Anderson acceleration applied to incompressible Navier-Stokes equations

J.M. Dankelman

TU Delft

Anderson acceleration applied to incompressible Navier-Stokes equations

by

J.M. Dankelman

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on June 20, 2025.

Student number:	5522919	
Project duration:	April 22, 2025 – June 20, 2025	
Thesis committee:	Dr. D. Toshniwal	TU Delft, supervisor
	K. W. Dijkstra, MSc.	TU Delft, supervisor
	Dr. H. N. Kekkonen	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Cover image: (Kichigin, 2024).

TU Delft

Laymen's summary

The Navier-Stokes equations describe the flow of fluids and thus have a lot of applications, for instance in modelling the weather and oceans, and in designing aircraft and cars. It is therefore important that numerical solutions can be found efficiently. There are methods that can be used when numerically solving the equations, namely the Newton-Raphson method and the Picard iteration, but these methods can fail when the fluid has certain properties. Anderson acceleration is a numerical method that can improve the rate at which the Picard iteration converges towards the solution. In this paper, Anderson acceleration is first implemented for problems in one dimension. It is then extended to problems in multiple dimensions and applied to the Navier-Stokes equations. Its performance is compared to that of the Newton-Raphson method and the Picard iteration. From this, it follows that Anderson acceleration can indeed improve the convergence rate of Picard. Furthermore, the method can find solutions in situations where the Newton-Raphson method and the Picard iteration do not.

Summary

The Navier-Stokes equations describe the flow of fluids and thus have a lot of applications, for instance in modelling the weather and oceans, and in designing aircraft and cars. It is therefore important that numerical solutions can be found efficiently. The equations are given by

$$(\vec{u}(\vec{x},t)\cdot\nabla)\vec{u}(\vec{x},t) - \mathrm{Re}^{-1}\Delta\vec{u}(\vec{x},t) + \nabla p(\vec{x},t) = \vec{f}(\vec{x},t), \tag{1a}$$

$$\nabla\cdot\vec{u}(\vec{x},t) = 0, \tag{1b}$$

where the velocity $\vec{u}(\vec{x},t)$ and the pressure $p(\vec{x},t)$ are unknown and $\vec{f}(\vec{x},t)$ represents external forces. Equation (1b) states that the fluid is incompressible. The Reynolds number, Re, indicates how viscous a fluid is. Relatively viscous fluids have low Reynolds numbers and fluids that flow more easily have high Reynolds numbers.

Because the term $(\vec{u}(\vec{x},t)\cdot\nabla)\vec{u}(\vec{x},t)$ is non-linear, the equations have to be linearised in order to solve them. This can be done with the Newton-Raphson method and the Picard iteration, but these methods can fail when the Reynolds number gets large. Anderson acceleration is a method that can improve the rate at which the Picard iteration converges. This method uses the $m+1$ previous approximations when calculating a new approximation. The variable m is called the depth of the acceleration.

In this paper, Anderson acceleration is first implemented for scalar-valued and vector-valued problems. It is then applied to a benchmark problem given by the incompressible Navier-Stokes equations. A Nutils implementation of the finite element method is used with different values of the number of elements and degree of the basis functions. The performance of Anderson is compared to that of Newton's method and the Picard iteration.

From this it follows that Anderson acceleration can indeed improve the convergence rate of Picard. Furthermore, the method can converge in situations where Newton's method and the Picard iteration do not, which is the case for high Reynolds numbers. Taking a depth of 3 or 4 results in better convergence rates than a depth of 1 or 2, and in turn less computational time. Higher depths also make the method more stable. However, for low Reynolds numbers, Newton's method is a faster choice. Even though this method takes more time per iteration, it has better convergence rates and thus needs less iterations and takes less time in total.

Contents

1

Introduction

The Navier-Stokes equations describe the flow of fluids and thus have a lot of applications, for instance in modelling the weather and oceans, and in designing aircraft and cars (Younsi, 2012). The equations are given by

$$(\vec{u}(\vec{x},t)\cdot\nabla)\vec{u}(\vec{x},t) - \mathrm{Re}^{-1}\Delta\vec{u}(\vec{x},t) + \nabla p(\vec{x},t) = \vec{f}(\vec{x},t), \tag{1.1a}$$

$$\nabla\cdot\vec{u}(\vec{x},t) = 0, \tag{1.1b}$$

where the velocity $\vec{u}(\vec{x},t)$ and the pressure $p(\vec{x},t)$ are unknown. The term $\vec{f}(\vec{x},t)$ represents external forces acting on the fluid (Łukaszewicz and Kalita, 2016).

Equation (1.1b) states that the fluid is incompressible. This means that the density of the fluid does not change along the trajectories caused by the velocity $\vec{u}$. In other words, the density of any small region stays constant when this region moves over time. Water is an example of an incompressible fluid (Boyer and Fabrie, 2013).

The Reynolds number, Re, indicates how viscous or 'sticky' the fluid is. This number represents the ratio between the convective term $(\vec{u}(\vec{x},t)\cdot\nabla)\vec{u}(\vec{x},t)$, and the diffusive term $\Delta\vec{u}(\vec{x},t)$ (Veldman, 2012). Fluids with low Reynolds numbers are relatively viscous and therefore very resistant to flow. On the other hand, fluids with high Reynolds numbers have relatively low viscosity and flow much easier. Flows associated with low Reynolds numbers are called laminar and those with high Reynolds numbers are turbulent (Boyer and Fabrie, 2013).

In this paper, the flow inside a two-dimensional lid-driven square cavity is considered. The domain is shown in Figure 1.1. Three of the boundaries are solid and the velocity on these boundaries is zero, which is the no-slip boundary condition. On the fourth, non-solid boundary there is no outflow, but there is some horizontal velocity. Because of this velocity, the fluid inside of the domain will move in a circular motion that is dependent on the Reynolds number. This problem is a good benchmark that is widely used and imitates real-world applications (Kamel et al., 2020).

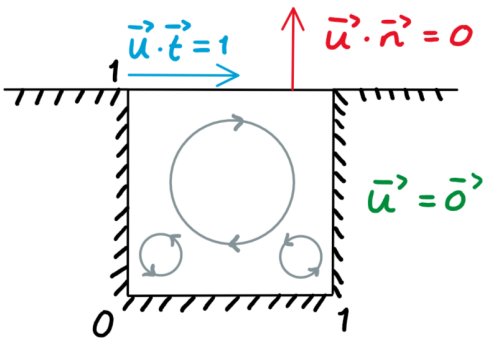

Figure 1.1: Domain with boundary conditions.

Since the convective term in the equations is non-linear, the equations have to be linearised in order to solve them. This can be done with the Picard iteration or the Newton-Raphson method. However, these root-finding methods can fail when the Reynolds number is large (Pollock et al., 2019). Therefore, the aim of this research is to implement Anderson acceleration, which is an algorithm that could improve the convergence of the Picard iteration, on the benchmark problem and compare its performance to the other methods for different parameters.

In order to do this, the methods and the types of problems that they can be used for are first examined. The algorithms are then implemented for scalar-valued problems and extended to vector-valued problems and the incompressible Navier-Stokes equations. The convergence rates are examined, as well as the actual solutions and the timings of the methods.

The structure of this paper is as follows. Firstly, in Chapter 2, the algorithms are introduced. Their performance for scalar-valued problems and vector-valued problems is discussed in Chapter 3. In Chapter 4, the methods are implemented for the Navier-Stokes equations with several different parameters, and their performance is again analysed. Finally, Chapter 5 contains the conclusion and discussion.

2

Root-finding algorithms

Roots of functions $\vec{f} : \mathbb{R}^n \to \mathbb{R}^n$ are points satisfying the system

$$\vec{f}(\vec{x}) = \vec{0}. \tag{2.1}$$

Root-finding problems with this form can be rewritten as fixed-point problems with the form

$$\vec{x} = \vec{G}(\vec{x}) \tag{2.2}$$

in multiple ways, for instance by adding $\vec{x}$ to both sides of Equation (2.1).

When $\vec{f}$ and $\vec{G}$ contain non-linear terms, these problems can become difficult to solve. To remedy this, parts of the equation that depend non-linearly on the solution $\vec{x}$ can be changed to be linearly dependent of $\vec{x}$, which results in a linear system (Van Kan et al., 2014).

This linearisation has to be done for the Navier-Stokes equations, since they contain the non-linear term $(\vec{u}(\vec{x},t) \cdot \nabla)\vec{u}(\vec{x},t)$. This can be achieved with different iterative methods, three of which are discussed in this chapter. Firstly, in Section 2.1, the Newton-Raphson method is treated. This is followed by the Picard iteration in Section 2.2. The last method, Anderson acceleration, is discussed in Section 2.3.

2.1. Newton-Raphson

The Newton-Raphson method, or just Newton's method, can be used to solve root-finding problems of the form (2.1). It is derived from first order Taylor expansions of the components of a function $\vec{f} : \mathbb{R}^n \to \mathbb{R}^n$ in root $\vec{x}_*$ around a point $\vec{x}$, which are given by

$$\begin{aligned} f_{(1)}(\vec{x}_*) &= f_{(1)}(\vec{x}) + \sum_{i=1}^{n} (x_{*(i)} - x_{(i)}) \frac{\partial f_{(1)}}{\partial x_{(i)}}(\vec{x}) + \mathcal{O}(||\vec{x}_* - \vec{x}||_2^2), \\ &\vdots \\ f_{(n)}(\vec{x}_*) &= f_{(n)}(\vec{x}) + \sum_{i=1}^{n} (x_{*(i)} - x_{(i)}) \frac{\partial f_{(n)}}{\partial x_{(i)}}(\vec{x}) + \mathcal{O}(||\vec{x}_* - \vec{x}||_2^2). \end{aligned} \tag{2.3}$$

Using that $\vec{x}_*$ is a root, and therefore $\vec{f}(\vec{x}_*) = \vec{0}$, and neglecting the second order term results into

$$\begin{aligned} \vec{f}_{(1)}(\vec{x}) + \sum_{i=1}^{n} (x_{*(i)} - x_{(i)}) \frac{\partial f_{(1)}}{\partial x_{(i)}}(\vec{x}) &= 0, \\ &\vdots \\ \vec{f}_{(n)}(\vec{x}) + \sum_{i=1}^{n} (x_{*(i)} - x_{(i)}) \frac{\partial f_{(n)}}{\partial x_{(i)}}(\vec{x}) &= 0. \end{aligned} \tag{2.4}$$

These equations can be written in the matrix-vector form

$$J(\vec{x})(\vec{x}_* - \vec{x}) = -\vec{f}(\vec{x}), \tag{2.5}$$

where $J(\vec{x})$ denotes the Jacobian matrix of $\vec{f}$ evaluated at $\vec{x}$, given by

$$J(\vec{x}) = \begin{pmatrix} \frac{\partial f_{(1)}}{\partial x_{(1)}} & \cdots & \frac{\partial f_{(1)}}{\partial x_{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{(n)}}{\partial x_{(1)}} & \cdots & \frac{\partial f_{(n)}}{\partial x_{(n)}} \end{pmatrix}. \tag{2.6}$$

Equation (2.5) is iteratively used in Newton's method, which is described in Algorithm 1 (Van Kan et al., 2014). Note that the equation that is solved in each iteration is linear, since the Jacobian matrix and the function value are evaluated in the known point $\vec{x}_k$.

