximation remained remarkably accurate provided the eigenvalues were close to the domain of applicability. Since the largest positive eigenvalue of $\tau J$ is small we do not take the approach outlined above with the shift on the exponential but instead use a direct rational approximation to $\varphi(z)$.

Simplification can be made to (4.3) by noting that since the coefficients of the denominator polynomial $q_2(z)$ are real, the poles and residues occur in conjugate pairs. Suppose we
take an arbitrary pole-residue pair \((\alpha, \zeta)\) and its corresponding conjugate partner \((\overline{\alpha}, \overline{\zeta})\) and consider the segment of the sum \((4.3)\) containing their addition. For \(z\) real, due to the analyticity of the reciprocal function, the conjugate and reciprocal operations can be reversed to give

\[
\frac{\alpha}{z - \zeta} + \frac{\overline{\alpha}}{z - \overline{\zeta}} = \alpha \frac{1}{z - \zeta} + \overline{\alpha} \left( \frac{1}{z - \zeta} \right) = 2 \Re \left\{ \frac{\alpha}{z - \zeta} \right\}, \tag{4.10}
\]

where we have use the fact that \(z_1 \overline{z}_2 + \overline{z}_1 z_2 = 2 \Re \{z_1 \overline{z}_2\} \) for \(z_1, z_2 \in \mathbb{C}\). This allows \((4.3)\) to be expressed as

\[
r_p(z) = \alpha^{(0)} + 2 \Re \left\{ \sum_{j=1}^{p/2} \frac{\alpha^{(j)}}{z - \zeta^{(j)}} \right\}, \tag{4.11}
\]

where we now need only half of the original \(p\) fractions. The real strength in using the partial fraction form, as previously mentioned, lies in the ability to capture the action of the matrix function on a vector. This is easily seen by considering the matrix analogue of \((4.11)\). With the introduction of a vector \(\mathbf{v}\) we can use the approximation

\[
\varphi(\tau J)\mathbf{v} \approx \alpha^{(0)} \mathbf{v} + 2 \Re \left\{ \sum_{j=1}^{p/2} \alpha^{(j)} (\tau J - \zeta^{(j)} \mathbf{I})^{-1} \mathbf{v} \right\}, \tag{4.12}
\]

with the expense of solving \(p/2\) square \(N \times N\) sparse, complex systems of linear equations of the form

\[
(\tau J - \zeta^{(j)} \mathbf{I}) \mathbf{x}^{(j)} = \mathbf{v} \quad j = 1, \ldots, p/2. \tag{4.13}
\]
Taking the real operator inside the summation, we can write (4.12) as

$$\varphi(\tau J)v \approx \alpha^{(0)}v + 2 \sum_{j=1}^{p/2} \left( \alpha^{(j)}_R x^{(j)}_R - \alpha^{(j)}_I x^{(j)}_I \right),$$

(4.14)

where the subscripts $\Re$ and $\Im$ are used respectively to denote real and imaginary components. A downside of the reduction in the number of fractions is that the analysis only holds for $z$ real and hence (4.12) is only true when the eigenvalues of $\tau J$ are real. Our preliminary investigations suggested that benefits gained from using the full expansion over (4.12) were small in comparison with the computational advantages of requiring half the number of solutions to equations of the form (4.13). We summarise the approximation to $\varphi(\tau J)v$ via best rational approximants below:

1. Specify $\alpha^{(0)}$ and the poles $\zeta^{(j)}$, $j = 1, \ldots, p/2$ of $r_p(z)$ unique under the conjugate operator.
2. For each pole $\zeta^{(j)}$ solve the equation $(\tau J - \zeta^{(j)}I)x^{(j)} = v$ for $x^{(j)}$.
3. Form the approximation $\varphi(\tau J)v = \alpha^{(0)}v + 2 \Re \left\{ \sum_{j=1}^{p/2} \alpha^{(j)}x^{(j)} \right\}$.

For all of the simulations reported in this work, the poles and residues tabulated by Lu (2003) for the $p = 14$ diagonal rational approximation to $\varphi(z)$ on $(-\infty, 0]$ were used for the approximant (4.12). The coefficients were computed to high accuracy and are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Real component</th>
<th>Imaginary component</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha^{(0)}$</td>
<td>0.68944296265527394984 (-15)</td>
</tr>
<tr>
<td>$\alpha^{(1)}$</td>
<td>-0.16598679663720768703 (+02)</td>
</tr>
<tr>
<td>$\alpha^{(2)}$</td>
<td>0.22963504666229092280 (+02)</td>
</tr>
<tr>
<td>$\alpha^{(3)}$</td>
<td>-0.753501496092468798 (+03)</td>
</tr>
<tr>
<td>$\alpha^{(4)}$</td>
<td>0.6544026011697416874 (+02)</td>
</tr>
<tr>
<td>$\alpha^{(5)}$</td>
<td>0.17992885537758290973 (+01)</td>
</tr>
<tr>
<td>$\alpha^{(6)}$</td>
<td>-0.2224782352681356103 (+02)</td>
</tr>
<tr>
<td>$\alpha^{(7)}$</td>
<td>0.1690103692838164789 (-04)</td>
</tr>
<tr>
<td>$\zeta^{(1)}$</td>
<td>0.65586170606958520061 (+01)</td>
</tr>
<tr>
<td>$\zeta^{(2)}$</td>
<td>0.6032968674314355458 (+01)</td>
</tr>
<tr>
<td>$\zeta^{(3)}$</td>
<td>0.4952707295428340179 (+01)</td>
</tr>
<tr>
<td>$\zeta^{(4)}$</td>
<td>0.3251520707621848967 (+01)</td>
</tr>
<tr>
<td>$\zeta^{(5)}$</td>
<td>0.8013602893611439276 (+00)</td>
</tr>
<tr>
<td>$\zeta^{(6)}$</td>
<td>-0.26587124072174283827 (+01)</td>
</tr>
<tr>
<td>$\zeta^{(7)}$</td>
<td>-0.7809594003956373966 (+01)</td>
</tr>
</tbody>
</table>

Table 4.1: Values of the residues $\alpha^{(j)}, j = 0, \ldots, 7$ and poles $\zeta^{(j)}, j = 1, \ldots, 7$ for the partial fraction representation of the best (14,14) rational approximant to $\varphi(z)$ on the nonpositive real line. The order of magnitude of each value is provided in parentheses.

In summary, best rational approximants have provided an avenue for approximating $\varphi(\tau J_n)G_n$, which is required in the exponential Euler scheme. The next sections focus on Step 2 of the approximation, where strategies are presented to handle equations of
the form (4.13), where for the test problem stated in §2.2 the Jacobian matrix \( J \) is large and sparse.

4.1.3 GMRES-like methods for the shifted linear systems

The Generalised Minimum Residual method (GMRES) was developed by Saad & Schultz (1986) for solving large, sparse non-symmetric linear systems of equations. The method when combined with a suitable preconditioner is the one of the most effective iterative solvers currently available in terms of efficiency and robustness (Saad 1994). The method is easily extended to the complex non-Hermitian case and we begin by applying it directly to the equation:

\[
(\tau J - \zeta I)x = v. \quad (4.15)
\]

Suppose \( x^{(0)} \) is some initial estimate to the solution of (4.15) with \( r^{(0)} = v - (\tau J - \zeta I)x^{(0)} \), the corresponding residual of this estimate. GMRES builds an approximate solution to (4.15) of the form

\[
x^{(k)} = x^{(0)} + V_k y_k, \quad (4.16)
\]

where the \( k \) columns of the matrix \( V_k \in \mathbb{C}^{N \times k} \) form an orthonormal basis for the Krylov subspace of dimension \( k \)

\[
K_k(\tau J - \zeta I, r^{(0)}) = \text{span}\left\{ r^{(0)}, (\tau J - \zeta I)r^{(0)}, \ldots, (\tau J - \zeta I)^{k-1}r^{(0)} \right\}. \quad (4.17)
\]

This basis is constructed using the well-known Arnoldi process (Arnoldi 1951), which is analogous to the modified Gram-Schmidt procedure for constructing an orthonormal basis for (4.17). Since the coefficient matrix \( \tau J - \zeta I \) is complex, we use the complex inner product for constructing the basis, defined as

\[
\langle x, y \rangle = \sum_{j=1}^{N} x_j y_j = x^H y, \quad (4.18)
\]

where \( x, y \in \mathbb{C}^N \). After \( k \) steps of the orthogonalisation process, an upper Hessenberg matrix \( H_k = V_k^H(\tau J - \zeta I)V_k \in \mathbb{C}^{k \times k} \) is produced which is the orthogonal projection of \( \tau J - \zeta I \) onto the subspace \( K_k \) in (4.17). This matrix satisfies the important so-called Arnoldi relation

\[
(\tau J - \zeta I)V_k = V_{k+1} H_k, \quad (4.19)
\]

where the columns of the matrix \( V_{k+1} = (V_k, v_{k+1}) \in \mathbb{C}^{N \times (k+1)} \) form an orthonormal basis for the Krylov subspace of dimension \( k + 1 \) and the matrix \( H_k \in \mathbb{C}^{(k+1) \times k} \) is defined
by

\[ H_k = \begin{pmatrix} H_k \\ h_{k+1,k} e_k^T \end{pmatrix}, \] (4.20)

where \( e_k \) is the \( k^{th} \) canonical basis vector in \( \mathbb{R}^k \). Using the assumed form of the approximate solution (4.16) along with the Arnoldi relation (4.19), it was shown by Saad & Schultz (1986) for a real coefficient matrix that the residual of the approximate solution \( x^{(k)} \) can be obtained via the expression

\[ r^{(k)} = V_{k+1} (\beta e_1 - H_k y_k), \] (4.21)

where \( \beta = ||r^{(0)}||_2 \). The same result holds for a complex coefficient matrix (Fraysse, Giraud, Gratton & Langou 2005). Since the columns of \( V_{k+1} \) are orthonormal with respect to the inner product (4.18), then

\[ ||r^{(k)}||_2 = ||\beta e_1 - H_k y_k||_2, \] (4.22)

and the problem of extracting the \( x^{(k)} \) that minimises the residual norm over all such vectors in the affine space \( x^{(0)} + K_k (\tau J - \zeta I, r^{(0)}) \) is reduced to the least squares problem of finding \( y_k \) such that

\[ y_k = \arg \min_{y \in \mathbb{C}^k} ||\beta e_1 - H_k y||_2. \] (4.23)

The resulting rectangular system

\[ H_k y_k = \beta e_1, \] (4.24)

is much more favourable than (4.15) as computing \( y_k \) is inexpensive since \( k \) is typically small (Saad 2003). Because alternate variants of the GMRES method are proposed in subsequent sections, we refer to the method outlined above and in Algorithm 2 as *Standard GMRES*. The algorithm breaks down if \( h_{k+1,k} = 0 \) since the vector \( v_{k+1} \) cannot be formed. However, when this occurs it is simple to see from (4.20) that

\[ V_{k+1} H_k = V_k H_k, \] (4.25)

and hence the Arnoldi relation (4.19) simplifies to

\[ (\tau J - \zeta I) V_k = V_k H_k. \] (4.26)

The residual of the approximate solution \( x^{(k)} \) can then be expressed as

\[ r^{(k)} = V_k (\beta e_1 - H_k y_k), \]

from which it is clear that \( ||r^{(k)}||_2 = 0 \) when \( y_k = \beta H_k^{-1} e_1 \). This means the solution obtained via \( x^{(k)} = x^{(0)} + V_k y_k \) provides the exact solution to the problem (4.15).
Algorithm 2: Standard GMRES for \((\tau J - \zeta I)x = v\)

1. **Start**
   - Choose \(x^{(0)}\) and compute \(r^{(0)} = v - (\tau J - \zeta I)x^{(0)}\) and \(\beta = \|r^{(0)}\|_2\)
   - Set the maximum allowable residual two norm tolerance \(TOL\)

2. **Arnoldi process**
   - \(v_1 = r^{(0)}/\beta\)
   - for \(k = 1\) until convergence do
     - Compute \(w = (\tau J - \zeta I)v_k\)
     - for \(i = 1, \ldots, k\) do
       - \(h_{i,k} = w^H v_i\)
       - \(w = w - h_{i,k}v_i\)
     - end for
     - Compute \(h_{k+1,k} = \|w\|_2\) and \(v_{k+1} = w/h_{k+1,k}\)
     - Obtain \(\|r^{(k)}\|_2 = \|\beta e_1 - \Pi_k y_k\|_2\) where \(y_k = \arg \min_{y \in \mathbb{C}^k} \|\beta e_1 - \Pi_k y\|_2\)
     - If \(\|r^{(k)}\|_2 < TOL\) convergence attained. Go to 3.
     - Update \(V_{k+1} = [V_k, v_{k+1}]\)
   - end for

3. **Form the approximate solution**
   - Compute \(x^{(k)} = x^{(0)} + V_k y_k\) where \(y_k = \arg \min_{y \in \mathbb{C}^k} \|\beta e_1 - \Pi_k y\|_2\)

the next section, a modification to the GMRES algorithm is provided in an attempt to reduce the computational cost associated with the complex coefficient matrix \(\tau J - \zeta I\).

**Complex arithmetic free Arnoldi**

A complex number requires twice the storage of a non-complex number. Moreover, complex arithmetic is more expensive than real arithmetic. Therefore, any action that can be taken to avoid the storage of, or arithmetic with, complex numbers will prove to be beneficial.