Algorithm 1: Newton-Raphson method

Choose $\vec{x}_0$.
for $k = 0, 1, 2, \ldots$ **do**
  Solve $J(\vec{x}_k)\vec{c}_k = -\vec{f}(\vec{x}_k)$.
  Set $\vec{x}_{k+1} = \vec{x}_k + \vec{c}_k$.
end

When $n = 1$, $J(x)$ is equal to $f'(x)$, so Equation (2.5) becomes

$$f'(x)(x_* - x) = -f(x) \Leftrightarrow x_* = x - \frac{f(x)}{f'(x)}. \tag{2.7}$$

In this one dimensional case, the method converges towards the root x_* when the starting point x_0 is close enough to the root (Dennis and Schnabel, 1996). The convergence is quadratic when $f'(x_*) \neq 0$, meaning

$$\exists M > 0 : |x_k - x_*| \leq M|x_{k-1} - x_*|^2 \Leftrightarrow \frac{|x_k - x_*|}{|x_{k-1} - x_*|^2} \leq M. \tag{2.8}$$

If $f'(x_*) = 0$, the multiplicity of x_* is bigger than one, so $(x - x_*)$ can be factored out of f multiple times. x_* is then called a multiple root. In this case, the method converges only linearly, so

$$\exists M > 0 : |x_k - x_*| \leq M|x_{k-1} - x_*| \Leftrightarrow \frac{|x_k - x_*|}{|x_{k-1} - x_*|} \leq M. \tag{2.9}$$

In higher dimensions, $J(\vec{x}_*)^{-1}$ has to exist in order for the method to converge. This convergence is quadratic, which means

$$\exists M > 0 : ||x_k - x_*||_2 \leq M||x_{k-1} - x_*||_2^2 \Leftrightarrow \frac{||x_k - x_*||_2}{||x_{k-1} - x_*||_2^2} \leq M. \tag{2.10}$$

The method can nicely be made visual in one dimension. This is done in Figure 2.1, which shows two iterations of Newton's method with starting point x_0 for a function f with root x_*. For each iteration k, the point x_{k+1} is set to be the root of the tangent line of f in x_k.

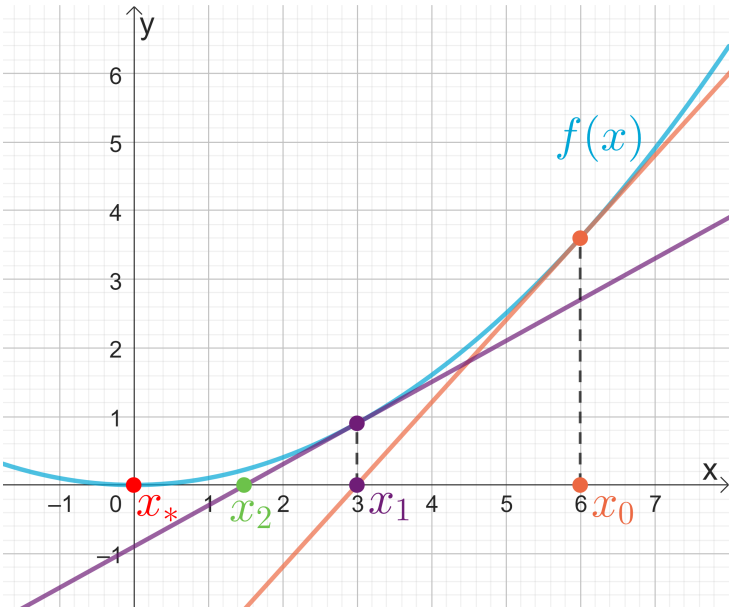

Figure 2.1: Two iterations of the Newton-Raphson method with starting point $x_0 = 6$ for $f(x) = 0.1x^2$ with root $x_* = 0$.

2.2. Picard

Solutions of problems of the form (2.2) can be approximated with the Picard iteration, which is also called the fixed-point iteration. This method is given by an iterative process where the next approximation is set to be the function value in the previous approximation, as can be seen in Algorithm 2 (Van Kan et al., 2014).

Algorithm 2: Picard iteration

Choose $\vec{x}_0$.
for $k = 0, 1, 2, \ldots$ **do**
  Set $\vec{x}_{k+1} = \vec{G}(\vec{x}_k)$.
end

By Banach's fixed-point theorem on a set $X \subseteq \mathbb{R}^n$, the Picard iteration converges to a unique solution $\vec{x}_* \in X$ for any starting point $\vec{x}_0 \in X$ when $\vec{G}$ is an a-contraction, meaning

$$\exists a \in [0, 1) \; \forall \vec{x}, \vec{y} \in X : ||\vec{G}(\vec{x}) - \vec{G}(\vec{y})||_2 \leq a||\vec{x} - \vec{y}||_2. \tag{2.11}$$

Furthermore, the rate of convergence is given by

$$||\vec{x}_k - \vec{x}_*||_2 \leq a||\vec{x}_{k-1} - \vec{x}_*||_2 \Leftrightarrow \frac{||\vec{x}_k - \vec{x}_*||_2}{||\vec{x}_{k-1} - \vec{x}_*||_2} \leq a \tag{2.12}$$

and is thus linear (Berinde, 2007).

In Figure 2.2, two iterations of the Picard iteration with starting point x_0 for a one-dimensional G with fixed-point x_* are shown visually.

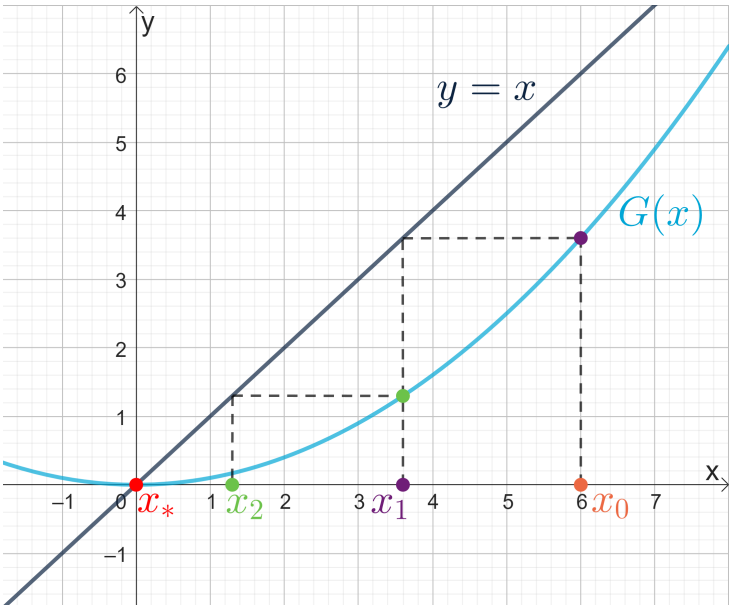

Figure 2.2: Two iterations of the Picard iteration with starting point $x_0 = 6$ for $G(x) = 0.1x^2$ with fixed-point $x_* = 0$.

2.3. Anderson acceleration

Anderson acceleration is a method developed by Anderson (1965) in order to improve the convergence of the Picard iteration. Whereas the Picard iteration only takes the previous approximation into account when calculating the next approximation, Anderson acceleration also uses the m iterations before that. The new approximation $\vec{x}_{k+1}$ is chosen as a linear combination of the function values in the $m+1$ previous approximations. The method is described in Algorithm 3 (Walker and Ni, 2011).

Algorithm 3: Constrained Anderson acceleration with depth $m \geq 1$

Choose $\vec{x}_0$.
Set $\vec{x}_1 = \vec{G}(\vec{x}_0)$.
for $k = 1, 2, \ldots$ **do**
  Set $m_k = \min\{k, m\}$.
  Set $F_k = (\vec{f}_{k-m_k}, \ldots, \vec{f}_k)$ where $\vec{f}_i = \vec{G}(\vec{x}_i) - \vec{x}_i$.
  Set $G_k = (\vec{G}(\vec{x}_{k-m_k}), \ldots, \vec{G}(\vec{x}_k))$.
  Find $\vec{\alpha}^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)})^T$ that solves

$$\min_{\vec{\alpha}=(\alpha_0,\ldots,\alpha_{m_k})^T} ||F_k\vec{\alpha}||_2 \text{ s.t. } \sum_{i=0}^{m_k} \alpha_i = 1.$$

  Set $\vec{x}_{k+1} = G_k\vec{\alpha}^{(k)}$.
end

In the constrained optimisation step, a linear combination of the residuals $\vec{f}_i$ is found with the smallest norm. This step can be written as an unconstrained least-squares problem, given in Algorithm 4 (Walker and Ni, 2011), by setting

$$\begin{aligned} \alpha_0 &= \gamma_0, \\ \alpha_i &= \gamma_i - \gamma_{i-1} \text{ for } i = 1, \ldots, m_k - 1, \\ \alpha_{m_k} &= 1 - \gamma_{m_k-1}. \end{aligned} \tag{2.13}$$

The size of the least-squares problem is $n \times m_k$, where n is the number of unknowns. An implementation of this second algorithm is used in the following chapters.

Algorithm 4: Unconstrained Anderson acceleration with depth $m \geq 1$

Choose $\vec{x}_0$.
Set $\vec{x}_1 = \vec{G}(\vec{x}_0)$.
for $k = 1, 2, \ldots$ **do**
  Set $m_k = \min\{k, m\}$.
  Set $F_k = (\vec{f}_{k-m_k}, \ldots, \vec{f}_k)$ where $\vec{f}_i = \vec{G}(\vec{x}_i) - \vec{x}_i$.
  Set $\mathcal{F}_k = (\Delta\vec{f}_{k-m_k}, \ldots, \Delta\vec{f}_{k-1})$ where $\Delta\vec{f}_i = \vec{f}_{i+1} - \vec{f}_i$.
  Set $\mathcal{X}_k = (\Delta\vec{x}_{k-m_k}, \ldots, \Delta\vec{x}_{k-1})$ where $\Delta\vec{x}_i = \vec{x}_{i+1} - \vec{x}_i$.
  Find $\vec{\gamma}^{(k)} = (\gamma_0^{(k)}, \ldots, \gamma_{m_k-1}^{(k)})^T$ that solves

$$\min_{\vec{\gamma}=(\gamma_0,\ldots,\gamma_{m_k-1})^T} ||\vec{f}_k - \mathcal{F}_k\vec{\gamma}||_2.$$

  Set $\vec{x}_{k+1} = \vec{x}_k + \vec{f}_k - (\mathcal{X}_k + \mathcal{F}_k)\vec{\gamma}^{(k)} = \vec{G}(\vec{x}_k) - (\mathcal{X}_k + \mathcal{F}_k)\vec{\gamma}^{(k)}$.
end

3

Convergence of root-finding algorithms

In this chapter, the convergence of the methods discussed in Chapter 2 is examined numerically by testing different root-finding and fixed-point problems with Python implementations of the methods. The codes can be found in https://gitlab.tudelft.nl/kwdijkstra/bsc-jasmijn.

In the implementations, the iterations are stopped when the residual, $||\vec{f}(\vec{x}_k)||_2$ for Newton or $||\vec{G}(\vec{x}_k) - \vec{x}_k||_2$ for Picard and Anderson, is smaller than tolerance $\delta = 1\text{e}{-10}$. This is done because then the approximation is considered to be quite accurate. For practical reasons, the iterations are also stopped when:

- the difference between successive approximations, $||\vec{x}_{k+1} - \vec{x}_k||_2$, is smaller than $\epsilon = 1\text{e}{-5}$, since more iterations will not significantly improve the approximation;
- k exceeds $N = 100$, in order to prevent long running times when the method does not converge or converges very slowly;
- $||\vec{x}_k||_2$ exceeds $L = 1\text{e}{+100}$, to prevent overflow errors when the method diverges.

Sections 3.1 to 3.4 treat various scalar problems. In Section 3.1, the three methods are applied to a linear problem. Because Banach's theorem gives information about the convergence of the Picard iteration for contractions, in Section 3.2 a contraction is considered. Since Newton's method theoretically converges linearly for multiple roots (satisfying $f'(x_*) = 0$), a function with a multiple root is discussed in Section 3.3. In Section 3.4, two problems for which Newton's method does not converged are examined. Then, in Section 3.5, the methods are applied to vector-valued problems. Finally, in Section 3.6, the findings are summarised and the methods are compared to each other.

3.1. Convergence for linear problems

The first test problem is the linear root-finding problem

$$f(x) = ax + b = 0 \text{ with } a, b \in \mathbb{R},\ a \neq 0. \tag{3.1}$$

The analytical solution of this problem is $x_* = -\frac{b}{a}$. The approximations of the Newton-Raphson method are given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{ax_k + b}{a} = -\frac{b}{a}. \tag{3.2}$$

This means that the exact solution of linear problems should be found after one iteration of Newton's method, regardless of the starting point.

By isolating x in Equation (3.1), it is found that the linear root-finding problem is equivalent to the fixed-point problem

$$x = G(x) = -\frac{b}{a}. \tag{3.3}$$

Note, however, that there are multiple ways to convert a root-finding problem into a fixed-point problem. In Section 3.4.1, it is studied how different choices may affect the convergence of the method.

Because the function G is constant, the exact solution should be found after one Picard iteration for any starting point x_0. Since Anderson acceleration starts with one regular Picard iteration, this implies that the solution is also found after one iteration with Anderson, regardless of depth m and starting point x_0.

To verify this, $f(x) = 2x + 1$ and $G(x) = -0.5$ with solution $x_* = -0.5$ are used in the implementations. For Newton's method, the Picard iteration and Anderson acceleration, the number of iterations done K, final approximations x_K and final errors $|x_K - x_*| = |x_K + 0.5|$ are displayed in respectively Table 3.1, 3.2 and 3.3. This is done for the starting points $x_0 = 0.00$ (which is quite close to the analytical solution), $x_0 = 1\text{e}{+15}$ and $x_0 = -1\text{e}{+15}$ (which are much further away). For every method and every starting point, the error is zero after one iteration, which is precisely what was expected.

Table 3.1: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 0.5|$ for the Newton-Raphson method applied to $f(x) = 2x + 1$.

x_0	K	x_K	$\lvert x_K + 0.5\rvert$
0.00	1	−5.00000e−1	0.00000e+0
1e+15	1	−5.00000e−1	0.00000e+0
−1e+15	1	−5.00000e−1	0.00000e+0

Table 3.2: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 0.5|$ for the Picard iteration applied to $G(x) = -0.5$.

x_0	K	x_K	$\lvert x_K + 0.5\rvert$
0.00	1	−5.00000e−1	0.00000e+0
1e+15	1	−5.00000e−1	0.00000e+0
−1e+15	1	−5.00000e−1	0.00000e+0

Table 3.3: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 0.5|$ for Anderson acceleration applied to $G(x) = -0.5$.

x_0	m	K	x_K	$\lvert x_K + 0.5\rvert$
0.00	1	1	−5.00000e−1	0.00000e+0
0.00	2	1	−5.00000e−1	0.00000e+0
0.00	3	1	−5.00000e−1	0.00000e+0
1e+15	1	1	−5.00000e−1	0.00000e+0
1e+15	2	1	−5.00000e−1	0.00000e+0
1e+15	3	1	−5.00000e−1	0.00000e+0
−1e+15	1	1	−5.00000e−1	0.00000e+0
−1e+15	2	1	−5.00000e−1	0.00000e+0
−1e+15	3	1	−5.00000e−1	0.00000e+0

3.2. Convergence for contractions

The second problem is the quadratic root-finding problem

$$f(x) = bx^2 - x = 0 \text{ with } b > 0, \tag{3.4}$$

with analytical solutions $x_a = 0$ and $x_b = \frac{1}{b}$.

This problem is equivalent to the fixed-point problem

$$x = G(x) = bx^2 \text{ with } b > 0. \tag{3.5}$$

The function G is, according to (2.11), a contraction on $X \subseteq \mathbb{R}$ when

$$\exists \alpha \in [0,1)\ \forall \vec{x}, \vec{y} \in X : |G(x) - G(y)| = |bx^2 - by^2| = b|x+y||x-y| \leq a|x-y| \Leftrightarrow |x+y| < \frac{1}{b}. \tag{3.6}$$

The following subsections respectively treat the convergence of Newton's method, the Picard iteration and Anderson acceleration for this problem.

3.2.1. Convergence of Newton-Raphson for contractions

As mentioned in Section 2.1, Newton's method converges quadratically when the multiplicity of a root is equal to one and linearly when the multiplicity is greater than one, meaning $f'(x_*) = 0$. The solutions of problem (3.4) have multiplicity one, since $f'(0) = 2 \cdot b \cdot 0 - 1 = -1 \neq 0$ and $f'(\frac{1}{b}) = 2 \cdot b \cdot \frac{1}{b} - 1 = 1 \neq 0$. This implies that the method converges quadratically when the starting point is close enough to the desired root.

The function has a global minimum at $x_{\min} = \frac{1}{2b}$, since $f'(\frac{1}{2b}) = 2 \cdot b \cdot \frac{1}{2b} - 1 = 0$ and $f''(1) = 2b > 0$. Therefore, the tangent line is always below the graph and switches directions in $x_{\min}$. Since each iteration is the root of the tangent line in the previous iteration (see Figure 2.1), the approximations can not skip over $x_{\min}$. This implies the approximations should go in the direction of x_a when the starting point is chosen to be smaller than $x_{\min}$ and in the direction of x_b otherwise.

To check that this is true, and that Newton's method converges quadratically, $f(x) = 0.5x^2 - x$ with solutions $x_a = 0$ and $x_b = 2$ is used in the implementation. This function has a minimum at $x_{\min} = 1$.

In the first two plots of Figure 3.1, the errors $|x_k - x_a| = |x_k - 0|$ per iteration k are displayed for the starting points $x_a < x_0 = 0.95 < x_{\min}$ and $x_0 = -1\text{e}{+15}$. Table 3.4 shows these starting points with the total number of iterations K needed to reach tolerance $\epsilon = 1\text{e}{-5}$, final approximations x_K, and final errors $|x_K - x_a|$. In the last plot of this figure and row of this table, the same is shown for $x_{\min} < x_0 = 1.05 < x_b$, except now the errors are with respect to $x_b = 2$.

From the first two plots it can be concluded that Newton's method indeed converges towards 0 when the starting point is smaller than $x_{\min}$, even when x_0 is very far away from the solution. Furthermore, the errors $|x_k - 0|$ can be bounded by $0.5|x_{k-1} - 0|^2$. Thus, condition (2.8) is satisfied, meaning the convergence is indeed quadratic.

The last plot shows that the method converges towards 2 when the starting point is bigger than $x_{\min}$. By the same reasoning as before, the convergence is quadratic.