Applying GMRES directly to the shifted linear system (4.15) does not take advantage of the structure of the coefficient matrix. However, if the Krylov subspace was formed with \(J\) rather than \(\tau J - \zeta I\) and if the initial estimate of the solution was chosen as \(x^{(0)} = 0\) so that

\[
r^{(0)} = v - (\tau J - \zeta I)x^{(0)} = v \in \mathbb{R}^N,
\]

then it is clear that we can avoid the need to work with and store complex numbers in the Arnoldi process. We now seek an approximate solution of (4.15), from the affine space \(x^{(0)} + \mathcal{K}_k(J, r^{(0)})\). In this case, the Arnoldi relation is

\[
J V_k = V_{k+1} \bar{H}_k, \tag{4.27}
\]

where the matrices \(V_k\) and \(\bar{H}_k\) are real provided \(x^{(0)} = 0\). Building an approximate solution of the form \(x^{(k)} = x^{(0)} + V_k y_k^s\) the residual can be expressed as

\[
r^{(k)} = v - (\tau J - \zeta I)x^{(k)}
= v - (\tau J - \zeta I)(x^{(0)} + V_k y_k^s)
= r^{(0)} - (\tau J V_k - \zeta V_k) y_k^s. \tag{4.28}
\]
Using (4.27)

\[
\mathbf{r}^{(k)} = \mathbf{r}^{(0)} - (\tau \mathbf{v}_{k+1} \mathbf{h}_k - \zeta \mathbf{v}_k) \mathbf{y}_k^s
\]

\[
= \mathbf{r}^{(0)} - \mathbf{v}_{k+1} \left[ \tau \mathbf{h}_k - \zeta \begin{pmatrix} \mathbf{I}_k \\ \mathbf{0}_k^T \end{pmatrix} \right] \mathbf{y}_k^s.
\]

Let \( \mathbf{H}_k^s \in \mathbb{C}^{(k+1)\times k} \) denote the matrix

\[
\mathbf{H}_k^s = \tau \mathbf{h}_k - \zeta \begin{pmatrix} \mathbf{I}_k \\ \mathbf{0}_k^T \end{pmatrix}.
\] (4.29)

With this notation and since the first column of \( \mathbf{V}_{k+1} \) is \( \mathbf{r}^{(0)} \) scaled by \( \beta \) we write

\[
\mathbf{r}^{(k)} = \mathbf{V}_{k+1} (\beta \mathbf{e}_1 - \mathbf{H}_k^s \mathbf{y}_k^s),
\] (4.30)

and the residual two norm is given by

\[
\| \mathbf{r}^{(k)} \|_2 = \| \beta \mathbf{e}_1 - \mathbf{H}_k^s \mathbf{y}_k^s \|_2.
\] (4.31)

Requiring (4.31) to be minimal over \( K_{k}^{(J, \mathbf{r}^{(0)})} \) we again arrive at a least squares problem of finding \( \mathbf{y}_k^s \) such that

\[
\mathbf{y}_k^s = \arg \min_{\mathbf{y} \in \mathbb{C}^k} \| \beta \mathbf{e}_1 - \mathbf{H}_k^s \mathbf{y} \|_2,
\] (4.32)

which is of identical dimension to the one featured in Standard GMRES (4.23). Throughout the remainder of this thesis we will refer to the method defined by the above analysis as Shifted GMRES. We note that this method has appeared previously in the work by Frommer & Glassner (1998), although applied in the context of a positive definite matrix and a real shift on the diagonal. The structure of the algorithm is identical to Standard GMRES and is outlined in Algorithm 3.

As with Algorithm 2, break down of the process is experienced when \( h_{k+1,k} = 0 \). In this case we again obtain (4.25) with the resulting Arnoldi relation modified to

\[
\mathbf{J} \mathbf{v}_k = \mathbf{v}_k \mathbf{h}_k.
\]

Repeating the analysis performed on \( \mathbf{r}^{(k)} \) in (4.28) yields the following expression for the residual vector

\[
\mathbf{r}^{(k)} = \mathbf{V}_k (\beta \mathbf{e}_1 - (\tau \mathbf{h}_k - \zeta \mathbf{I}_k) \mathbf{y}_k^s).
\]

Clearly, if \( \mathbf{y}_k^s = \beta (\tau \mathbf{h}_k - \zeta \mathbf{I}_k)^{-1} \mathbf{e}_1 \) then \( \mathbf{r}^{(k)} = \mathbf{0} \) and hence \( \mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathbf{V}_k \mathbf{y}_k^s \) is the exact solution to the shifted linear system (4.15). In the following section, in an attempt to obtain further computational savings we present a strategy for avoiding all complex arithmetic in the solution process.
Algorithm 3: Shifted GMRES for \((\tau J - \zeta I)x = v\)

1. Start
   Choose \(x^{(0)}\) and compute \(r^{(0)} = v - (\tau J - \zeta I)x^{(0)}\) and \(\beta = \|r^{(0)}\|_2\)
   Set the maximum allowable residual two norm tolerance \(TOL\)

2. Arnoldi process
   \(v_1 = r^{(0)}/\beta\)
   for \(k = 1\) until convergence do
   Compute \(w = Jv_k\)
   for \(i = 1, \ldots, k\) do
   \(h_{i,k} = w^Hv_i\)
   \(w = w - h_{i,k}v_i\)
   end for
   Compute \(h_{k+1,k} = \|w\|_2\) and \(v_{k+1} = w/h_{k+1,k}\)
   Obtain \(\|r^{(k)}\|_2 = \|\beta e_1 - \Pi_k y_k^*\|_2\) where \(y_k^* = \arg\min_{y \in C^k} \|\beta e_1 - \Pi_k y\|_2\)
   If \(\|r\|_2 < TOL\) convergence attained. Go to 3.
   Update \(V_{k+1} = [V_k, v_{k+1}]\)
   end for

3. Form the approximate solution
   Compute \(x^{(k)} = x^{(0)} + V_k y_k^*\) where \(y_k^* = \arg\min_{y \in C^k} \|\beta e_1 - \Pi_k y\|_2\)

\[\text{Complex arithmetic free GMRES}\]

By separating the shifted linear system (4.15) into its real and imaginary components, the following discussion will show that this leads to a quadratic equation in \(J\) that must be solved for the imaginary component of the solution while the real component is obtained explicitly from this result. By equating real and imaginary components of (4.15), we arrive at a pair of simultaneous systems of equations in the unknown real and imaginary components of the solution \(x\):

\[\begin{align*}
(\tau J - \zeta R I)x_R + \zeta I x_\Im &= v \\
(\tau J - \zeta R I)x_\Im - \zeta R x_R &= 0.
\end{align*}\]  

(4.33)  
(4.34)

Solving these equations for \(x_R\) and \(x_\Im\) we see that \(x_\Im\) is given by the solution of the quadratic equation

\[\left(\tau^2 J^2 - 2\zeta R \tau J + |\zeta|^2 I\right) x_\Im = \zeta I v,\]

(4.35)

where \(|\zeta|^2 = \zeta_R^2 + \zeta_I^2\) is the square of the complex modulus. The real component of \(x\) is then obtained via the product

\[x_R = \frac{1}{\zeta_R} (\tau J - \zeta R I)x_\Im.\]

(4.36)

Using the expression for \(x_R\) we can express the approximation to \(\varphi(\tau J)v\) entirely in terms of the imaginary part of the solution of the shifted systems. Using (4.14) obtains the approximation

\[\varphi(\tau J)v \approx \alpha^{(0)} v + 2 \sum_{j=1}^{p/2} \left( \frac{\alpha_R^{(j)}}{\alpha_\Im^{(j)}} (\tau J - \zeta_R^{(j)} I) - \alpha_\Im^{(j)} I \right) x_\Im^{(j)} ,\]

(4.37)
which avoids having to take the real part of a vector, required when working in complex arithmetic. To determine $x_3$ we look to apply GMRES to the quadratic equation (4.35).

One option, would be to expand the Krylov subspace in the polynomial $\tau^2J^2 - 2\zeta\tau J + |\zeta|^2I$, although this would involve three matrix-vector products per iteration of the Arnoldi process and the savings by using only real arithmetic would likely be sacrificed. A second option is to expand the Krylov subspace using the matrix $J$ and attempt to determine a small least squares problem, as achieved with Shifted GMRES. In fact, this is possible, as is demonstrated in the subsequent analysis.

We attempt to build an approximate solution of the form

$$x^{(k)}_3 = x^{(0)}_3 + V_k y^q_k,$$  \hspace{1cm} (4.38)

and force that this solution minimise the residual two norm over all vectors in the affine space $x^{(0)}_3 + K_k(J, r^{(0)})$, where the residual of the initial guess is now defined by

$$r^{(0)} = \zeta_0 v - (\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2I) x^{(0)}_3.$$

We consider the residual of the approximate solution (4.38):

$$r^{(k)} = \zeta_0 v - (\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2I) x^{(k)}_3$$
$$= \zeta_0 v - (\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2I) (x^{(0)}_3 + V_k y^q_k)$$
$$= r^{(0)} - (\tau^2 J V_k) - 2\zeta \tau (J V_k) + |\zeta|^2V_k y^q_k$$
$$= r^{(0)} - (\tau^2 V_{k+1} H_k - 2\zeta \tau V_{k+1} H_k + |\zeta|^2V_k) y^q_k$$
$$= r^{(0)} - (\tau^2 V_{k+2} H_{k+1} - 2\zeta \tau V_{k+1} H_k + |\zeta|^2V_k) y^q_k,$$

where two consecutive Arnoldi relations have been used

$$J V_k = V_{k+1} H_k,$$ \hspace{1cm} (4.40)
$$J V_{k+1} = V_{k+2} H_{k+1}.$$ \hspace{1cm} (4.41)

This means the solution strategy will be different to the previous GMRES processes with the knowledge of an additional Arnoldi step in advance required to obtain $y^q_k$. Writing

$$r^{(k)} = V_{k+2} (\beta e_1 - H_k y^q_k),$$ \hspace{1cm} (4.42)

where $H^q_k \in \mathbb{R}^{(k+2) \times k}$ we have that

$$H^q_k = \tau^2 H_{k+1} H_k - 2\zeta \tau \begin{pmatrix} H_k & 0_k^T \\ 0_k & 0_k^T \\ 0_k & 0_k^T \end{pmatrix}.$$ \hspace{1cm} (4.43)

Clearly the residual two norm

$$\|r^{(k)}\|_2 = \|\beta e_1 - H^q_k y^q_k\|_2,$$ \hspace{1cm} (4.44)
is minimal when $y^q_k$ is defined by

$$y^q_k = \arg \min_{y \in \mathbb{R}^k} \| \beta e_1 - \bar{H}_k^q y \|_2.$$  \hspace{1cm} (4.45)

Unlike the least squares problems for Standard GMRES (4.23) and Shifted GMRES (4.32) the overdetermined system consists of $(k+2)$ equations in $k$ unknowns. To the extent of our resources, as far as we are aware, a GMRES method for a quadratic equation of this form has not been previously published. Algorithm 4 provides an outline of the strategy for solving $(\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2 I) x_3 = \zeta_3 v$ using GMRES.

Algorithm 4: Quadratic GMRES for $(\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2 I)x_3 = \zeta_3 v$

1. Start
   - Choose $x_3^{(0)}$ and compute $r^{(0)} = \zeta_3 v - (\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2 I)x_3^{(0)}$ and $\beta = \|r^{(0)}\|_2$
   - Set the maximum allowable residual two norm tolerance $TOL$

2. Arnoldi process
   - $v_1 = r^{(0)}/\beta$
   - $w = Jv_1$
   - $h_{1,1} = w^T v_1$
   - $w = w - h_{1,1} v_1$
   - Compute $h_{2,1} = \|w\|_2$ and $v_2 = w / h_{2,1}$
   - for $k = 1$ until convergence do
     - Compute $w = Jv_{k+1}$
     - for $i = 1, \ldots, k+1$ do
       - $h_{i,k} = w^T v_i$
       - $w = w - h_{i,k} v_i$
     - end for
   - Compute $h_{k+2,k+1} = \|w\|_2$ and $v_{k+2} = w / h_{k+2,k+1}$
   - Obtain $\|r^{(k)}\|_2 = \|\beta e_1 - \bar{H}_k^q y_k^q\|_2$ where $y_k^q = \arg \min_{y \in \mathbb{C}^k} \|\beta e_1 - \bar{H}_k^q y\|_2$
   - If $\|r^{(k)}\|_2 < TOL$ convergence attained. Go to 3.
   - Update $V_{k+1} = [V_k, v_{k+2}]$
   - end for

3. Form the approximate solution
   - Compute $x_3 = x_3^{(0)} + V_k y_k^q$ where $y_k^q = \arg \min_{y \in \mathbb{C}^k} \|\beta e_1 - \bar{H}_k^q y\|_2$

Clearly, Algorithm 4 breaks down when $h_{k+2,k+1} = 0$ with division performed with the scalar. In this case the Arnoldi relation (4.40) remains unchanged however (4.41) becomes

$$J V_{k+1} = V_{k+1} H_{k+1}.$$  

With this modified relation, a repeat of the analysis performed in (4.39) yields the following expression for the residual

$$r^{(k)} = V_{k+1} \left( \beta e_1 - \left[ \tau^2 H_{k+1} \bar{H}_k - 2\zeta \tau \bar{H}_k + |\zeta|^2 \begin{pmatrix} I_k & 0_k \end{pmatrix} \right] y_k^q \right).$$

Minimising the residual norm $\|r^{(k)}\|_2$ produces a $(k+1) \times k$ least squares problem and not the exact solution as with Standard and Shifted GMRES.