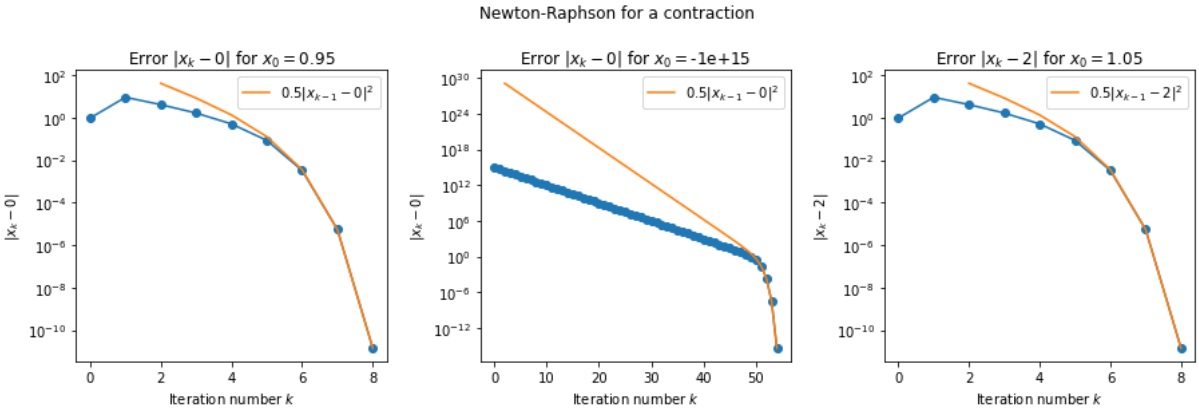

Figure 3.1: Error $|x_k - x_a| = |x_k - 0|$ for starting points $x_0 = 0.95$ and $x_0 = -1\text{e}{+15}$ and $|x_k - x_b| = |x_k - 2|$ for starting point $x_0 = 1.05$ per iteration k for the Newton-Raphson method applied to $f(x) = 0.5x^2 - x$.

Table 3.4: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_a| = |x_K - 0|$ when converging to 0 or $|x_K - x_b| = |x_K - 2|$ when converging to 2 for the Newton-Raphson method applied to $f(x) = 0.5x^2 - x$.

x_0	K	x_K	$\lvert x_K - 0\rvert$	$\lvert x_K - 2\rvert$
0.95	8	−1.49215e−11	1.49215e−11	-
−1e+15	54	−4.50736e−16	4.50736e−16	-
1.05	8	2.00000e+0	-	1.492184e−11

3.2.2. Convergence of Picard for contractions

Using (3.6), Banach's theorem says that the Picard iteration should converge to $x_a = 0$ for all starting points in intervals containing 0 and satisfying $|x + y| < \frac{1}{b}$. There are no intervals containing $x_b = \frac{1}{b}$ that satisfy this condition, so the theorem does not say anything about this solution.

By numerical testing it is found that this problem converges to $x_a = 0$ when $x_0 \in (-\frac{1}{b}, \frac{1}{b})$, even though $|x + y|$ is not smaller than $\frac{1}{b}$ for every x and y in this interval. To show this, $G(x) = 0.5x^2$ is chosen as an example. For this value of b, the method converges to 0 when $x_0 \in (-2, 2)$.

In Figure 3.2, the errors $|x_k - 0|$ per iteration k are plotted for starting points around the right edge of the interval $(-2, 2)$, namely $x_0 = 1.95$ and $x_0 = 2.05$. The middle plot shows the errors $|x_k - 2|$ for starting point $x_0 = 2.00$. The number of iterations done, final approximations and final errors can be found in Table 3.5.

In the first plot of Figure 3.2 and first row of Table 3.5, it can be seen that the Picard iteration does converge towards the solution $x_a = 0$ when the starting point is smaller than 2.00. The errors can be bounded by $0.5|x_{k-1} - 0|^2$, implying quadratic convergence by (2.8). This can be explained by Banach's theorem. In (3.6), a can be written as $b|x + y|$. In combination with (2.12) this gives

$$\begin{aligned} |x_{k+1} - x_a| &\leq a|x_k - x_a| \\ |x_{k+1} - x_a| &\leq b|x_k + x_a||x_k - x_a| \\ |x_{k+1} - 0| &\leq b|x_k + 0||x_k - 0| \\ \frac{|x_{k+1} - 0|}{|x_k - 0|^2} &\leq b \end{aligned} \tag{3.7}$$

Thus the Picard iteration converges quadratically at rate b in this case.

When $x_0 = 2.00$, the method stays at this second analytical solution, since $|G(x_0) - x_0| = 0 < \delta = 1\text{e}{-10}$. Finally, when $x_0 > 2.00$, the method diverges. The results are similar for the left edge of the interval $(-2, 2)$.

These results show that there can be starting points that lead to convergence of the Picard iteration even though they do not follow from Banach's theorem. However, this does not have to be the case for every fixed-point, as the solution $x_b = \frac{1}{b}$ is not found for any other starting points than $x_0 = \pm x_b$.

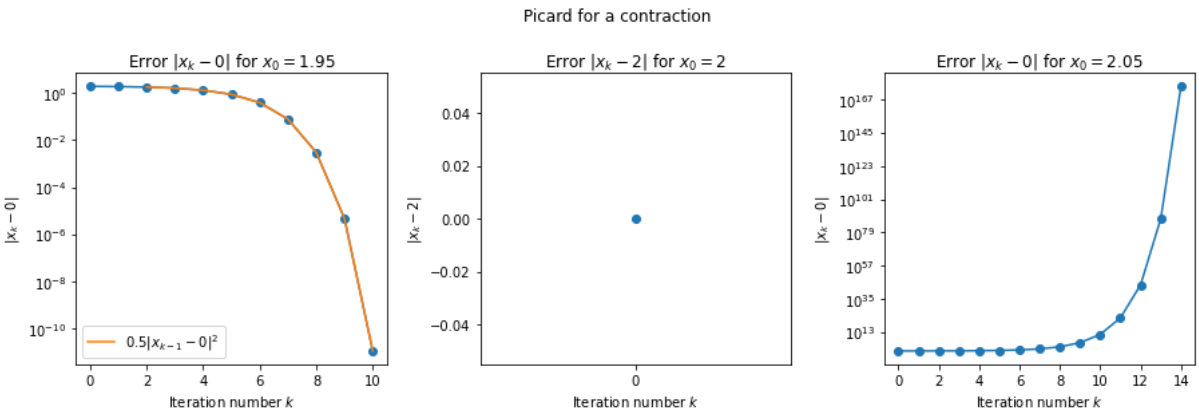

Figure 3.2: Error $|x_k - x_a| = |x_k - 0|$ for starting points $x_0 = 1.95$ and $x_0 = 2.05$ and $|x_k - x_b| = |x_k - 2|$ for starting point $x_0 = 2$ per iteration k for the Picard iteration applied to $G(x) = 0.5x^2$.

Table 3.5: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_a| = |x_K - 0|$ when converging to 0 or $|x_K - x_b| = |x_K - 2|$ when converging to 2 for the Picard iteration applied to $G(x) = 0.5x^2$.

x_0	K	x_K	$\lvert x_K - 0\rvert$	$\lvert x_K - 2\rvert$
1.95	10	1.10092e−11	1.10092e−11	-
2.00	0	2.00000e+0	-	0.00000e+0
2.05	14	1.00194e+176	1.00194e+176	-

3.2.3. Convergence of Anderson acceleration for contractions

For Anderson acceleration applied to $G(x) = 0.5x^2$, it is found by numerical testing that there are alternating intervals for which the method converges to $x_a = 0$ and $x_b = 2$. This is apparent from Table 3.6, which shows different starting points, the depth of the acceleration, number of iterations done, final approximations and final errors.

For instance, for a depth of $m = 1$, the method converges to 0 when $x_0 \in [-1.55, 1.28]$, but also when x_0 is smaller than -4.40. In between these values, the method converges towards 2. When $m = 2$ or $m = 3$, these intervals are slightly different, but similar.

To illustrate this difference, the errors $|x_k - x_a| = |x_k - 0|$ per iteration are displayed in Figure 3.3 and the errors $|x_k - x_b| = |x_k - 2|$ in Figure 3.4 for starting points $x_0 = 1.26$, $x_0 = 1.27$ and $x_0 = 1.28$. In the left plots, it is visible that the iterations converge towards $x_a = 0$ for every choice of m. The middle plots show that the method first starts converging to $x_b = 2$ when $x_0 = 1.27$ and $m = 3$, and the last plots show that this is the case for $m = 2$ when $x_0 = 1.28$.

These results show that Anderson acceleration can converge for more choices of x_0 compared to the Picard iteration. Whereas the Picard method only converges to x_a on a limited interval and does not converge to x_b for starting points other than $\pm x_b$, Anderson acceleration shows convergence to a solution for more choices of x_0. However, it is hard to say to which solution the method will converge for a certain starting point and choice of depth m. Furthermore, an ideal choice of m does not directly follow from these results, as the number of iterations needed and order of the error varies per starting point and is quite similar for different m.

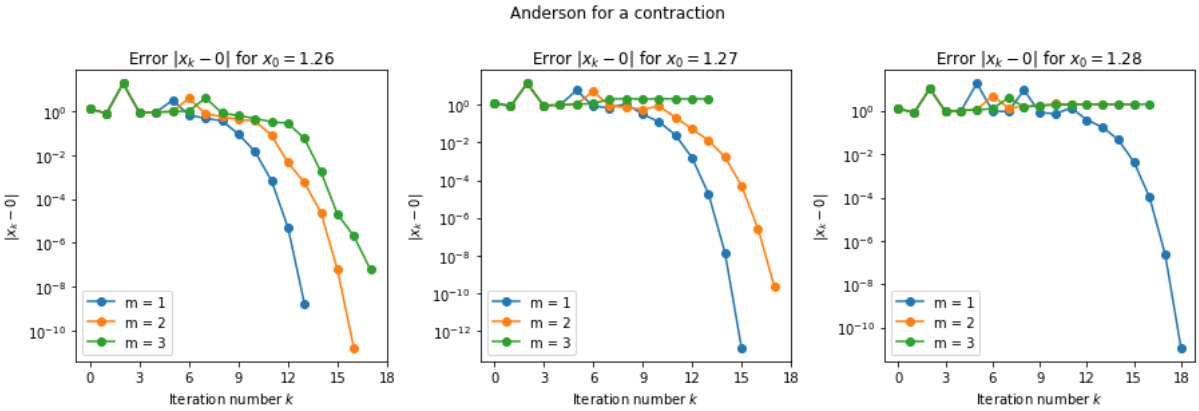

Figure 3.3: Error $|x_k - x_a| = |x_k - 0|$ per iteration k for Anderson acceleration applied to $G(x) = 0.5x^2$ with starting points $x_0 = 1.26$, $x_0 = 1.27$ and $x_0 = 1.28$.

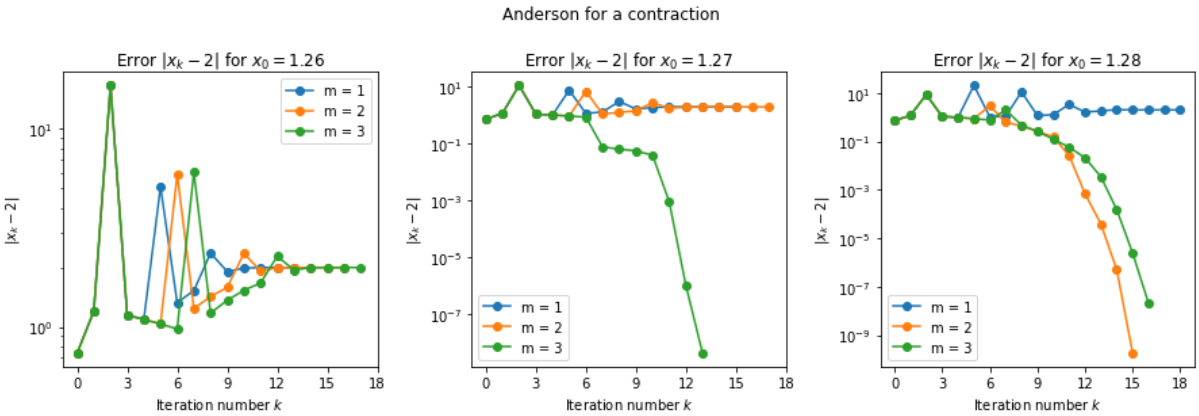

Figure 3.4: Error $|x_k - x_b| = |x_k - 2|$ per iteration k for Anderson acceleration applied to $G(x) = 0.5x^2$ with starting points $x_0 = 1.26$, $x_0 = 1.27$ and $x_0 = 1.28$.

Table 3.6: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_a| = |x_K - 0|$ when converging to 0 or $|x_K - x_b| = |x_K - 2|$ when converging to 2 for Anderson acceleration applied to $G(x) = 0.5x^2$.

x_0	m	K	x_K	$\lvert x_K - 0\rvert$	$\lvert x_K - 2\rvert$
−4.65	1	17	−6.71481e−12	6.71481e−12	-
−4.65	2	11	1.42635e−9	1.42635e−9	-
−4.65	3	12	−5.48557e−8	5.48557e−8	-
−4.40	1	27	−1.35525e−11	1.35525e−11	-
−4.40	2	18	2.00000e+0	-	2.52357e−9
−4.40	3	21	2.00000e+0	-	2.76947e−8
−1.75	1	9	2.00000e+0	-	5.92002e−11
−1.75	2	10	2.00000e+0	-	1.29216e−10
−1.75	3	11	−8.31863e−9	8.31863e−9	-
−1.60	1	14	2.00000e+0	-	1.43152e−12
−1.60	2	10	−1.57570e−11	1.57570e−11	-
−1.60	3	10	−4.80836e−8	4.80836e−8	-
−1.55	1	13	5.87166e−11	5.87166e−11	-
−1.55	2	10	−1.59548e−10	1.59548e−10	-
−1.55	3	10	−1.54943e−8	1.54943e−8	-
1.26	1	13	1.76605e−9	1.76605e−9	-
1.26	2	16	−1.63632e−11	1.63632e−11	-
1.26	3	17	−6.12777e−8	6.12777e−8	-
1.27	1	15	1.26410e−13	1.26410e−13	-
1.27	2	17	−2.17295e−10	2.17295e−10	-
1.27	3	13	2.00000e+0	-	4.33840e−9
1.28	1	18	1.12537e−11	1.12537e−11	-
1.28	2	15	2.00000e+0	-	1.78969e−10
1.28	3	16	2.00000e+0	-	2.07670e−8

3.3. Convergence for multiple roots

As mentioned in Section 2.1, when $f'(x_*) = 0$ for a root x_* of f, x_* is called a multiple root. In this section, the root-finding problem

$$f(x) = x^2 + 2x + 1 = 0 \tag{3.8}$$

with root $x_* = -1$ is considered. This is a multiple root, because $f'(-1) = 2 \cdot -1 + 2 = 0$.

By isolating x in Equation (3.8), it is found that the root-finding problem is equivalent to the fixed-point problem

$$x = G(x) = -0.5x^2 - 0.5. \tag{3.9}$$

In the following subsections, the convergence properties for this problem are discussed for respectively Newton's method, the Picard iteration and Anderson acceleration.

3.3.1. Convergence of Newton-Raphson for multiple roots

For multiple roots, Newton's method converges only linearly, as was found in Section 2.1. Furthermore, since $f'(x_*) = 0$ and $f''(x_*) = 2 > 0$, the root x_* is equal to $x_{\min}$. Thus, Section 3.2.1 suggests the method should converge to x_* when $x_0 > x_*$ and when $x_0 < x_*$. This is verified by using problem (3.8) in the implementation of the method.

In Figure 3.5, the errors $|x_k - x_*| = |x_k + 1|$ per iteration k are displayed for the starting points $x_0 = 0.00$, $x_0 = 1\text{e}{+15}$ and $x_0 = -1\text{e}{+15}$. Table 3.7 contains these starting points with the total number of iterations done, final approximations, and final errors.

Each plot can be bounded by $0.5|x_{k-1} + 1|$, implying the convergence is indeed linear by (2.9). The rate of convergence is 0.5, which is the slope of the plots. Besides this, the plots show that the method converges to x_* when $x_0 > x_*$, namely when $x_0 = 0.00$ and $x_0 = 1\text{e}{+15}$, but also when $x_0 < x_*$, namely when $x_0 = -1\text{e}{+15}$, as was expected. However, because the convergence is only linear, the method does need quite a lot of iterations when x_0 is very far away from the root, which is apparent from the last two plots.

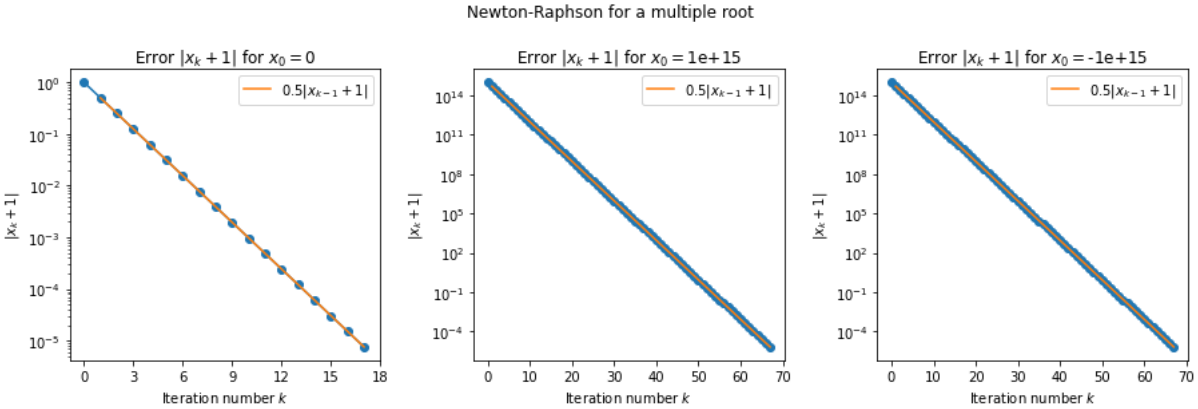

Figure 3.5: Error $|x_k - x_*| = |x_k + 1|$ per iteration k for the Newton-Raphson method applied to $f(x) = x^2 + 2x + 1$ with starting points $x_0 = 0.00$, $x_0 = 1\text{e}{+15}$ and $x_0 = -1\text{e}{+15}$.

Table 3.7: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 1|$ for the Newton-Raphson method applied to $f(x) = x^2 + 2x + 1$.

x_0	K	x_K	$\lvert x_K + 1\rvert$
0.00	17	−9.99992e−1	8.00000e−6
1e+15	67	−9.99993e−1	7.00000e−6
−1e+15	67	−1.00001e+0	7.00000e−6

3.3.2. Convergence of Picard for multiple roots

The function $G(x) = -0.5x^2 - 0.5$ is a contraction when

$$|G(x) - G(y)| = |-0.5x^2 + 0.5y^2| = 0.5|x + y||x - y| \leq a|x - y| \text{ with } 0 \leq a < 1 \Leftrightarrow |x + y| < 2. \quad (3.10)$$

By numerical testing it is found that the interval that x_0 should be in, in order for the Picard iteration to converge, is $[-1, 1]$. Apart from the boundaries, this interval satisfies the condition $|x+y| < 2$. Therefore, the method does not converge on intervals where G is not a contraction, in contrast to Section 3.2.2.

Convergence in this interval follows from Figure 3.6, showing the errors $|x_k - x_*| = |x_k + 1|$ for starting points around the right edge of the interval, $x_0 = 0.95$, $x_0 = 1.00$ and $x_0 = 1.05$, and Table 3.8, containing the number of iterations done, final approximations and final errors for these starting points. The results on the left edge of the interval are similar.

The first plot can be bounded by $|x_{k-1} + 1|$, implying that the method converges linearly at a rate of 1. This linear convergence is in accordance with Banach's theorem. Because this rate is very slow, the method is stopped after the maximum of 100 iterations.

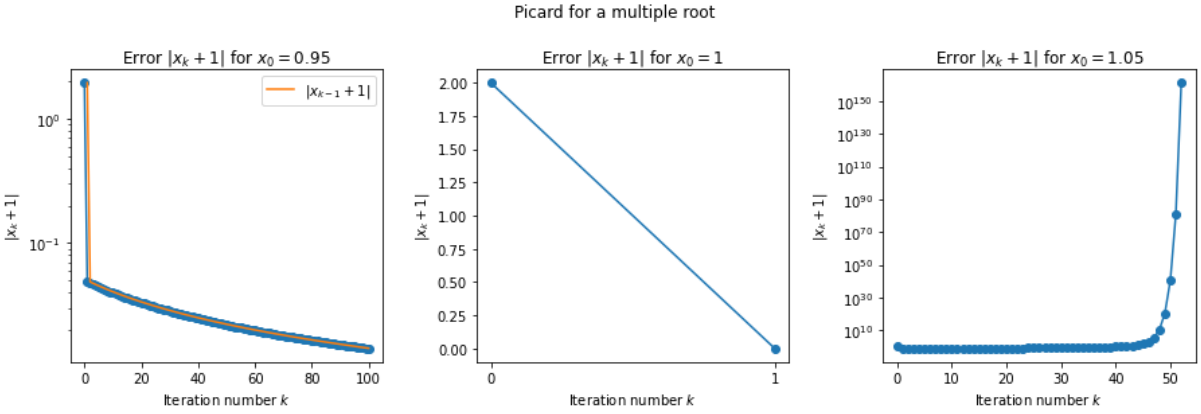

Figure 3.6: Error $|x_k - x_*| = |x_k + 1|$ per iteration k for the Picard iteration applied to $G(x) = -0.5x^2 - 0.5$ with starting points $x_0 = 0.95$, $x_0 = 1.00$ and $x_0 = 1.05$.