In summary, thus far, we have presented three different strategies for solving the shifted linear systems arising from best rational approximants to $\varphi(\tau J)v$ for a large, sparse
The following section discusses further implementations regarding each of the methods, namely Standard, Shifted and Quadratic GMRES, discussed to this point.

4.1.4 Practical implementation of GMRES-like methods

A disadvantage of both the shifted and quadratic variants of GMRES lies in attempting to precondition the iterations. The problem stems from expanding the Krylov subspace in $\mathbf{J}$; for the shifted system introducing the right preconditioner $\mathbf{M}^{-1}$ gives

$$\begin{align*}
(\tau \mathbf{J} - \zeta \mathbf{I})\mathbf{M}^{-1}\mathbf{x} &= \mathbf{v}; \\
\mathbf{Mx} &= \mathbf{x}.
\end{align*}$$

Using the standard affine space $\mathbf{x}^{(0)} + \mathbf{M}^{-1}\mathcal{K}_k(\mathbf{J}\mathbf{M}^{-1}, \mathbf{r}^{(0)})$ for right preconditioning and the corresponding Arnoldi relation

$$\mathbf{JM}^{-1}\mathbf{V}_k = \mathbf{V}_{k+1} \mathbf{H}_k,$$

the residual of the approximate solution $\mathbf{x}^{(k)}$ is expressed as

$$\begin{align*}
\mathbf{r}^{(k)} &= \mathbf{v} - (\tau \mathbf{J} - \zeta \mathbf{I})\mathbf{x}^{(k)} \\
&= \mathbf{v} - (\tau \mathbf{J} - \zeta \mathbf{I})(\mathbf{x}^{(0)} + \mathbf{M}^{-1}\mathbf{V}_k y_k^s) \\
&= \mathbf{r}^{(0)} - (\tau \mathbf{J} - \zeta \mathbf{I})\mathbf{M}^{-1}\mathbf{V}_k y_k^s \\
&= \mathbf{r}^{(0)} - (\tau \mathbf{V}_{k+1} \mathbf{H}_k - \zeta \mathbf{M}^{-1}\mathbf{V}_k) y_k^s \\
&= \mathbf{V}_{k+1} \left( \beta \mathbf{e}_1 - (\tau \mathbf{H}_k - \zeta \mathbf{V}^\dagger_{k+1} \mathbf{M}^{-1}\mathbf{V}_k) y_k^s \right),
\end{align*}$$

where $\mathbf{V}^\dagger_{k+1} \in \mathbb{R}^{(k+1)\times N}$ is the right pseudo-inverse of the rectangular matrix $\mathbf{V}_{k+1}$, defined by

$$\mathbf{V}^\dagger_{k+1} = \mathbf{V}^T_{k+1}(\mathbf{V}_{k+1} \mathbf{V}^T_{k+1})^{-1}.$$

Unlike GMRES it is impossible to obtain a least squares problem independent of $\mathbf{V}_{k+1}$ or $\mathbf{V}_k$. Not surprisingly, the same problem is encountered for the quadratic equation where a reduced problem independent of $\mathbf{V}_{k+2}$, $\mathbf{V}_{k+1}$ or $\mathbf{V}_k$ cannot be formed. In practice this is problematic, since the use of a preconditioner is invaluable in improving the convergence rate. However, for problems where convergence is slow, preconditioned iterations for Standard GMRES most definitely can be achieved.

In practice for each of the methods discussed so far, we expand the Krylov subspace until the residual two norm is $\|\mathbf{r}^{(k)}\|_2$ is below some specified tolerance $\text{TOL}$ as featured in Algorithms 2, 3 and 4. In the following sections we describe how to obtain $\|\mathbf{r}^{(k)}\|_2$ at each stage of the solution process, in each of the methods discussed above, without the need to solve the least squares problems (4.23), (4.32) and (4.45). This is very beneficial from a practical point of view.
Standard GMRES

In general the problem (4.23) is solved using the QR-factorisation. Due to the upper Hessenberg structure of the coefficient matrix (4.20)

\[
\mathbf{H}_k = \begin{pmatrix}
h_{11} & h_{12} & h_{13} & \ldots & h_{1k} \\
h_{21} & h_{22} & h_{23} & \ldots & h_{2k} \\
h_{32} & h_{33} & \ldots & h_{3k} \\
\vdots & \vdots & \ddots & \ddots \\
h_k & & & & h_{k+1,k}
\end{pmatrix},
\]

the non-zero entries beneath the main diagonal \( (h_{i+1,i}, i = 1, \ldots, k) \) can be annihilated by repeated application of Givens’ rotations (Givens 1958). By multiplying both sides of (4.24) by the sequence of matrices \( \{\Omega_i \in \mathbb{C}^{(k+1)\times(k+1)}, i = 1, \ldots, k\} \)

\[
\Omega_i = \begin{pmatrix}
1 & & & & \\
& \ddots & & & \\
& & c_i & s_i & \\
& & -s_i & c_i & \\
& & & \ddots & \\
& & & & 1
\end{pmatrix} \quad \text{← row } i \quad \text{← row } i+1
\] (4.47)

we successively “zero” the entries \( h_{i+1,i}, i = 1, \ldots, k \), of the coefficient matrix \( \mathbf{H}_k \). The values of \( c_i \) and \( s_i \) are given as (Saad 2003)

\[
c_i = \frac{h_{ii}}{\sqrt{|h_{ii}|^2 + h_{i+1,i}^2}}; \quad s_i = \frac{h_{i+1,i}}{\sqrt{|h_{ii}|^2 + h_{i+1,i}^2}},
\] (4.48)

where the most up-to-date values of \( h \) are used in the computations and \( c_i^2 + s_i^2 = 1 \) so that \( \Omega_i^H \Omega_i = I \). By denoting \( \mathbf{Q}_k \) as the product of these \( k \) matrices then it clear from this last property that \( \mathbf{Q}_k^H \mathbf{Q}_k = I \) and that \( \mathbf{Q}_k \) is unitary. Setting \( \mathbf{R}_k = \mathbf{Q}_k \mathbf{H}_k \) and \( \mathbf{g}_k = \mathbf{Q}_k \beta \mathbf{e}_1 \) the transformed problem is

\[
\mathbf{R}_k \mathbf{y}_k = \mathbf{g}_k,
\] (4.49)

or

\[
\begin{pmatrix}
\mathbf{R}_k \\
\mathbf{Q}_k^T
\end{pmatrix} \mathbf{y}_k = \begin{pmatrix}
\mathbf{g}_k \\
g_{k+1}
\end{pmatrix},
\] (4.50)

where \( \mathbf{R}_k \in \mathbb{C}^{k\times k} \) is upper triangular and \( g_{k+1} \) denotes the last entry of \( \mathbf{g} \). Furthermore, since \( \mathbf{Q}_k \) is unitary and the last row of \( \mathbf{R}_k \) is zero, the minimisation of the residual is achieved by solving the square \( k \times k \) upper triangular linear system for \( \mathbf{y}_k \). That is (4.23)
is reformulated as follows

\[
y_k = \arg \min_{y \in \mathbb{C}^k} \| \beta e_1 - \mathbf{H}_k y \|_2
\]

\[
= \arg \min_{y \in \mathbb{C}^k} \| Q_k (\beta e_1 - \mathbf{H}_k y) \|_2
\]

\[
= \arg \min_{y \in \mathbb{C}^k} \| g_k - R_k y \|_2
\]

\[
= \arg \min_{y \in \mathbb{C}^k} \| g_k - R_k y \|_2
\]

\[
= R_k^{-1} g_k.
\]

(4.51)

From (4.50) it is clear that

\[
\| r^{(k)} \|_2^2 = \| g_k - R_k y_k \|_2^2 + |g_{k+1}|^2,
\]

and since \( y_k \) is the exact solution of the upper triangular system, the residual two norm at each stage of the process is given by

\[
\| r^{(k)} \|_2 = |g_{k+1}|.
\]

(4.52)

This avoids the need to solve (4.23) for \( y_k \) at all to determine when the solution is sufficiently accurate. Furthermore, it is evident that

\[
\mathbf{H}_k = \begin{pmatrix}
\mathbf{H}_{k-1} & \mathbf{\Pi}_k \\
0^T & \mathbf{I}_k
\end{pmatrix},
\]

where \( \mathbf{\Pi}_k \in \mathbb{C}^{(k+1) \times k} \) are the entries produced at the \( k \)th step of the Arnoldi process. This means we need only to annihilate the entry in the \( k \)th column of \( \mathbf{H}_k \) as the sub main-diagonal entries in \( \mathbf{H}_{k-1} \) were annihilated at the previous iteration. Applying the rotations \( \Omega_{k-1} \ldots \Omega_1 \) to the new column, the new rotation matrix \( \Omega_k \) is constructed based on the entries of \( \Omega_{k-1} \ldots \Omega_1 \mathbf{\Pi}_k \).

In exact arithmetic GMRES converges to the exact solution in at most \( N \) steps (Frayssé et al. 2005). Recall, however that in our Richards’ equation test problem, \( N \) denotes the number of nodes and is large and hence as \( k \) approaches \( N \) the least squares problem (4.23) is large and the storage of \( \mathbf{V}_k \) prohibitive. To guard against large \( k \) the process is restarted every \( m \) iterations. In this case \( r^{(0)} \) is set to be the residual of the approximation \( \mathbf{x}^{(m)} \) and \( \mathbf{x}^{(0)} \) is set to \( \mathbf{x}^{(m)} \) and the process repeated. Fortunately, the residual can be computed without the need for the matrix-vector product \( (\tau J - \zeta I) \mathbf{x}^{(m)} \). Using (4.21)
with \( k = m \) and by noting that \( Q_m^H Q_m = I \) produces

\[
\tau^{(m)} = V_{m+1}(\beta e_1 - \bar{H}_m y_m)
= V_{m+1} Q_m^H Q_m(\beta e_1 - \bar{H}_m y_m)
= V_{m+1} Q_m^H (\bar{g}_m - \bar{R}_m y_m)
= V_{m+1} Q_m^H g_{m+1} e_{m+1}.
\]

where \( e_{m+1} \) is the \((m+1)\)th canonical basis vector in \( \mathbb{R}^{m+1} \).

**Shifted GMRES**

Clearly the upper \( k \times k \) block of \( \bar{H}_k \) is upper Hessenberg, with the shift induced by the pole \( \zeta \) altering only the non-zero diagonal entries of \( H_k \). Furthermore to obtain the coefficient matrix \( \bar{H}_k \) at each step of the process we only need to update the matrix \( \bar{H}_{k-1} \) from the previous step, that is,

\[
\bar{H}_k^s = \tau \bar{H}_k - \zeta \begin{pmatrix} I_k \\ 0^T \end{pmatrix}
= \begin{pmatrix} \bar{H}_{k-1}^s \\ 0^T_{k-1} \end{pmatrix} + \tau \bar{h}_k - \zeta e_k,
\]

where \( e_k \) is the \( k \)th canonical basis vector in \( \mathbb{R}^{k+1} \) and \( \bar{h}_k \) is as defined previously. This means that the residual norm can be obtained at each stage of the process identically to that discussed for **Standard GMRES**, with rotations only required to be applied to the updated column of \( \bar{H}_k^s \).

**Quadratic GMRES**

The structure of the coefficient matrix of the corresponding rectangular system of (4.45)

\[
\bar{H}_k^s y_k = \beta e_1,
\]

is not obvious at first and is clearly different to the structure of \( \bar{H}_k \) and \( \bar{H}_k^s \), since the dimensions do not match. To investigate further we consider the product \( \bar{H}_{k+1} \bar{H}_k \), which can be expressed as

\[
\bar{H}_{k+1} \bar{H}_k = \begin{pmatrix} \bar{H}_k \\ 0^T_k \end{pmatrix} \begin{pmatrix} \bar{h}_{k+1} \\ 0^T_{k+1} \end{pmatrix} + h_{k+1,k} \begin{pmatrix} \bar{h}_{k+1,k} e_{k+1} e_k^T + \bar{h}_{k+1,k} e_k e_{k+1}^T + \bar{h}_{k+1,k} e_{k+1} e_k^T \end{pmatrix}.
\]

where all vectors are members of \( \mathbb{R}^{k+2} \). To proceed the knowledge of the sparsity structure of the matrix \( H_k^2 \) or in general, the product of an upper Hessenberg matrix with itself
is required. This commands the following proposition for which we provide a proof via mathematical induction.

**Proposition:** Consider the upper Hessenberg matrix $H_k = (h_{ij}) \in \mathbb{R}^{k \times k}$ from the Arnoldi process whose entries satisfy $h_{ij} = 0$ for $i > j + 1$. Then the matrix $\Lambda_k = (\lambda_{ij}) = H_k^2$ is such that $\lambda_{ij} = 0$ for $i > j + 2$.

**Proof:** The case $k = 1$ is trivially true. Assuming the proposition holds for $k = m$ we consider $k = m + 1$,

\[
\Lambda_{m+1} = H_{m+1}^2 \\
= \begin{pmatrix} H_m & h_{m+1} \\ 0_m & h_{m+1,m} \end{pmatrix} \begin{pmatrix} H_m & h_{m+1} \\ 0_m^T & h_{m+1,m} \end{pmatrix} \\
= \begin{pmatrix} H_m^2 & H_m h_{m+1} + h_{m+1,m} h_{m+1} \\ 0_m^T & h_{m+1,m}^2 \end{pmatrix}
\]

which satisfies $\lambda_{ij} = 0$ for $i > j + 2$ under the assumption. Therefore by mathematical induction the proposition is true for all $k \in \mathbb{N}$.

By the above proposition, squaring the upper Hessenberg matrix $H_k$ adds an additional diagonal of non-zero entries to the sparsity structure. Therefore from the result given in (4.54), it clear that the matrix $H_{k+1}^t H_k$ consists of an upper triangular matrix in the upper $k \times k$ block with two additional diagonals of non-zero entries beneath the main diagonal. Furthermore, from the form (4.43) it is apparent that the remaining terms do not alter this structure and hence the coefficient matrix of the least squares problem $H_k^t$ possesses the structure illustrated below:

\[
H_k^t = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1k} \\
h_{21} & h_{22} & h_{23} & \cdots & h_{2k} \\
h_{31} & h_{32} & h_{33} & \cdots & h_{3k} \\
h_{41} & h_{42} & h_{43} & \cdots & h_{4k} \\
\ddots & \ddots & \ddots & \ddots & \ddots \\
h_{k+1,1} & \cdots & h_{k+1,k} \\
h_{k+2,1} & \cdots & h_{k+2,k} \\
\end{pmatrix}
\]

In a strategy similar to the least squares problems for the previous two GMRES methods, the rectangular system (4.53) can be transformed to an upper triangular form via Givens’ rotations. First, the lower non-zero diagonal $(h_{i+2,i}, i = 1, \ldots, k)$ is annihilated by multiplying both sides of (4.53) by the sequence of rotation matrices $\{\Omega_i^{(2)}, i = 1, \ldots, k\}$ where
\( \Omega_i^{(2)} \in \mathbb{R}^{(k+2) \times (k+2)} \) is defined by

\[
\Omega_i^{(2)} = \begin{pmatrix}
1 & & & \\
 & \ddots & & \\
 & & c_i & s_i \\
& & -s_i & c_i \\
& & & \ddots \\
& & & & 1
\end{pmatrix} \quad \leftarrow \text{row } i+1
\]

\[
\Omega_i^{(2)} = \begin{pmatrix}
1 & & & \\
 & \ddots & & \\
 & & c_i & s_i \\
& & -s_i & c_i \\
& & & \ddots \\
& & & & 1
\end{pmatrix} \quad \leftarrow \text{row } i+2
\]

and \( c_i \) and \( s_i \) are given as

\[
c_i = \frac{h_{i+1,i}}{\sqrt{h_{i+1,i}^2 + h_{i+2,i}^2}}; \quad s_i = \frac{h_{i+2,i}}{\sqrt{h_{i+1,i}^2 + h_{i+2,i}^2}}.
\]

Secondly, to annihilate the diagonal directly below the leading diagonal \((h_{i+1,i}, i = 1, \ldots, i)\) we multiply by the sequence of matrices \(\{\Omega_i^{(1)}, i = 1, \ldots, k\}\) where \(\Omega_i^{(1)} \in \mathbb{R}^{(k+2) \times (k+2)}\) is the \((k+2) \times (k+2)\) analogue of (4.47) for real entries \(h_{ii}\) and \(h_{i+1,i}\) and real values of \(c_i\) and \(s_i\). We will let \(Q_k^{(2)}\) denote the product of the first \(k\) rotation matrices and \(Q_k^{(1)}\) the second. Redefining \(R_k = Q_k^{(1)} Q_k^{(2)} H_k^q\) and \(g_k = Q_k^{(1)} Q_k^{(2)} \beta e_1\) then the rectangular system (4.53) is transformed to

\[
R_k y_k^q = g_k;
\]

or

\[
\begin{pmatrix}
R_k \\
0_k^T \\
0_k^T
\end{pmatrix} y_k^q = \begin{pmatrix}
g_k \\
g_k+1 \\
g_k+2
\end{pmatrix}, \quad (4.55)
\]

where \(R_k\) is upper triangular. As with GMRES, due to the unitary properties of the matrices \(Q_k^{(2)}\) and \(Q_k^{(1)}\) the problem of minimising the residual two norm of (4.53) is unaffected and the minimisation is achieved by solving the upper triangular system exactly for \(y_k^q\).

\[
y_k^q = \arg \min_{y \in \mathbb{R}^k} \| \beta e_1 - H_k^q y \|_2
\]

\[
= \arg \min_{y \in \mathbb{R}^k} \| Q_k^{(1)} Q_k^{(2)} (\beta e_1 - H_k^q y) \|_2
\]

\[
= \arg \min_{y \in \mathbb{R}^k} \| g_k - R_k y \|_2
\]

\[
= \arg \min_{y \in \mathbb{R}^k} \| g_k - R_k y \|_2
\]

\[
= R_k^{-1} g_k.
\]
From (4.55) it is clear that
\[
\|r^{(k)}\|_2^2 = \|g_k - R_k y_k^T\|_2^2 + g_{k+1}^2 + g_{k+2}^2,
\]
and since \(y_k^T\) is the exact solution of the upper triangular system, the residual two norm at each stage of the process is given by
\[
\|r^{(k)}\|_2 = \sqrt{g_{k+1}^2 + g_{k+2}^2}.
\] (4.56)

Again we can delay solving the least squares problem until the approximate solution is formed. Furthermore by using the form (4.54) it is evident \(H^q_k\) can be progressively updated at each stage of the process
\[
H^q_k = \tau^2 H_{k+1} - 2\zeta \tau H_k + |\zeta|^2 \left( \begin{array}{c} I_k \\ 0_k^T \end{array} \right)
\]
where \(e_k\) is the \(k\)th canonical basis vector in \(R^{k+2}\). This means that a rotation matrix need only be constructed for this additional column. To restart the process, the residual after \(m\) steps can be obtained via
\[
r^{(m)} = V_{m+2}(\beta e_1 - H^q_m y_m)
\]
\[
= V_{m+2} Q_m^{(2)^T} Q_m^{(1)^T} Q_m^{(1)} Q_m^{(2)} (\beta e_1 - H^q_m y_m)
\]
\[
= V_{m+2} Q_m^{(2)^T} Q_m^{(1)^T} (g_m - R_m y_m)
\]
\[
= V_{m+2} Q_m^{(2)^T} Q_m^{(1)^T} (g_{m+1} e_{m+1} + g_{m+2} e_{m+2}).
\]

Next, we conduct a short numerical experiment investigating the convergence rates of the three GMRES-like methods presented thus far.

4.1.5 Investigation into the convergence of GMRES-like methods

Algorithms 2, 3 and 4 were implemented without restarting with a representative Jacobian matrix taken from the test problem stated in §2.2. The matrix \(J\) and the vector \(v\) from both equations (4.15) and (4.35) satisfied
\[
\text{cond}(J) = 9.66 \times 10^{10}; \quad \|v\|_2 = 1.24 \times 10^{-6}.
\]
The value of \(\zeta\) was taken as the first pole, \(\zeta^{(1)}\), in Table 4.1 while we chose \(\tau = 10000\). The stopping criterion was selected as \(\|r^{(k)}\|_2 < 10^{-12}\) with the zero vector chosen as the starting iterate. This choice of \(x^{(0)}\) for the full variants of Shifted GMRES is equivalent to altering the search space to simply \(K_k(J, v)\), while for Full Quadratic GMRES the
choice $x_\zeta^{(0)} = 0$ implies the change of the search space to $K_k(J, \zeta \nu)$. The two norm of the residual at each step of each of the iterative processes is represented graphically in Figure 4.3. The value of $\|r^{(0)}\|_2 \approx 10^{-6}$ confirms the zero vector is a good choice for the starting iterate.

Figure 4.3: Plot of $\|r^{(k)}\|_2$ for Algorithms 2, 3 and 4 throughout the solution process for the numerical experiment described in §4.1.5.

Without the use of a preconditioner convergence rates of the shifted and standard variants of GMRES were identical. The rate of convergence for the quadratic variant is visibly very similar. These results were typical for all simulations performed with the test problem. Based on the results of this experiment, we propose, in the subsequent section, an improved strategy for approximating $\varphi(\tau J)\nu$ via best rational approximants.

4.1.6 Strategies using a single Krylov subspace

A downfall of the GMRES strategies, proposed in §4.1.3, is that each of the Algorithms 2, 3 and 4 must be applied seven times, for each of the 7 shifted linear systems (4.13) required to form the approximation to $\varphi(\tau J)\nu$. Since the basis of the Krylov subspace is constructed with the $\zeta$-dependent initial residual vector $r^{(0)}$, in general the subspaces are different for each shifted system (4.15).

However, if an approximate solution was sought from the space $K_k(J, \nu)$ then the Arnoldi basis construction need only be performed once, reducing the number of Jacobian-vector products by about 1/7. Although, one notes that the process cannot be restarted and therefore is only suitable provided convergence is fast and $k$ remains small. This assumption seems to be reasonable for the test problem based on the results of the previous section where the value of $k$ did not exceed 16.

This time, building approximate solutions of the form:

$$x^{(k)} = V_k y_k,$$

$$x_\zeta^{(k)} = V_k y^{(k)}_{\zeta}.$$  

(4.57)

(4.58)
where the columns \( V_k \) form an orthonormal basis of \( K_k(J, v) \) and repeating the analysis on the residual in (4.28) and (4.39) we obtain the following expressions for \( y_s^k \) and \( y_q^k \):

\[
y_s^k = \arg\min_{y \in \mathbb{C}^k} \| \beta e_1 - \Pi_k y \|_2, \tag{4.59}
\]

\[
y_q^k = \arg\min_{y \in \mathbb{R}^k} \| \zeta \beta e_1 - \Pi_k y \|_2, \tag{4.60}
\]

where \( \beta = \|v\|_2 \). The latter expressions depend on \( \zeta \) and thus the minimisation problems (4.59) and (4.60) must be considered separately for each pole \( \zeta^{(j)} \), \( j = 1, \ldots, 7 \). The proposed methods are presented in Algorithms 5 and 6.

**Algorithm 5: Best rational approximant to \( \varphi(\tau J) v \) via Shifted GMRES**

1. **Start**
   
   Compute \( \beta = \|v\|_2 \) and \( v_1 = v / \beta \) and set the global residual two norm tolerance \( TOL \).

2. **Arnoldi process**
   
   for \( k = 1 \) until convergence do
   
   Compute \( w = Jv_k \)
   
   for \( i = 1, \ldots, k \) do
   
   \( h_{i,k} = w^T v_i \)
   
   \( w = w - h_{i,k} v_i \)
   
   end for
   
   Compute \( h_{k+1,k} = \|w\|_2 \) and \( v_{k+1} = w / h_{k+1,k} \)
   
   for each pole not converged
   
   Obtain \( \|r^{(k)}\|_2 = \|\beta e_1 - \Pi_k y_k\|_2 \) where \( y_k = \arg\min_{y \in \mathbb{C}^k} \| \beta e_1 - \Pi_k y \|_2 \)
   
   end if
   
   If all \( \|r^{(k)}\|_2 < TOL \) convergence attained. Go to 3.
   
   Update \( V_{k+1} = [V_k, v_{k+1}] \)
   
   end for

3. **Form the approximate solutions of** \( (\tau J - \zeta^{(j)} I)x^{(j)} = v \) for \( j = 1, \ldots, 7 \)
   
   Compute \( x^{(j)} = V_k y_k^{(j)} \) where \( y_k^{(j)} = \arg\min_{y \in \mathbb{C}^k} \| \beta e_1 - \Pi_k y \|_2 \)

4. **Form the approximation to** \( \varphi(\tau J) v \)
   
   Compute \( \varphi(\tau J) v = \alpha^{(0)} v + 2 \Re \left\{ \sum_{j=1}^{7} \alpha^{(j)} (\tau J - \zeta^{(j)} I)^{-1} v \right\} \)

The implementations discussed above result in an increase in the storage requirements. This is because for each pole \( \zeta \) the coefficient matrix \( \Pi_k \) (\( \Pi_k^T \)) and right-hand-side vector \( \beta e_1 (\zeta \beta e_1) \) must be stored simultaneously. Although, this appears minimal in comparison with the savings obtained from the reduction in the number of matrix-vector products involving the Jacobian. The implementation of Givens’ rotations for transforming each of the seven coefficient matrices into an upper triangular form and obtaining each subsequent residual norm remains unchanged from that described in §4.1.4. Requiring each residual two norm to be below a specified global tolerance \( TOL \) determined the stopping criterion for the Arnoldi process.

The two methods outlined in Algorithms 5 and 6, appear very promising for efficiently approximating the term \( \varphi(\tau J_n) G_n \) required in the exponential Euler scheme. In the remaining sections of this chapter an established technique for large, sparse approximations to \( \varphi(\tau J) v \), also involving the Krylov subspace, is described.
In order to extract such an approximation an orthonormal basis \( K \) of \( \mathbb{C}^n \) is designed for large, sparse computations of \( \phi(\tau J) \). Krylov subspace approximations are designed for large, sparse computations of \( \phi(\tau J) \).

\[ 4.2 \text{ Krylov subspace approximants} \]

Krylov subspace approximations are designed for large, sparse computations of \( \phi(\tau J) \) with matrix-vector products of \( J \) and a vector, the only operations featuring the large matrix. The basic idea behind these techniques is to approximately project the evaluation of the large matrix onto a small Krylov subspace (Saad 1992). This is very attractive from a computational point of view, since the only evaluation of \( \phi \) is performed on a matrix of much smaller dimension.