Table 3.8: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 1|$ for the Picard iteration applied to $G(x) = -0.5x^2 - 0.5$.

x_0	K	x_K	$\lvert x_K - x + 1\rvert$
0.95	100	−9.85843e−1	1.41572e−2
1.00	1	−1.00000e+0	0.00000e+0
1.05	52	−2.27477e+161	2.27477e+161

3.3.3. Convergence of Anderson acceleration for multiple roots

For a case with two different solutions, it was shown that Anderson acceleration can converge for more choices of x_0 than the Picard iteration in Section 3.2.3. However, it is difficult to say which solution the method will converge to for a given starting point. Because $G(x) = -0.5x^2 - 0.5$ only has one fixed-point, this difficulty should not be present anymore.

In Figure 3.7, the errors $|x_k - x_*| = |x_k + 1|$ per iteration are plotted for starting points both close to and far away from x_*, namely $x_0 = 0.00$, $x_0 = 1\text{e}{+5}$ and $x_0 = 1\text{e}{+6}$. Table 3.9 shows the number of iterations done, final approximations and final errors for these starting points. The results for $x_0 = -1\text{e}{+5}$ and $x_0 = -1\text{e}{+6}$ are similar.

From these results, it can be concluded that Anderson acceleration converges to the solution for x_0 up to an order of 10^5, which is a massive improvement from the Picard iteration, since that method only converges when $x_0 \in [-1, 1]$. Furthermore, in the first two plots, it is visible that the errors can be bounded by $0.74|x_{k-1} + 1|$ or stricter. This means that the convergence, although still linear, has a rate of 0.74 or better, depending on m. This is also an improvement from the Picard iteration.

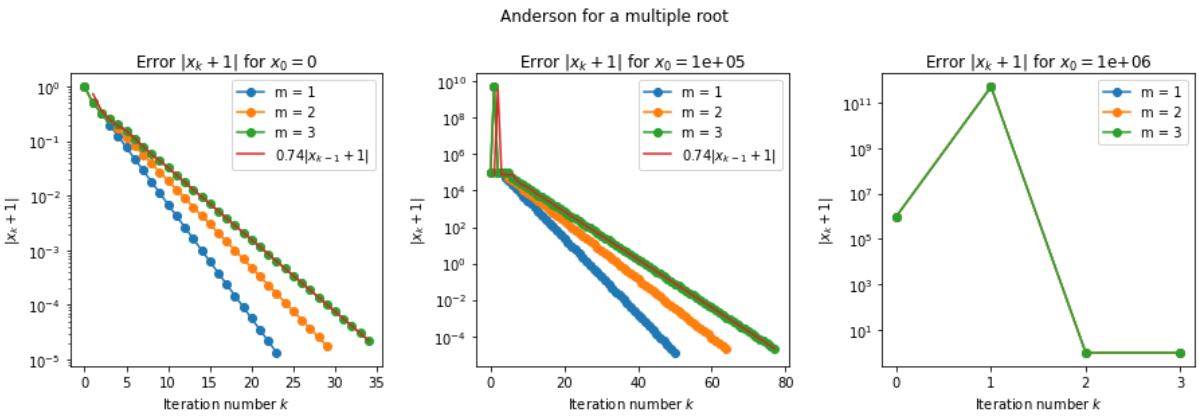

Figure 3.7: Error $|x_k - x_*| = |x_k + 1|$ per iteration k for Anderson acceleration applied to $G(x) = -0.5x^2 - 0.5$ with starting points $x_0 = 0.00$, $x_0 = 1e{+}5$ and $x_0 = 1e{+}6$.

Table 3.9: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K + 1|$ for Anderson acceleration applied to $G(x) = -0.5x^2 - 0.5$.

x_0	m	K	x_K	$\lvert x_K + 1\rvert$
0.00	1	23	−9.99987e−1	1.30000e−5
0.00	2	29	−9.99982e−1	1.80000e−5
0.00	3	34	−9.99978e−1	2.20000e−5
1e+5	1	50	−9.99987e−1	1.30000e−5
1e+5	2	64	−9.99980e−1	2.00000e−5
1e+5	3	77	−9.99978e−1	2.20000e−5
1e+6	1	3	1.99996e−12	1.00000e+0
1e+6	2	3	1.99996e−12	1.00000e+0
1e+6	3	3	1.99996e−12	1.00000e+0

3.4. Cases where Newton-Raphson fails

Newton's method has converged for all discussed problems so far. However, there are situations where the method does not converge. This happens when the method diverges away from the solution, or oscillates between values.

This first case occurs when trying to find root $x_* = 0$ of the problem

$$f(x) = \arctan(x) = 0. \tag{3.11}$$

In Figure 3.8, showing the errors $|x_k - x_*| = |x_k - 0|$ for $x_0 = 0.50$, $x_0 = 1.35$ and $x_0 = 1.40$, and Table 3.10, containing these starting points with the number of iterations done, final approximations and final errors, it can be seen that the method diverges when choosing $x_0 = 1.40$. This is also the case for starting points bigger than 1.40.

However, when x_0 is close to, but smaller than, 1.40, for instance $x_0 = 1.35$, the method converges quite quickly, as can be seen in the middle of Figure 3.8 and Table 3.10. The first plot of this figure and first row of this table show that the method performs even better when the starting point is very close to the analytical root, which is the case for $x_0 = 0.50$. This may be because the arctangent behaves like a linear function around $x_* = 0$, as the derivative $\frac{1}{1+x^2}$ approaches a constant, namely 1, and Newton's method converges extremely quickly for linear functions, as was shown before.

These results are the same for negative values of x_0. They show that the Newton-Raphson method is quite sensitive to the choice of the starting point x_0.

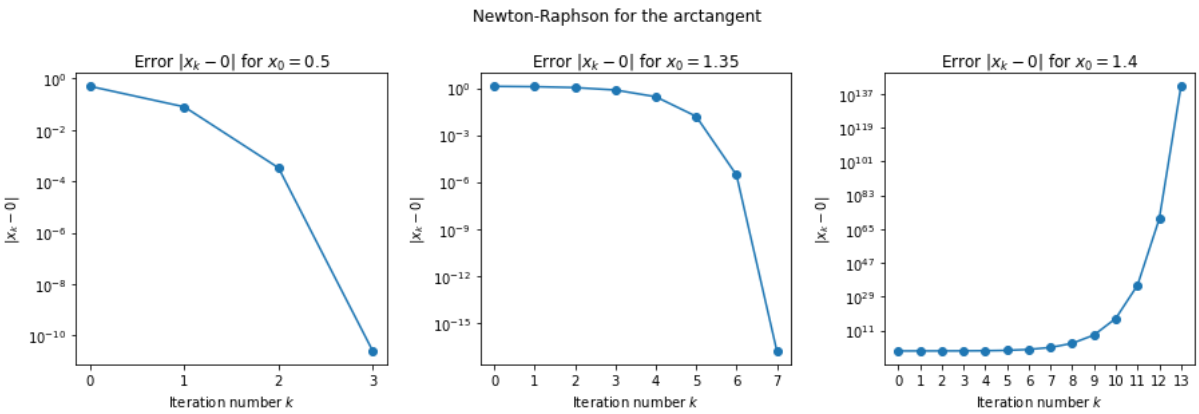

Figure 3.8: Error $|x_k - x_*| = |x_k - 0|$ per iteration k for the Newton-Raphson method applied to $f(x) = \arctan(x)$ with starting points $x_0 = 0.50$, $x_0 = 1.35$ and $x_0 = 1.40$.

Table 3.10: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 0|$ for the Newton-Raphson method applied to $f(x) = \arctan(x)$.

x_0	K	x_K	$\lvert x_K - 0\rvert$
0.50	3	−2.51315e−11	2.51315e−11
1.35	7	−1.56129e−17	1.56129e−17
1.40	13	−1.18247e+141	1.18247e+141

In Figure 3.9, three iterations of Newton's method for the arctangent with starting point $x_0 = 1.40$ are shown. From this visualisation, it becomes clear that the approximations move away from root x_*, which is why the method does not converge for this starting point.

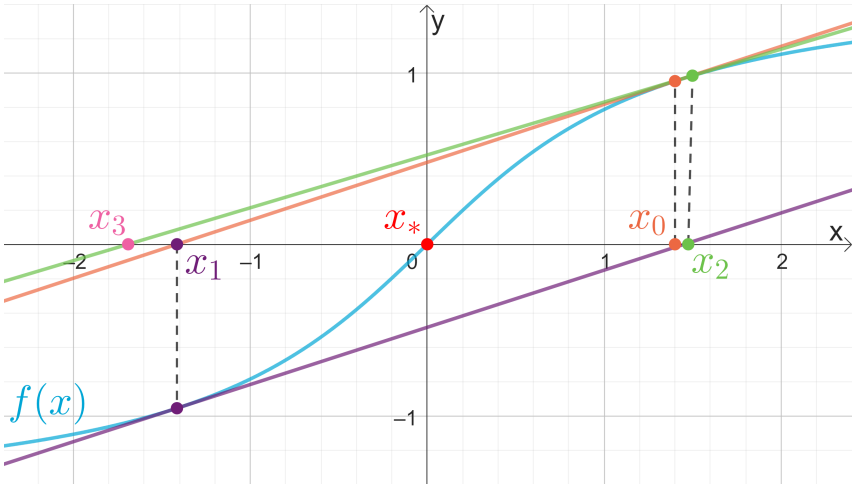

Figure 3.9: Three iterations of the Newton-Raphson method applied to $f(x) = \arctan(x)$ with starting point $x_0 = 1.40$.