Krylov subspace techniques build an approximation to \( \phi(\tau J) \) from the subspace

\[ K_k(J, \nu) = \text{span} \left\{ \nu, J \nu, \ldots, J^{k-1} \nu \right\}. \]  

(4.61)

In order to extract such an approximation an orthonormal basis \( V_k = (v_1, v_2, \ldots, v_k) \) of \( K_k(J, \nu) \) is generated via the Arnoldi process satisfying the relation:

\[ J V_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T. \]  

(4.62)

Multiplying throughout by \( \tau V_k^T \) yields:

\[ V_k^T(\tau J) V_k = V_k^T V_k(\tau H_k) + \tau h_{k+1,k} V_k^T v_{k+1} e_k^T. \]
Due to the orthogonality of the columns of $V_{k+1}$, the product $V_k^T v_{k+1} = 0$ and $V_k^T V_k = I$ and hence $\tau H_k$ is the orthogonal projection of $\tau J$ onto the Krylov subspace $K_k(J, v)$:

$$\tau H_k = V_k^T (\tau J) V_k.$$  \hfill (4.63)

The goal is to determine the unique vector contained in $K_k(J, v)$ which is the best approximation to $\varphi(\tau J)v$. This approximation is given by the orthogonal projection of $\varphi(\tau J)v$ onto $K_k(J, v)$:

$$P_{K_k} (\varphi(\tau J)v) = V_k V_k^T \varphi(\tau J)v.$$  \hfill (4.64)

Furthermore, since the first orthonormal basis vector $v_1$ from the Arnoldi process is $v$ scaled by its length this approximation can be written as

$$P_{K_k} (\varphi(\tau J)v) = \beta V_k V_k^T \varphi(\tau J)v_1,$$  \hfill (4.64)

where $\beta = \|v\|_2$. However, this expression still contains the evaluation of $\varphi$ at the large matrix $\tau J$ and offers no benefit over computing $\varphi(\tau J)v$ directly. Motivated from the result (4.63), an accepted approach in the literature (Saad 1992, Minchev 2004, Hochbruck et al. 1996, Hochbruck 2004) is to replace $V_k^T \varphi(\tau J)v_k$ with $\varphi(\tau H_k)$. The resulting approximation is

$$\varphi(\tau J)v \approx \beta V_k \varphi(\tau H_k)v_1,$$  \hfill (4.65)

where $k$ is typically small and hence the cost of computing $\varphi(\tau H_k)$ is minimal. For a polynomial matrix function $p$ of degree less than $k$, equivalence is obtained, that is

$$p(\tau J)v = \beta V_k p(\tau H_k)v_1,$$

and hence the approximation for the analytic function $\varphi$ seems natural. The equivalent result for the matrix exponential $e^{\tau J}v$ has been used throughout the literature (Saad 1992, Sidje 1998) exhibiting remarkable success with very accurate approximations observed for relatively small values of $k$.

To compute the resulting approximation, methods that evaluate $\varphi(\tau H_k)$ explicitly or $\varphi(\tau H_k)v_1$ directly can be used. The latter option could be achieved using best rational approximations and since $H_k$ is small, direct methods could be used to solve the resulting shifted linear systems $(\tau H_k - \zeta I)x = e_1$. This seems favourable for a direct approximation of $\varphi(\tau J)v$. However, in the exponential Euler and exponential Rosenbrock schemes numerous such evaluations are required per time step. We demonstrate in the next section why explicit computation of $\varphi(\tau H_k)$ is favoured due to special properties of the $\varphi$ function.
4.2.1 Recurrence relations for the \( \varphi \) function

An appealing feature of working explicitly with the matrix function \( \varphi(\tau H_k) \) comes from the ability to use the relationships:

\[
\varphi(2A) = \left( \frac{1}{2}A\varphi(A) + I \right)\varphi(A); \tag{4.67}
\]
\[
\varphi(3A) = \frac{2}{3}(A\varphi(A) + I)\varphi(2A) + \frac{1}{3}\varphi(A), \tag{4.68}
\]

which are easily proven from the form of \( \varphi(A) \). These properties allow \( \varphi(2A) \) and \( \varphi(3A) \) to be obtained from a single matrix function evaluation of \( \varphi(A) \). These results are very useful for the adaptive time stepping strategy for the exponential Euler method proposed in §3.3. Recall, the difference between the solution obtained using the full time step \( \tau \) with the solution obtained using two half time steps of \( \frac{\tau}{2} \) is used to calculate \( \text{err} \). The former requires the evaluation \( \varphi(\tau J)v \), while the latter requires \( \varphi(\frac{\tau}{2} J)v \) where \( J = J(u_n) \).

Using (4.65) the Krylov subspace approximation to \( \varphi(\frac{\tau}{2} J)v \) is

\[
\varphi(\frac{\tau}{2} J)v \approx \beta V_k\varphi(\frac{1}{2} \tau H_k)e_1.
\]

Computing \( \varphi(\frac{1}{2} \tau H_k) \) explicitly, the evaluation of \( \varphi(\tau H_k) \) required in the approximation

\[
\varphi(\tau J)v \approx \beta V_k\varphi(\tau H_k)e_1,
\]

can be easily extracted via a matrix product using the relation (4.67):

\[
\varphi(\tau H_k) = \left( \frac{1}{2} \tau H_k\varphi(\frac{1}{2} \tau H_k) + I \right)\varphi(\frac{1}{2} \tau H_k).
\]

These relations can also be used to reduce the number of evaluations of \( \varphi \) required in the exponential Rosenbrock method. According to Hochbruck et al. (1996) this is a key point for the efficient implementation of the method. The scheme requires the computation of \( \varphi(\frac{1}{2} \tau J)v \), \( \varphi(\frac{3}{2} \tau J)v \) and \( \varphi(\tau J)v \) for two different vectors \( v \) as described in §3.2. It is clear from both relations (4.67) and (4.68) that each of \( \varphi(\frac{3}{2} \tau H_k) \) and \( \varphi(\tau H_k) \) can be obtained from \( \varphi(\frac{1}{2} \tau H_k) \), as given by

\[
\varphi(\frac{3}{2} \tau H_k) = \left( \frac{1}{2} \tau H_k\varphi(\frac{1}{2} \tau H_k) + I \right)\varphi(\frac{1}{2} \tau H_k),
\]
\[
\varphi(\tau H_k) = \left( \frac{1}{2} \tau H_k\varphi(\frac{1}{2} \tau H_k) + I \right)\varphi(\frac{3}{2} \tau H_k) + \frac{1}{3}\varphi(\frac{1}{2} \tau H_k).
\]

This reduces the original seven \( \varphi \) function evaluations in the scheme to three. The fact, that it is impossible to recover \( \varphi(\tau H_k)e_1 \) from \( \varphi(\frac{1}{2} \tau H_k)e_1 \) as well as \( \varphi(\tau H_k)e_1 \) and \( \varphi(\frac{3}{2} \tau H_k)e_1 \) from \( \varphi(\frac{1}{2} \tau H_k)e_1 \), is the reason why methods that explicitly compute the matrix function have been favoured in this work.

4.2.2 Padé approximation

To compute \( \varphi(\tau H_k) \) we follow Hochbruck et al. (1996) and use the Padé approximation implemented with scaling and squaring. This is a variant of one of the most effective
methods for small dense evaluations of the matrix exponential as explained by Moler & van Loan (2003). The Padé approximation is a rational function analogue of the Taylor polynomial, with the \((n,m)\) Padé approximant matching the Taylor series expansion up to order \(n+m\). The coefficients of the polynomials of \(P_{mn}(z) = p_m(z)/q_n(z)\) are obtained by solving the algebraic equation

\[
\sum_{k=1}^{\infty} \frac{1}{k!} z^{k-1} - \frac{p_m(z)}{q_n(z)} = O(z^{m+n+1}).
\]

In the case of the matrix exponential, for stability and reduced computational cost, it is advantageous to use the diagonal approximant and set \(m = n = d\) (Moler & van Loan 2003). We follow both Berland et al. (2005) and Hochbruck et al. (1996) and take this approach. A closed form expression for the diagonal Padé approximant to \(\varphi(z)\) exists and is given by (Berland et al. 2005):

\[
p_d(z) = \frac{d!}{(2d+1)!} \sum_{i=0}^{d} \left[ \sum_{j=0}^{i} \frac{(2d+1 - j)!(-1)^j}{j!(d-j)!(l+i-j)!} \right] z^i,
\]

\[
q_d(z) = \frac{d!}{(2d+1)!} \sum_{i=0}^{d} \frac{(2d+1 - i)!}{i!(d-i)!} (-z)^i.
\]

Our implementation used the \((6,6)\) Padé approximant

\[
P_{66}(z) = \frac{1 + \frac{1}{12} z + \frac{5}{14}\overline{5} z^2 + \frac{1}{14}\overline{7} z^3 + \frac{1}{14}\overline{720} z^4 + \frac{1}{2100}\overline{3} z^5 + \frac{1}{84}\overline{16!} z^6}{1 - \frac{6}{13} z + \frac{5}{12} z^2 - \frac{5}{129}\overline{3} z^3 + \frac{1}{1144}\overline{7} z^4 - \frac{1}{25740}\overline{5} z^5 + \frac{1}{12355360}\overline{6} z^6},
\]

via the \texttt{phipade.m} routine from Berland et al. (2005), which uses Horner evaluations (Horner 1819) to efficiently evaluate the numerator and denominator polynomials. Two problems exist with the direct evaluation of \(\varphi(\tau H_k)\) using the Padé approximant. The first is that when \(\tau H_k\) has widely spread eigenvalues, the computation of \(P_{pp}(\tau H_k)\) involves an ill-conditioned denominator matrix. Furthermore, Padé approximations are only accurate near the origin, so that the assumption \(\varphi(\tau H_k) \approx P_{pp}(\tau H_k)\) is only valid when \(||\tau H_k||_2\) is not too large.

To retain the high accuracy about the origin, in \texttt{phipade.m} the approximant is applied to a scaled version of the original matrix whose norm is \(O(1)\). The scaling is then reversed by applying the “squaring”. An integer power of two scaling is implemented which allows the relation (4.67) to be used. However, Hochbruck et al. (1996) claim that (4.67) is unstable for large \(||A||_2\) due to the multiplication with \(A\). An alternate stable evaluation which uses (4.67) initially and subsequently performs the squaring on the exponential is implemented in \texttt{phipade.m}. We briefly summarise the algorithm below:

1. Determine the smallest nonnegative integer \(s\) such that \(2^s \geq ||\tau H_k||_2\).
2. Set \(H_s = \tau H_k/2^{\max(0,s+1)}\) and compute \(\varphi(H_s) \approx P_{66}(H_s)\).
3. Compute \(e^{H_s} = H_s \varphi(H_s) + I\) and \(\varphi(H_s) = \frac{1}{2} (e^{H_s} + I) \varphi(H_s)\).
4. Reverse scaling by repeated application of:

\[ e^{2H_x} = (e^{H_x})^2 \]

\[ \varphi(2H_x) = \frac{1}{2}(e^{2H_x} + I)\varphi(H_x). \]

5. Return Padé approximant to \( \varphi(\tau H_k) \).

### 4.2.3 Practical implementation

A criterion for determining when the value of \( k \) is sufficiently large to ensure the Krylov approximation (4.65) is an accurate approximation to \( \varphi(\tau J)v \) is needed. We have implemented an approach taken by Hochbruck et al. (1996), that involves approximating the true error of the approximation, defined by:

\[ \epsilon_k = \beta V_k \varphi(H_k)e_1 - \varphi(\tau J)v. \]  \hspace{1cm} (4.70)

Since the \( \varphi \) function is entire, using the Cauchy Integral formula, we can write

\[ \varphi(\tau J)v = \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) (\lambda I - \tau J)^{-1} v \, d\lambda, \]

and hence

\[ \beta V_k \varphi(\tau H_k)e_1 = \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) \beta V_k (\lambda I_k - \tau H_k)^{-1} e_1 \, d\lambda, \]

where \( \Gamma \) encloses the eigenvalues of both \( \tau J \) and \( \tau H_k \). This means the exact representation of the error (4.70) can be expressed as

\[ \epsilon_k = \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) \left( (\lambda I - \tau J)^{-1} v - \beta V_k (\lambda I_k - \tau H_k)^{-1} e_1 \right) \, d\lambda. \]  \hspace{1cm} (4.71)

The term

\[ (\lambda I - \tau J)^{-1} v - \beta V_k (\lambda I_k - \tau H_k)^{-1} e_1, \]  \hspace{1cm} (4.72)

in the integrand is the exact error associated with the \( k^{th} \) Krylov approximation to the solution of the linear system \( (\lambda I - \tau J)x = v \) from the subspace \( K_k(J, v) \) using the Full Orthogonalisation Method (FOM). FOM uses the Ritz-Galerkin condition imposing that the residual be orthogonal to \( K_k \), that is,

\[ \begin{align*}
    V_k^T (v - (\lambda I - \tau J)V_k y_k) &= 0 \\
    V_k^T v - (\lambda I - \tau V_k^T J V_k) y_k &= 0 \\
    (\lambda I - \tau H_k) y_k &= V_k^T v \\
    (\lambda I - \tau H_k) y_k &= \beta e_1.
\end{align*} \]
The approximate solution is then

\[ x^{(k)} = V_k y_k = \beta V_k (\lambda I - \tau H_k)^{-1} e_1, \quad (4.73) \]

and hence with the exact solution equal to \((\lambda I - \tau J)^{-1} v\), the error is as given in (4.72). Of course, this value is not available in the FOM iterative process since the exact solution \((\lambda I - \tau J)^{-1} v\) is unknown. In FOM, as with GMRES, the stopping criterion is based on the residual of the approximate solution \(x^{(k)}\), simplified using the Arnoldi relation (4.62):

\[
\begin{align*}
    r^{(k)} &= v - (\lambda I - \tau J)x^{(k)} \\
    &= v - (\lambda I - \tau J)V_k y_k \\
    &= v - (\lambda V_k - \tau JV_k)y_k \\
    &= v - (\lambda V_k - \tau V_k H_k - \tau h_{k+1,k} v_{k+1} e_k^T) y_k \\
    &= V_k (\beta e_1 - (\lambda I - \tau H_k) y_k) + \tau h_{k+1,k} v_{k+1} e_k^T y_k \\
    &= \tau h_{k+1,k} y_{k+1} e_k^T (\lambda I - \tau H_k)^{-1} e_1 \\
    &= \beta \tau h_{k+1,k} (e_k^T (\lambda I - \tau H_k)^{-1} e_1) v_{k+1}.
\end{align*}
\]

This provides a practical estimate for the true error (4.72) and consequently Hochbruck et al. (1996) uses the following approximation to the error in the Krylov approximation to \(\varphi(\tau J)v\)

\[ 
\varepsilon_k \approx \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) (\beta \tau h_{k+1,k} (e_k^T (\lambda I - \tau H_k)^{-1} e_1) v_{k+1}) d\lambda.
\]

If \(\rho_k\) denotes the above approximation to the exact error \(\varepsilon_k\), then

\[ 
\rho_k = \beta \tau h_{k+1,k} e_k^T \left( \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) (\lambda I - \tau H_k)^{-1} d\lambda \right) e_1 v_{k+1} \\
= \beta \tau h_{k+1,k} e_k^T \varphi(\tau H_k) e_1 v_{k+1} \\
\]

by the Cauchy-Integral formula. Furthermore, we note that \(e_k^T \varphi(\tau H_k) e_1\) is the \((k, 1)\) entry of the matrix \(\varphi(\tau H_k)\) and is a scalar. Unlike the GMRES methods proposed for best rational approximants where the solution of the least squares problem can be avoided until the approximate solution is formed, in order to compute \(\rho_k\), the evaluation of \(\varphi(\tau H_k)\) must be performed. In Algorithm 7 the Krylov subspace approximation to \(\varphi(\tau J)v\) is outlined; with the iterations continued until \(\|\rho_k\|_2\) is less than a specified tolerance \(TOL\).

### 4.3 Implementation issues

The attraction of the Krylov subspace approximant presented in the previous section and the best rational approximants proposed in §4.1.4 lies in the fact that the only operations involving the large matrix \(J\) are matrix-vector products. These products \(Jv_k\) feature in
Algorithm 7: Krylov subspace approximation to \( \varphi(\tau J)v \)

1. **Start**
   Set the error tolerance \( TOL \)

2. **Arnoldi process**
   Compute \( \beta = \|v\|_2 \) and \( v_1 = v/\beta \)
   for \( k = 1 \) until convergence do
     Compute \( w = Jv_k \)
     for \( i = 1, \ldots, k \) do
       \( h_{i,k} = w^T v_i \)
       \( w = w - h_{i,k} v_i \)
     end for
     Compute \( h_{k+1,k} = \|w\|_2 \) and \( v_{k+1} = w/h_{k+1,k} \)
     Find Padé approximation to \( \varphi(\tau H_k) \)
     Compute \( \|\rho_k\|_2 = \|\beta h_{k+1,k} \varphi(\tau H_k) v_{k+1}\|_2 \)
     If \( \|\rho_k\|_2 < TOL \) convergence attained. Go to 3.
     Update \( V_{k+1} = [V_k, v_{k+1}] \)
   end for

3. **Form the approximation**
   Compute \( \varphi(\tau J)v = \beta V_k \varphi(\tau H_k)e_1 \)

the Arnoldi process where \( v_k \) has unit length and since \( J \) is the Jacobian matrix, can be approximated by directional derivatives

\[
Jv_k = J(u_n) v_k \approx \frac{G(u_n + \epsilon v_k) - G(u_n)}{\epsilon},
\]

where \( \epsilon = \sqrt{\epsilon M} \|u_n\|_2 \) with \( \epsilon_M \) equal to the machine epsilon. The error of the approximation (4.75) is proportional to \( \epsilon \) (Knoll & Keyes 2004). In this case, \( J \) need never be formed and the both the exponential Euler and exponential Rosenbrock schemes are “Jacobian-free” (Knoll & Keyes 2004).

In both the exponential Rosenbrock scheme and the method for approximating \( \varphi(\tau J)v \) outlined in Algorithm 6, additional matrix-vector products with the Jacobian are required. The corresponding approximation of (4.75), for a vector \( v_k \) not necessarily of unit length was used. In this case, the perturbation \( \epsilon \) is replaced by \( \sqrt{\epsilon M} \|u_n\|_2 \|v_k\|_2 \).

Note, however that there do exist strategies for approximating the Jacobian depicted in Figure 4.1 that take advantage of the banded structure. Extending an idea for a tridiagonal matrix featured in Kelley (2002), the \( N \) function evaluations associated with the columnwise calculation can be reduced to the bandwidth of the matrix. This was the original approach taken in this thesis, however with the implementation of (4.75) significant decreases in computation time were observed.

The three most promising methods for computing \( \varphi(\tau J)v \) featured in this chapter are those outlined in Algorithms 5, 6 and 7. The following chapter is devoted to conducting numerical experiments to assess the performance of these three techniques, in addition to comparing the exponential integrators discussed in Chapter 3.
Chapter 5

Results

The contents of this chapter begin with the presentation of a numerical solution to the two dimensional problem of unsaturated flow into heterogenous soil stated in §2.2. By applying the exponential integrators of Chapter 3 to the spatially discrete form of Richards’ equation, produced by the finite volume method in Chapter 2, the initial transformed pressure head distribution is advanced in time.

The remaining sections are devoted to numerical experiments; simulations are conducted using the exponential Euler method embedded with best rational and Krylov subspace approximants to the $\varphi$ function. Comparisons are then made against the higher-order exponential Rosenbrock scheme. All comparisons are made based on computational cost, performance in the adaptive time stepping strategy proposed in §3.3 and the accuracy of produced solutions. To gain insight into whether exponential integrators are a viable alternative to implicit stiff integrators, we end the chapter with comparisons with an established stiff integrator, the backward Euler method.

5.1 Numerical solution to Test problem

Recall, the test problem consists of nine alternating blocks of clay and sand with a constant influx of water at a rate of 5 cm/day applied to a 1 m segment of the top block of sand. Figure 5.1 illustrates the saturation ($S$) distribution within the rectangular region after 4, 8 and 12.5 days respectively of continuous application of the water. A constant time step of $\tau = 1000$ (1080 time steps in total) was used with a discretisation of 36 nodes in both the $x$ and $z$ directions (1296 in total). The depicted solution was obtained using the exponential Euler method implemented with a Krylov subspace approximant to the $\varphi$ function, although, the solution behaviour was typical of all methods tested.

The transport of water through the regions of sand is rapid due to its higher saturated conductivity $K_{sat}$ ($6.262 \times 10^{-5}$ ms$^{-1}$ compared with $1.516 \times 10^{-6}$ ms$^{-1}$ for clay) with the water reaching the centre block of clay after 4 days. The flow stagnates when reaching the adjacent blocks of clay, where the water is less easily transported through the void space due to the lower conductivity and fails to deeply penetrate the clay after 8 and 12.5 days. This results in highly saturated regions developing with peak values of about 0.9 after 12.5 days and steep profiles occurring where the saturation changes rapidly over the sand-clay interfaces. The saturation distribution after 12.5 days compared well with that produced by McBride et al. (2006).
Figure 5.1: Saturation profiles after 4, 8 and 12.5 days of continuous application of water to the centre, top block of sand. The saturation of the white region remains unchanged throughout the simulation.
5.2 Numerical experiments

According to Minchev (2004) the greatest unanswered question concerning exponential integrators is whether these methods are fully competitive with existing numerical techniques for solving stiff problems. The author proceeds to claim that the answer requires extensive numerical experiments with fixed and variable step size codes.

In the proceeding sections, the performance of the exponential Euler and exponential Rosenbrock schemes are assessed for the integration of the initial value problem

\[ \frac{du}{dt} = G(u); \quad u(0) = \left( \frac{500}{500 \xi - 1}, \ldots, \frac{500}{500 \xi - 1} \right)^T, \quad (5.1) \]

which was obtained in Chapter 2 using a finite volume spatial discretisation of the test problem. Recall, from Chapter 2, the parameter \( \xi \) controls the nonlinearity of the problem; in all initial simulations the value \( \xi = -4 \text{ m}^{-1} \) was used. The fixed time step values of 10000 s, 5000 s and 1000 s were tested, while for all simulations conducted with adaptive time stepping the time step was intitially set to 1 second and allowed to grow. All simulations were performed in MATLAB (R14) on an Intel Centrino Duo processor running at 2 GHz with 1 GB of RAM.

To gauge the performance of all methods tested, simulations were conducted using the spatial discretisations tabulated in Table 5.1

<table>
<thead>
<tr>
<th>Mesh</th>
<th>( N )</th>
<th>Nodes in ( x ) direction</th>
<th>Nodes in ( z ) direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>144</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Medium</td>
<td>576</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Fine</td>
<td>2304</td>
<td>48</td>
<td>48</td>
</tr>
</tbody>
</table>

Table 5.1: Definition of the coarse, medium and fine spatial discretisations used throughout all numerical experiments. Nodes are divided equally among blocks of clay and sand and are equally spaced within each block. Configuration is chosen to ensure sand-clay interfaces align with control volume faces and nodes are situated on the boundaries of the domain, as depicted in Figure 2.2.

The computational cost associated with each simulation was measured according to the statistics:

\[ \text{CPU time} = \text{time taken to advance the solution to } t = 12.5 \text{ days}, \]

\[ \text{Function evaluations} = \text{number of evaluations of the function } G \text{ featured in (5.1)}. \]

To assess the accuracy of each of the methods, since an exact solution is inaccessible, a saturation distribution \((S_e)\) after 12.5 days was obtained with a mesh consisting of 84 nodes in each of the \( x \) and \( z \) directions \((N = 7506 \text{ nodes in total})\) and a time step size of 10 s \((108000 \text{ time steps in total})\). This solution was produced in 3.4 hours using the exponential Euler method implemented with a Krylov subspace approximant to \( \varphi(\tau J_n)G_n \).

To allow comparisons to be made, MATLAB’s two dimensional interpolant interp2, which uses linear interpolation, was used. The “exact” solution \((S_e)\) was interpolated at
the discrete node points in each of the mesh sizes above for a given computed approximate solution \( S \in \mathbb{R}^N \) whose entries are the discrete nodal saturation values. The error metric

\[
\text{Error} = \frac{1}{\sqrt{N}} \|S_e - S\|_2, \tag{5.2}
\]

was preferred as a measure of the overall accuracy of the computed solutions, rather than the equivalent infinity norm which we found focused solely on the position of the steep wetting front.

In the next section, we begin our experiments by assessing the performance of the exponential Euler integrator.

5.2.1 Exponential Euler method

The exponential Euler method (3.7) has been implemented with the three most promising methods for approximating \( \varphi(\tau J_n)G_n \) identified in Chapter 4; namely

**BRSHIFT:** Best rational approximant with Shifted GMRES as per Algorithm 5.

**BRQUAD:** Best rational approximant with Quadratic GMRES as per Algorithm 6.

**KRYLOV:** Krylov subspace approximant as per Algorithm 7.

The tolerance \( TOL \) for each algorithm was set to \( 10^{-12} \) for all simulations performed. These methods are first compared using a fixed time step size, with relevant statistics tabulated in Table 5.2.

**Fixed step size**

Each of the methods BRSHIFT, BRQUAD and KRYLOV produced solutions that were consistent with that shown in Figure 5.1. An exception occurred when implementing BRSHIFT on the fine mesh with fixed time step values of 10000 and 5000 s. The first simulation failed when attempting to advance the solution from \( t = 80000 \) s to \( t = 90000 \) s while the latter failed when stepping from \( t = 230000 \) s to \( t = 235000 \) s. In both cases, values of the transformed pressure head \( u \) fell outside of the range of the transformation (2.11), which for \( \xi = -4 \) maps negative values of the pressure head \( h \) to the open interval \( u \in (-0.25, 0) \). Reducing the time step to 1000 s rectified this problem. It is interesting to note however, that for simulations implemented with BRQUAD, which is also based on best rational approximants to \( \varphi \), such constraints on the step size were not required.