Newton's method oscillates when trying to find the root $x_* = 1.671699882$ of the problem

$$f(x) = -x^3 + x + 3 = 0. \tag{3.12}$$

This happens when $x_0 = -3.00$, as can be seen in the first plot of Figure 3.10, showing the errors $|x_k - x_*| = |x_k - 1.671699882|$. Since the iterations are stopped at $N = 100$, they are prevented from oscillating endlessly. However, when the starting point is close to -3.00, for instance $x_0 = -3.05$ or $x_0 = -2.95$, the method does converge. This is apparent from the other plots and from Table 3.11, which shows the number of iterations done, final approximations and final errors.

Again, this example shows that the performance of Newton's method is heavily decided by the choice of the starting point x_0.

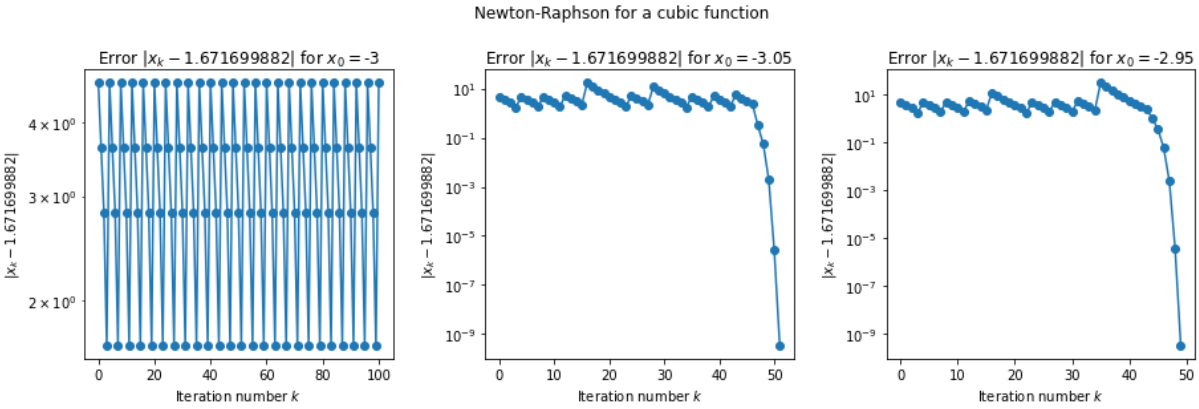

Figure 3.10: Error $|x_k - x_*| = |x_k - 1.671699882|$ per iteration k for the Newton-Raphson method applied to $f(x) = -x^3 + x + 3$ with starting points $x_0 = -3.00$, $x_0 = -3.05$ and $x_0 = -2.95$.

Table 3.11: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 1.671699882|$ for the Newton-Raphson method applied to $f(x) = -x^3 + x + 3$.

x_0	K	x_K	$\lvert x_K - 1.671699882\rvert$
-3.00	100	$-3.00050\text{e+}0$	$4.67220\text{e+}0$
-3.05	51	$1.67170\text{e+}0$	$3.38508\text{e}{-10}$
-2.95	49	$1.67170\text{e+}0$	$3.32664\text{e}{-10}$

Figure 3.11 shows four iterations of Newton's method for this problem with starting point $x_0 = -3.00$. In this figure, it can be seen that approximation x_4 coincides with x_0. Because of this, the method loops between four different values.

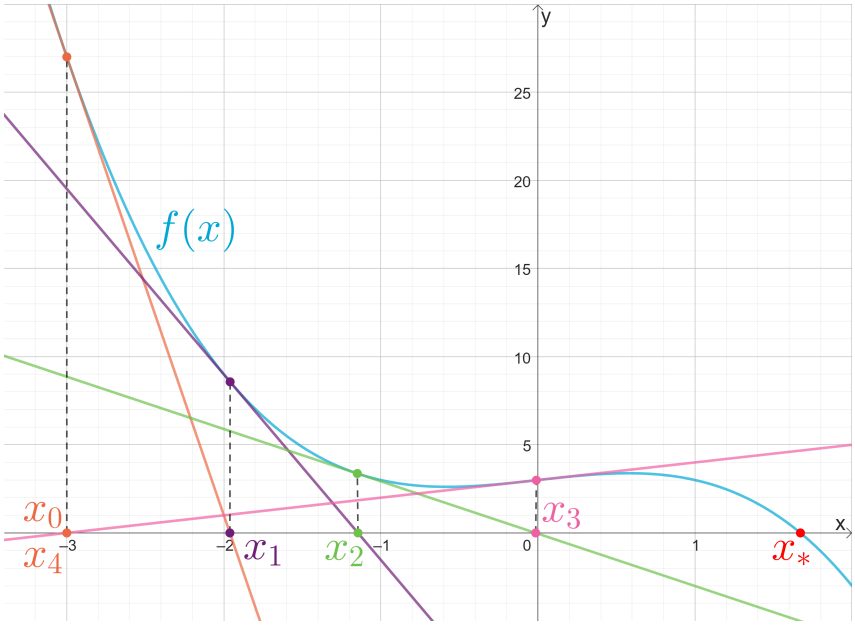

Figure 3.11: Four iterations of the Newton-Raphson method applied to $f(x) = -x^3 + x + 3$ with starting point $x_0 = -3.00$.

3.4.1. Picard when Newton-Raphson fails

Problem (3.11) with solution $x_* = 0$ is equivalent to the fixed-point problem

$$x = G(x) = \tan(0) = 0. \tag{3.13}$$

The Picard iteration will find the solution of this equation in one step, since G is constant. The problem can also be written as

$$x = G_1(x) = x + \arctan(x). \tag{3.14}$$

However, the method does not converge for any x_0 for this function. This could be due to the fact that this function is not a contraction on any interval. Because G_1 is continuous and differentiable on $\mathbb{R}$, by the Mean Value Theorem (Lay, 2014) there exists a c in any interval (x, y) such that

$$|G_1'(c)| = \frac{|G_1(y) - G_1(x)|}{|y - x|} \Leftrightarrow |G_1(x) - G_1(y)| = |G_1'(c)||x - y|. \tag{3.15}$$

The term $|G_1'(c)|$ can be bounded below by 1, since

$$|G_1'(c)| = |1 + \frac{1}{1 + c^2}| > 1. \tag{3.16}$$

Since this holds for any interval (x, y), G_1 is never a contraction by definition (2.11). This means that Banach's theorem does not give any starting points for which the method converges.

There is, however, an equivalent problem for which the method does converge, namely

$$x = G_2(x) = x - \arctan(x). \tag{3.17}$$

Whereas $|G_1'(c)|$ could be bounded from below, $|G_2'(c)|$ can be bounded from above, because

$$|G_2'(c)| = |1 - \frac{1}{1 + c^2}| < 1 \text{ if } \frac{1}{1 + c^2} < 2 \Leftrightarrow c^2 > -0.5. \tag{3.18}$$

Because this is always satisfied, the Picard iteration should always converge by Banach's theorem.

In order to check this, both G_1 and G_2 are used in the implementation of the method. Figure 3.12 shows the errors $|x_k - x_*| = |x_k - 0|$ for both functions for the starting points $x_0 = 0.05$, $x_0 = 1.40$ and $x_0 = 1\text{e}{+5}$. Table 3.12 contains these starting points with the number of iterations done, final approximations and final errors for G_1 and Table 3.13 shows this information for G_2.

Where Newton's method starts to fail around $x_0 = 1.40$, the Picard iteration diverges for G_1 even when the starting point is very close to the solution, for instance $x_0 = 0.05$. It does diverge at a much slower pace than Newton's method. Even when x_0 is far away from the root, such as $x_0 = 1\text{e}{+5}$, after the maximum of 100 iterations the approximations still do not exceed the limit $L = 1\text{e}{+100}$. For G_2, however, the method does converge for $x_0 = 0.05$ and $x_0 = 1.40$. The iterations do go towards 0 when $x_0 = 1\text{e}{+5}$, although this process is very slow.

These experiments show that the performance of the Picard iteration is heavily influenced by the choice of G. Equivalent problems can have very different outcomes when G is chosen slightly differently.

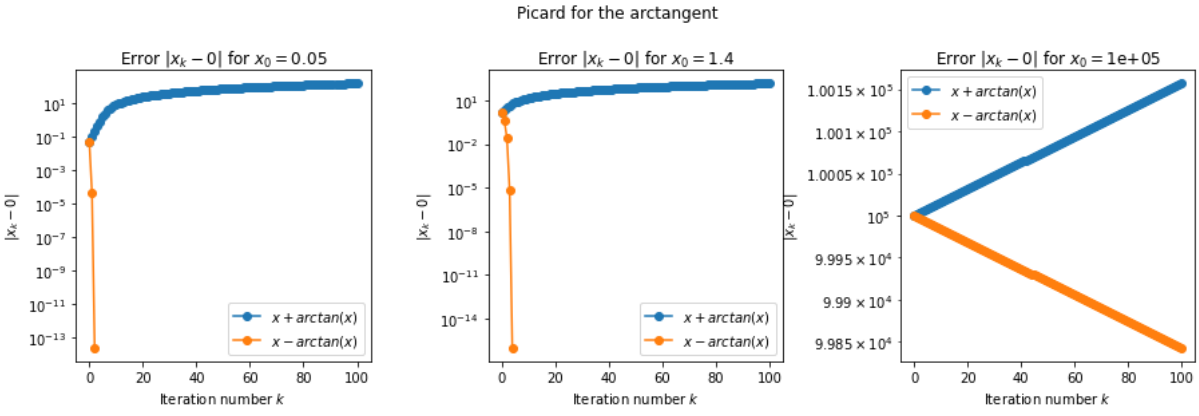

Figure 3.12: Error $|x_k - x_*| = |x_k - 0|$ per iteration k for the Picard iteration applied to $G_1(x) = x + \arctan(x)$ and $G_2(x) = x - \arctan(x)$ with starting points $x_0 = 0.05$, $x_0 = 1.40$ and $x_0 = 1\text{e}{+5}$.

Table 3.12: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 0|$ for the Picard iteration applied to $G_1(x) = x + \arctan(x)$.

x_0	K	x_K	$\lvert x_K - 0\rvert$
0.05	100	1.47020e+2	1.47020e+2
1.40	100	1.54777e+2	1.54777e+2
1e+5	100	1.00157e+5	1.00157e+5

Table 3.13: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 0|$ for the Picard iteration applied to $G_2(x) = x - \arctan(x)$.

x_0	K	x_K	$\lvert x_K - 0\rvert$
0.05	2	2.40045e−14	2.40045e−14
1.40	4	9.57274e−17	9.57274e−17
1e+5	100	9.98429e+4	9.98429e+4

The difference between G_1 and G_2 is visible in Figure 3.13, which shows three iterations of the Picard iteration for both functions with starting point $x_0 = 1.40$. For G_1, the approximations move away from the solution x_*, as the function is not a contraction. For G_2, however, the approximations quickly approach x_*.