The first observation from Table 5.2 is the difference in the number of function evaluations between the methods. For each completed simulation both the BRSHIFT and KRYLOV implementations required the fewest evaluations of \( G \). The large discrepancy in the function evaluations for BRQUAD is mainly due to the additional 7 evaluations required to form the approximant, as discussed in §4.3. For this method, the number of evaluations featured exclusively in Algorithm 6 is provided in parentheses. This number provides a measure of the rate of convergence of the underlying iterative process, allowing comparisons between methods to be made.
<table>
<thead>
<tr>
<th>Time step</th>
<th>Method</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>BRSHIFT</td>
<td>2.2</td>
<td>3.0</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>2.6</td>
<td>5.5</td>
<td>30.1</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>2.4</td>
<td>3.1</td>
<td>17.1</td>
</tr>
<tr>
<td>5000</td>
<td>BRSHIFT</td>
<td>5.2</td>
<td>6.0</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>5.6</td>
<td>10.7</td>
<td>45.5</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>5.5</td>
<td>6.0</td>
<td>24.0</td>
</tr>
<tr>
<td>1000</td>
<td>BRSHIFT</td>
<td>28.8</td>
<td>30.0</td>
<td>61.3</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>29.4</td>
<td>41.0</td>
<td>154.8</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>29.6</td>
<td>30.3</td>
<td>63.1</td>
</tr>
</tbody>
</table>

(a) CPU time (s)

<table>
<thead>
<tr>
<th>Time step</th>
<th>Method</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>BRSHIFT</td>
<td>0.0689</td>
<td>0.0594</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>0.0687</td>
<td>0.0584</td>
<td>0.0226</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>0.0687</td>
<td>0.0585</td>
<td>0.0226</td>
</tr>
<tr>
<td>5000</td>
<td>BRSHIFT</td>
<td>0.0658</td>
<td>0.0520</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>0.0657</td>
<td>0.0517</td>
<td>0.0285</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>0.0657</td>
<td>0.0517</td>
<td>0.0285</td>
</tr>
<tr>
<td>1000</td>
<td>BRSHIFT</td>
<td>0.0639</td>
<td>0.0481</td>
<td>0.0351</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>0.0639</td>
<td>0.0481</td>
<td>0.0351</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>0.0639</td>
<td>0.0481</td>
<td>0.0351</td>
</tr>
</tbody>
</table>

(b) Error

<table>
<thead>
<tr>
<th>Time step</th>
<th>Method</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>BRSHIFT</td>
<td>459</td>
<td>710</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>1397(641)</td>
<td>1650(894)</td>
<td>2617(1861)</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>424</td>
<td>628</td>
<td>1393</td>
</tr>
<tr>
<td>5000</td>
<td>BRSHIFT</td>
<td>871</td>
<td>1097</td>
<td>not completed</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>2631(1119)</td>
<td>2974(1462)</td>
<td>4132(2620)</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>815</td>
<td>1023</td>
<td>1991</td>
</tr>
<tr>
<td>1000</td>
<td>BRSHIFT</td>
<td>3461</td>
<td>4345</td>
<td>5155</td>
</tr>
<tr>
<td></td>
<td>BRQUAD</td>
<td>12757(5197)</td>
<td>13040(5480)</td>
<td>14307(6747)</td>
</tr>
<tr>
<td></td>
<td>KRYLOV</td>
<td>3486</td>
<td>4332</td>
<td>5275</td>
</tr>
</tbody>
</table>

(c) Function evaluations

Table 5.2: Simulation statistics associated with the performance of the exponential Euler scheme with fixed time steps of $\tau = 10000, 5000, 1000$ s implemented with the methods BRSHIFT, BRQUAD and KRYLOV for computing the $\varphi$ function. Note for BRQUAD the number provided in parentheses denotes the number of function evaluations used from calls to Algorithm 6.
For each simulation completed by BRSHIFT, the number of function evaluations required in the Arnoldi method was less than that used by BRQUAD. This means Shifted GMRES, as outlined in Algorithm 3, when applied to the shifted linear system

$$(\tau J - \zeta I)x = v,$$

converged faster than the GMRES variant for the quadratic equation

$$(\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2 I)x = \zeta v,$$

as described in Algorithm 4. This finding was also evident for the representative Jacobian matrix investigated in §4.1.5. In terms of computation time the BRSHIFT and KRYLOV methods for computing the $\phi$ function outperformed BRQUAD. While CPU times were similar when the simulations were performed on the coarse mesh, on average for the medium and fine mesh sizes BRSHIFT and KRYLOV were 40% faster than BRQUAD.

For each combination of mesh size and time step, discrepancies in the error and runtime between the most efficient methods BRSHIFT and KRYLOV, were minor. Since the former failed for the largest two fixed time steps tested on the fine mesh we favoured the KRYLOV implementation. This lead to the conclusion that for a fixed time step implementation of the exponential Euler scheme, a Krylov subspace approximant to compute $\phi(\tau J_n)G_n$ is preferred for the test problem.

Next, the three methods are compared in an adaptive time stepping framework.

**Variable step size**

Numerical experiments were conducted using the exponential Euler method in the variable stepsize code outlined in Algorithm 1 with the chosen local error tolerance $tol = 10^{-6}$. Initial investigations proved that this value produced the best overall balance between accuracy and computation time. No benefit was gained for smaller tolerances, with little decrease in the error experienced, although the CPU time and function evaluations increased significantly. The results of all simulations implemented with adaptive time stepping are provided in Table 5.3 with “Mean step” defined as the average time step size throughout the simulation.

It is evident from these results that using adaptive time stepping on both the coarse and medium mesh sizes has produced solutions that improve on the accuracy of the fixed time step solutions. The observed errors of 0.0637 (coarse) and 0.0476 (medium) were less than each of the corresponding values obtained using the fixed time step implementations. Impressively, for the coarse mesh, when compared with the fixed time step of 1000 s, the control on the step size produced a solution of higher accuracy in roughly a third of the time.

For both the medium and fine mesh sizes the average time step recorded was greater than 1000 s. This reduction in the number of time steps compared with the fixed value of $\tau = 1000$, however, is not transferred to the CPU time. The reason for this is the
Table 5.3: Simulation statistics associated with the performance of the exponential Euler scheme with adaptive time stepping \((tol = 10^{-6})\) implemented with the methods BRSHIFT, BRQUAD and KRYLOV for computing the \(\varphi\) function.

<table>
<thead>
<tr>
<th>Method</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time (s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRSHIFT</td>
<td>9.7</td>
<td>59.6</td>
<td>807.2</td>
</tr>
<tr>
<td>BRQUAD</td>
<td>13.8</td>
<td>83.5</td>
<td>882.9</td>
</tr>
<tr>
<td>KRYLOV</td>
<td>9.8</td>
<td>58.1</td>
<td>703.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BRSHIFT</th>
<th>BRQUAD</th>
<th>KRYLOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>0.0637</td>
<td>0.0476</td>
<td>0.0360</td>
</tr>
<tr>
<td>Mean step (s)</td>
<td>3204.7</td>
<td>2084.9</td>
<td>1065.1</td>
</tr>
</tbody>
</table>

The results of this section highlight the additional computational overhead introduced by the adaptive time stepping strategy for the exponential Euler method. The savings by using the recurrence relations discussed in §4.2.1 for computing \(\varphi(\tau J_n)G_n\) from \(\varphi(0.5 J_n)G_n\) are realised for the fine mesh. The BRSHIFT implementation is competitive with KRYLOV when the solution is required on a coarse or medium mesh, however with the increased dimension KRYLOV is approximately 10% faster. These results, in addition to those of the previous section confirm that for both a fixed and variable step size implementation of the exponential Euler scheme, KRYLOV is the preferred choice for approximating \(\varphi(\tau J_n)G_n\), for the chosen test problem.

The performance of the exponential Euler integrator is now assessed against the higher-order exponential Rosenbrock scheme.

### 5.2.2 Exponential Rosenbrock method

Recall from Chapter 3, that the exponential Rosenbrock scheme is fourth order accurate in time compared with the second order accuracy achieved by the exponential Euler scheme. In this section, numerical experiments are conducted to determine whether the higher order accuracy in time transfers to more accurate solutions, or whether larger time steps are exhibited when implemented with stepsize control. All simulations using the exponential Rosenbrock scheme use Krylov subspace approximants to the \(\varphi\) functions and make use of the recurrence relations, as outlined in §4.2.1.

Identical fixed step size numerical experiments to those performed in the previous section, were conducted for the exponential Rosenbrock scheme in addition to simulations performed with adaptive time stepping. All results are provided in Table 5.4. For each mesh size and time step combination, exponential Rosenbrock required more function
evaluations and CPU time when compared with the exponential KRYLOV implementation of the exponential Euler method. This was to be expected however, since the scheme requires, per time step, two additional approximations to the $\varphi$ function, in addition to two additional function evaluations required for computing $d_1$ and $d_7$, both of which require matrix-vector products with the Jacobian. The higher order in time is observed for the fixed time step of 10000 s with the produced solutions using the exponential Rosenbrock method, approximately 5% and 10% more accurate than the corresponding solutions computed using the KRYLOV implementation of exponential Euler.

<table>
<thead>
<tr>
<th>Time step</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>2.9</td>
<td>6.2</td>
<td>34.5</td>
</tr>
<tr>
<td>5000</td>
<td>6.0</td>
<td>10.4</td>
<td>50.9</td>
</tr>
<tr>
<td>1000</td>
<td>31.6</td>
<td>51.9</td>
<td>194.0</td>
</tr>
</tbody>
</table>

(a) Fixed time step - CPU time (s)

<table>
<thead>
<tr>
<th>Time step</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>0.0656</td>
<td>0.0516</td>
<td>0.0316</td>
</tr>
<tr>
<td>5000</td>
<td>0.0644</td>
<td>0.0491</td>
<td>0.0334</td>
</tr>
<tr>
<td>1000</td>
<td>0.0638</td>
<td>0.0478</td>
<td>0.0357</td>
</tr>
</tbody>
</table>

(b) Fixed time step - Error

<table>
<thead>
<tr>
<th>Time step</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1121</td>
<td>1552</td>
<td>2874</td>
</tr>
<tr>
<td>5000</td>
<td>2060</td>
<td>2592</td>
<td>4207</td>
</tr>
<tr>
<td>1000</td>
<td>9834</td>
<td>10549</td>
<td>12941</td>
</tr>
</tbody>
</table>

(c) Fixed time step - Function evaluations

<table>
<thead>
<tr>
<th>Time step</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>13.2</td>
<td>108.3</td>
<td>1495.9</td>
</tr>
<tr>
<td>5000</td>
<td>0.0637</td>
<td>0.0476</td>
<td>0.0360</td>
</tr>
<tr>
<td>1000</td>
<td>3698.6</td>
<td>2037.7</td>
<td>1065.1</td>
</tr>
</tbody>
</table>

(d) Adaptive time stepping

Table 5.4: Simulation statistics associated with the performance of the exponential Rosenbrock scheme with fixed time steps of $\tau = 10000, 5000, 1000$ s and adaptive time stepping ($tol = 10^{-6}$).

With adaptive time stepping, for the coarse mesh, exponential Rosenbrock allowed for, on average, larger values of the time step, although for the finer mesh sizes a reduction in the average time step from the exponential Euler method was observed. With the produced solutions of similar accuracy, the method took 4 s longer to perform the simulation for the coarse mesh and recorded runtimes for the medium and fine mesh sizes twice that observed for the KRYLOV implementation of exponential Euler.

The results of this section indicate that for the initial value problem (5.1) with the parameter $\xi = -4$ there is no benefit obtained in using the higher-order exponential Rosenbrock method over the exponential Euler method for performing the time integration. In the next section, both integrators are compared with an established implicit scheme.

### 5.3 Comparison with an implicit scheme: the backward Euler method

According to Bradie (2006), among the most widely used numerical techniques for approximating the solution of stiff problems are backward differentiation formulas. The simplest example of one such technique is the backward Euler method, which is first order accurate in time (Bradie 2006). The system of ordinary differential equations

$$\frac{d\mathbf{u}}{dt} = \mathbf{G}(\mathbf{u}),$$

from the initial value problem, is evaluated at time level $t = t_{n+1}$ and a first order backward difference approximation to the time derivative is used. This produces a discrete
analogue of (5.3), in the form of a nonlinear system of equations

\[ F(u_{n+1}) = 0, \quad (5.4) \]

where

\[ F(u_{n+1}) = u_{n+1} - u_n - \tau G(u_{n+1}), \quad (5.5) \]

and the system must be solved in order to advance the solution from \( t = t_n \) to \( t = t_{n+1} \).

Our particular implementation of backward Euler used a globally convergent Newton-Krylov method with a simple linesearching strategy as described by Kelley (1987). The preconditioner \( M_J = \text{diag}(J_F) \) produced reasonable results, although convergence of the inner GMRES iterations were in general fast. The nonlinear iterations were terminated according to the criterion

\[ \|F^{(k)}\|_2 < 10^{-8} + 10^{-8} \|F^{(0)}\|_2 \]

with the modified Eisenstat-Walker formula (Eisenstat & Walker 1996) used for determining the forcing term \( \eta_k \) for the stopping criterion of the GMRES inner iterations

\[ \|F^{(k)} + J_F \delta u^{(k)}\|_2 < \eta_k \|F^{(k)}\|_2. \]

A Jacobian-free strategy (Knoll & Keyes 2004) was implemented for approximating the matrix-vector products involving \( J_F \) required in the Arnoldi process, in a manner identical to that outlined in §4.2.3. A simple heuristic-based time stepping strategy was used to adaptively change \( \tau \) throughout the simulation, whereby the convergence rate of the Newton iterations, a measure of the nonlinearity of the problem, determines the size of the time step to be used. An approach similar to the one described by D’Haese et al. (2007) was implemented and is detailed below.

At the completion of the nonlinear iterations if convergence was not achieved in fewer than 15 iterations, or the solution fell outside of the range \((1/\xi, 0)\) of the transformation, the time step was halved and the time step repeated. The time step was increased by a factor of 1.1 provided the result remained below 5000 s and convergence of the nonlinear iterations was fast. The criteria here was that the iterations never exceeded 4 for 10 consecutive time steps.

The relationship between the Jacobian of \( F \) and \( G \) is evident from the form of the nonlinear equation,

\[ J_F = I - \tau J, \quad (5.6) \]

where \( J \) is the Jacobian matrix of \( G \). In fact, for symmetric \( J \), Krylov subspace approximations to the exponential \( e^{\tau J} \mathbf{v} \) have been shown by Druskin & Knizhnerman (1991) to converge considerably faster than those for the shifted linear system \((I - \tau J) \delta u^{(k)} = -F^{(k)}, \)
whose solution is the Newton step required in the backward Euler scheme. This suggests exponential integrators may outperform this implicit scheme.

In addition to the error metric (5.2) we have included a second measure, based on the mass-balance error. This measures how well the total volume of water in the solution domain balances the net inflow of water at the boundaries and the initial amount of water in the domain. To determine this error we consider the governing partial differential equation

\[ \frac{\partial \theta}{\partial t} + \nabla \cdot Q = 0. \] (5.7)

Integrating the left hand side of (5.7) over the entire solution domain gives that

\[ \int_0^3 \int_0^5 \left( \frac{\partial \theta}{\partial t} + \nabla \cdot Q \right) \, dx \, dz = 0. \] (5.8)

Applying the Divergence theorem, we arrive at

\[ \text{MBE}(t_n) = \frac{d}{dt} \int_0^3 \int_0^5 \theta \, dx \, dz + \sum_{j=1}^{4} (Q \cdot n)_{b_j}. \] (5.9)

where \( b_j \) denotes one of the four boundaries of the solution domain. Due to the boundary conditions of the test problem and using a forward difference approximation to the derivative we obtain

\[ \text{MBE}(t_n) \approx \frac{1}{t_{n+1} - t_n} \left( \int_0^3 \int_0^5 \theta_{n+1} \, dx \, dz - \int_0^3 \int_0^5 \theta_n \, dx \, dz \right) - 5.787 \times 10^{-7}, \] (5.10)

which is an approximation to the mass-balance error incurred by advancing the solution from \( t_n \) to \( t_{n+1} \). The integrals in equation (5.10) are approximated using the Trapezoidal Rule. Note that since the global error associated with the Trapezoidal rule is second order and the finite volume method is at best second order in space, little benefit is gained by using a higher order quadrature method. We define the error metric as the sum of the mass-balance errors in absolute value:

\[ \text{Accumulated MBE} = \sum_{0 \leq t_n \leq 12.5} |\text{MBE}(t_n)|. \]

The exponential Euler and exponential Rosenbrock integrators have been compared to the backward Euler method implemented with adaptive time stepping on the coarse mesh size. Resulting statistics are tabulated in Table 5.5. The number of failed time steps refers to the case when the time step is rejected \((err > tol)\) for the exponential integrator, while for the backward Euler method a failed step is defined when the nonlinear iterations failed to converge within the allocated 15 iterations for a given time step \( \tau \) and the time step was halved.

Both exponential integrators clearly outperformed the backward Euler method on the test problem. Figure 5.2 illustrates both exponential Euler and exponential Rosenbrock allowed for time steps, that were two orders of magnitude greater than the backward Euler.
Exponential Euler  |  Exponential Rosenbrock  |  Backward Euler
---|---|---
CPU time (s)  |  9.8  |  13.9  |  1496.9  
Total time steps  |  335  |  314  |  19879  
Mean step (s)  |  3223.9  |  3439.5  |  54.82  
Total failed steps  |  14  |  2  |  242  
Linesearches  |  n/a  |  n/a  |  1283  
Function evaluations  |  5948  |  10823  |  195656  
Error  |  0.0637  |  0.0637  |  0.0637  
Accumulated MBE  |  $1.1881 \times 10^{-4}$  |  $8.3579 \times 10^{-5}$  |  $1.0994 \times 10^{-2}$  

Table 5.5: Simulation statistics associated with the performance of the exponential Euler and exponential Rosenbrock schemes implemented with Krylov subspace approximants to the $\phi$ function and the backward Euler scheme on the test problem stated in §2.2 with $\xi = -4 \text{ m}^{-1}$.

time step on average. Despite the reduced time step, backward Euler failed to produce more accurate solutions, with each integrator recording identical values of 0.0637 for the error metric. With the increased number of time steps performed by backward Euler, the mass-balance error incurred is larger, as would be expected.

![Figure 5.2](image)

Figure 5.2: Evolution of the time step of the exponential Euler, exponential Rosenbrock and backward Euler schemes for the test problem outlined in §2.2 with $\xi = -4$.

Setting the transformed parameter $\xi$ to be zero, the untransformed highly nonlinear $h$ based form of Richards’ equation is obtained. The results of these simulations are provided in Table 5.6.

This marks the transition from a mildly stiff problem to a stiff one, where integrating the initial value problem in time becomes more challenging. The implicit integrator constrains the time-step size to an impractical value of $10^{-3}$, in order for the fraction of the Newton step to be greater than $10^{-4}$. Figure 5.3 illustrates the evolution of the time step for the untransformed equation for both exponential integrator methods. Both methods are notably less affected with the size of each time step still allowing for reasonably accurate solutions to be obtained in a realistic timeframe. An increase in the error and accumulated mass-balance errors is observed however, indicating that a smaller value of the parameter $tol$ in Algorithm 1 may be required.
Figure 5.3: Evolution of the time step of the exponential Euler and exponential Rosenbrock schemes for
the test problem outlined in §2.2 with $\xi = 0$

<table>
<thead>
<tr>
<th></th>
<th>Exponential Euler</th>
<th>Exponential Rosenbrock</th>
<th>Backward Euler</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time (s)</td>
<td>476.9</td>
<td>721.2</td>
<td>Not completed</td>
</tr>
<tr>
<td>Total time steps</td>
<td>10138</td>
<td>12121</td>
<td>-</td>
</tr>
<tr>
<td>Mean step (s)</td>
<td>106.5</td>
<td>89.1</td>
<td>$O(10^{-3})$</td>
</tr>
<tr>
<td>Total failed steps</td>
<td>9</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Function evaluations</td>
<td>190637</td>
<td>444732</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>0.1024</td>
<td>0.1024</td>
<td>-</td>
</tr>
<tr>
<td>Accumulated MBE</td>
<td>$5.5265 \times 10^{-3}$</td>
<td>$6.6124 \times 10^{-3}$</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.6: Simulation statistics associated with the performance of the exponential Euler and exponential
Rosenbrock schemes implemented with Krylov subspace approximants to the $\varphi$ function and
the backward Euler scheme on the test problem stated in §2.2 with $\xi = 0$.

Each of these simulations clearly illustrate that both exponential integrators outperform
the implicit integrator. Although, in some instances the exponential Rosenbrock method
allows for large time steps, the KRYLOV implementation of the exponential Euler scheme
still outperformed the higher-order method in terms of CPU time and the number of
function evaluations. Converting the problem from mildly stiff ($\xi = -4$) to stiff ($\xi = 0$),
exponential Euler not only produced the solution in less CPU time and fewer function
evaluations, but allowed for a time step 50% greater than that of exponential Rosenbrock.
All the results in this chapter indicate, that for the test problem the exponential Euler
method is the preferred time integration option.
6.1 Summary

In an attempt to determine the suitability of exponential integrators for the time integration of spatially discrete forms of Richards’ equation, a two-dimensional problem featured in the literature was discretised using the finite volume method. The exponential Euler and exponential Rosenbrock schemes were implemented for integrating the resulting ODE system in time and some large, sparse strategies for the evaluation of the matrix function \( \varphi \) were reviewed. The first technique was concerned with best rational approximants to \( \varphi \), while the second involved projecting the large matrix onto a small Krylov subspace and performing the evaluation on a matrix of much smaller dimension.

Numerical experiments were conducted for various spatial and temporal increments using both fixed and variable step size strategies. The experiments indicated that the exponential Euler scheme implemented with a Krylov subspace approximant to the \( \varphi \) function appears to be the best option. This choice of integrator consistently outperformed the exponential Rosenbrock scheme, while in addition the Krylov subspace approximant outperformed the best rational approximants for approximating the \( \varphi \) function. Comparisons with a classical implicit stiff integrator produced results to support the claim that the exponential Euler is a promising explicit alternative to established implicit methods.

In the context of Richards’ equation, based on the results of the previous chapter, the exponential Euler method appears to be well suited to the time integration of spatially discrete forms of the equation. To compete with the versatility of implicit schemes, these methods must be extended to Differential Algebraic Equations (DAE). This is to allow the simulation of variably saturated problems, where the pressure head \( (h) \) becomes nonnegative. In this case the specific water capacity \( C(h) \) is zero and the time derivative vanishes producing an algebraic equation. While, the exponential Rosenbrock method has been extended to deal with DAEs by Hochbruck et al. (1996), it is unclear whether the exponential Euler method is as easily modified.

In the remaining sections, the main achievements of this thesis are summarised and suggestions for future research given.
6.2 Contributions

In this work, we demonstrated how best rational approximants, often reserved for small matrices, can be used for large, sparse evaluations of the $\varphi$ function. This process involved deriving GMRES-like strategies for the solution of the shifted systems of equations

$$(\tau J - \zeta I)x = v,$$

that arise in the approximants. In addition, it was revealed how costly complex arithmetic can be avoided by selection of the Krylov search space. It was determined by equating real and imaginary components of the shifted system, that the solution could be obtained by working completely within the real number system. This lead to the following quadratic matrix equation for the imaginary component of the solution $x$:

$$(\tau^2 J^2 - 2\zeta \tau J + |\zeta|^2 I)x_r = \zeta \Im v.$$

Seeking an approximate solution to this equation in the affine space $x_r^{(0)} + K_k(J, v)$ we showed that for each $k$ the approximate solution $x_r^{(k)}$ which uniquely minimised the residual two norm is given by $x_r^{(0)} + V_k y_k^q$ where the columns of $V_k$ formed an orthonormal basis for the subspace $K_k(J, v)$ and $y_k^q$ was the least squares solution of a small $(k+2) \times k$ rectangular system of equations. Furthermore, it was demonstrated that by applying two sequences of Givens’ rotations, the residual two norm could be extracted at each stage of the solution process without forming $x_r^{(k)}$ or solving for $y_k^q$. Details were then provided on how the process can be restarted.

Provided convergence was fast we suggested altering the search space to $K_k(J, v)$. This allowed each of the solutions to be obtained with a single Arnoldi basis construction and required only one seventh the number of matrix-vector products. The Shifted GMRES variant implemented in this way were faster than start of the art Krylov subspace methods when used with a fixed time step.

6.3 Future Work

The exponential Euler method appears computationally very promising for the integration of discretised partial differential equations. Further research opportunities in this area lie in the construction and improvement of techniques for approximating $\varphi(\tau J)v$, required in the time-stepping formula.

Research into the accuracy of lower degree best rational approximants, implemented with the methods BRSHIFT and BRQUAD may prove to be beneficial. This decreases the number of shifted linear systems that must be solved per computation of $\varphi(\tau J)v$ and may result in methods which are more competitive than the $(14,14)$ rational approximant.

Further savings could be realised with investigations into the effect of the pole $\zeta$ on the convergence rate. Our observations indicated that for the different values of $\zeta$ differences in the dimension of the subspace $K_k(J, v)$ required for convergence were rare. This suggests that the stopping criterion for Algorithms 5 and 6 could be based purely on the
residual two norm of a single pole. This would reduce unnecessary application of Givens’ rotations required in Algorithms 5 and 6 for each $\zeta$-dependent least squares problem. Finally, research into techniques for preconditioning the iterations of Shifted and Quadratic GMRES as discussed in Algorithms 3 and 4 would prove invaluable.

To further assess the applicability of best rational approximants, numerical experiments could be conducted to determine when the assumption that $J$ be negative semidefinite breaks down. For problems where the eigenvalues of the Jacobian matrix have large imaginary components, research into best rational approximants in regions of the complex plane are needed. We are, however, unaware whether such approximants exist.

For the Krylov subspace approximation to $\varphi(\tau J)v$, the error estimate for stopping the process could be improved upon. This would likely entail performing the integration along a suitably defined contour in the complex plane, rather than replacing the integrand with the residual of an appropriate FOM process. Our experiments suggested that using the Padé approximant to compute $\varphi(\tau H_k)$ was inefficient for small values of $k$, where an eigenvalue decomposition of $\tau H_k$ may suffice.

As previously mentioned, extending the exponential Euler method to a DAE setting for Richards’ equation is important. In our preliminary experiments, setting $C(h) = 10^{-10}$ for $h \geq 0$ produced results for a two dimensional variably saturated flow problem which compared well with those featured in McBride et al. (2006). This may provide an avenue for applying the exponential Euler scheme for variably saturated Richards’ equation problems, although further experimental evidence is required.

The adaptive time stepping strategy for the exponential Euler scheme presented in this thesis is not ideal because too much additional computation is introduced into the scheme to estimate the local error. Future research could involve the construction of a suitable higher-order exponential integrator, which re-uses computation already performed in the exponential Euler scheme.

Finally, we plan to pursue an idea of a backward-exponential-Euler embedded method, where the exponential integrator is used to gauge the error of the implicit scheme. Our initial considerations have indicated that minimal computational overhead is introduced by using the exponential Euler time-stepping formula at each step in the implicit scheme.