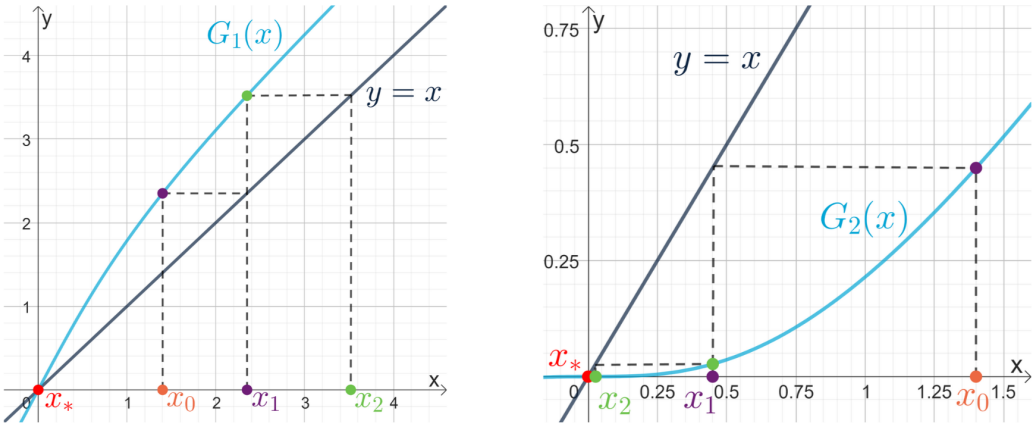

Figure 3.13: Three iterations of the Picard iteration applied to $G_1(x) = x + \arctan(x)$ and $G_2(x) = x - \arctan(x)$ with starting point $x_0 = 1.40$.

Newton's method also fails to find root $x_* = 1.671699882$ of problem (3.12) when starting at $x_0 = -3.00$. This problem can be written as the fixed-point problem

$$x = G(x) = x^3 - 3. \tag{3.19}$$

This G is a contraction when

$$|G(x) - G(y)| = |x^3 - y^3| = |x^2 + xy + y^2||x - y| \leq a|x - y| \text{ with } 0 \leq a < 1 \Leftrightarrow |x^2 + xy + y^2| < 1. \tag{3.20}$$

However, this condition is not satisfied for any interval containing the solution $x_* = 1.671699882$. Therefore, no starting points for which the Picard iteration converges are given by Banach's theorem.

Indeed, in Figure 3.14 and Table 3.14, it can be seen that for starting points close to the solution, $x_0 = 1.60$ and $x_0 = 1.70$, the method does not converge. The method also diverges for $x_0 = -3.00$, whereas Newton's method oscillates for this value.

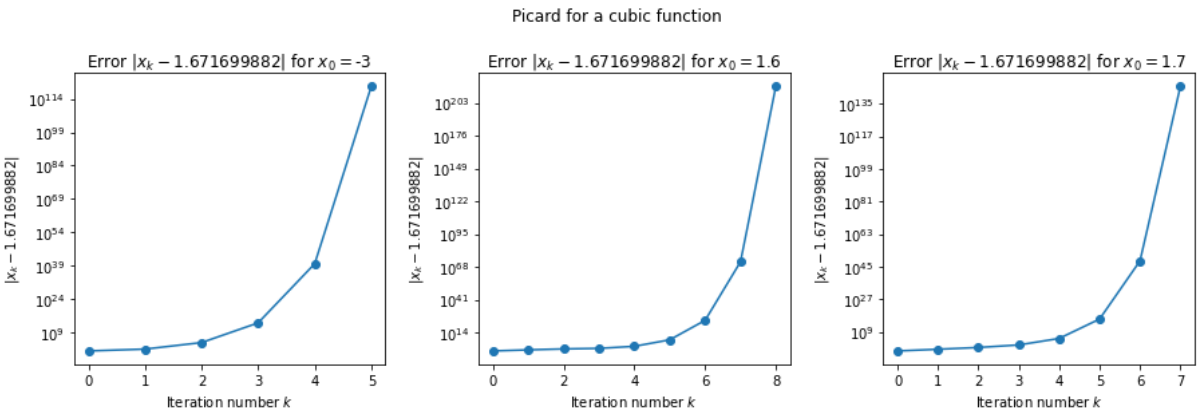

Figure 3.14: Error $|x_k - x_*| = |x_k - 1.671699882|$ per iteration k for the Picard iteration applied to $G(x) = x^3 - 3$ with starting points $x_0 = -3.00$, $x_0 = 1.60$ and $x_0 = 1.70$.

Table 3.14: Starting points x_0, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 1.671699882|$ for the Picard iteration applied to $G(x) = x^3 - 3$.

x_0	K	x_K	$\lvert x_K - 1.671699882\rvert$
-3.00	5	$-4.44759\text{e+}119$	$4.44759\text{e+}119$
1.60	8	$-4.08077\text{e+}216$	$4.08077\text{e+}216$
1.70	7	$4.29015\text{e+}144$	$4.29015\text{e+}144$

3.4.2. Anderson acceleration when Newton-Raphson fails

The experiments with $G_1(x) = x + \arctan(x)$ and $G_2(x) = x - \arctan(x)$ are now repeated for Anderson acceleration. The results for G_1 can be found in Figure 3.15 and Table 3.15 and for G_2 in Figure 3.16 and Table 3.16, using starting points $x_0 = 1.40$, $x_0 = 1\text{e+}5$ and $x_0 = 1\text{e+}6$.

For G_1, Newton's method starts to diverge around $x_0 = 1.40$ and the Picard iteration does not converge at all. However, in Figure 3.15 and Table 3.15 it can be seen that Anderson does converge to 0 for $x_0 = 1.40$ for all choices of m. This is an improvement compared to Newton's method and the Picard iteration. For x_0 that are much further away from x_*, up to $x_0 = 1\text{e+}5$, the method only converges when m is bigger than 1. When x_0 gets even bigger, for $m = 2$ and $m = 3$ the approximations do not converge within the maximum of 100 iterations. However, increasing this maximum to 150 does result in convergence.

For G_2, the results are similar for $x_0 = 1.40$ and $x_0 = 1\text{e+}5$, but for $x_0 = 1\text{e+}6$, the convergence is better for $m = 2$ and $m = 3$ compared to G_1. This implies that the choice of G can also affect the performance of Anderson acceleration.

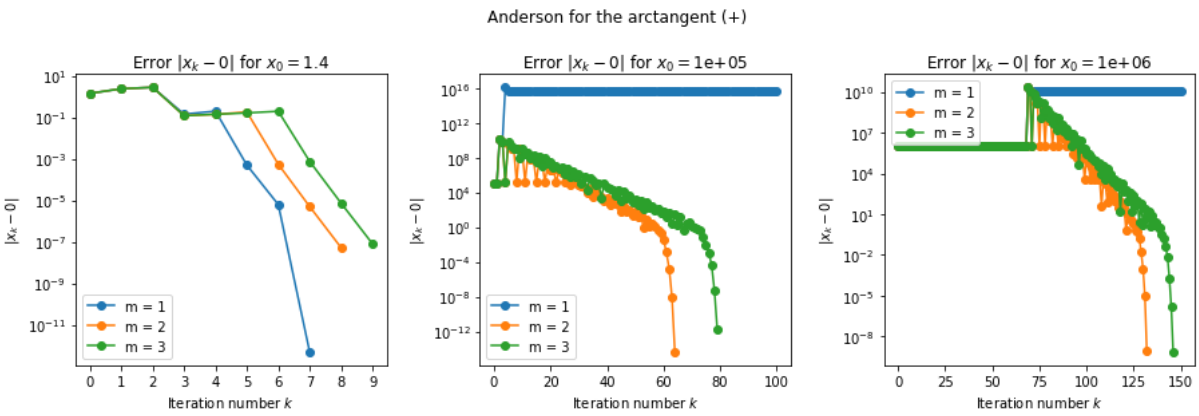

Figure 3.15: Error $|x_k - x_*| = |x_k - 0|$ per iteration k for Anderson acceleration applied to $G_1(x) = x + \arctan(x)$ with starting points $x_0 = 1.40$, $x_0 = 1\text{e+}5$ and $x_0 = 1\text{e+}6$.

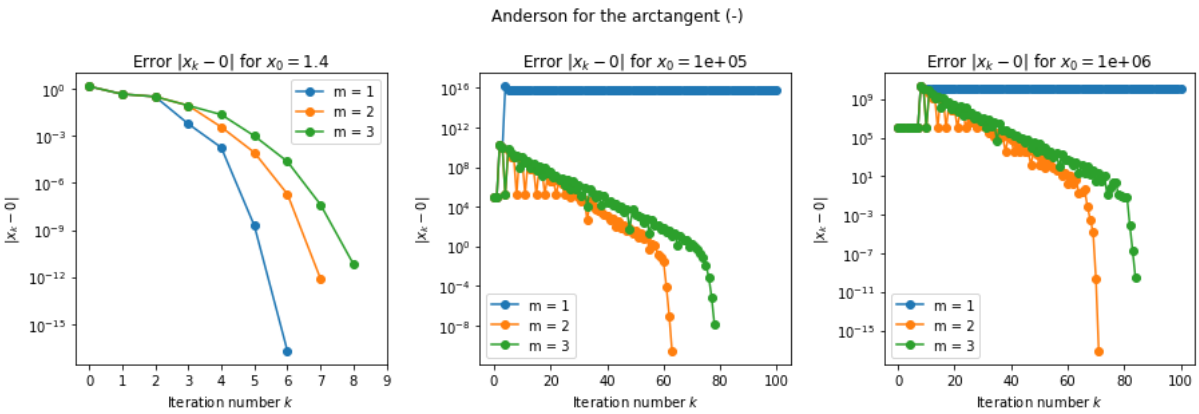

Figure 3.16: Error $|x_k - x_*| = |x_k - 0|$ per iteration k for Anderson acceleration applied to $G_2(x) = x - \arctan(x)$ with starting points $x_0 = 1.40$, $x_0 = 1e+5$ and $x_0 = 1e+6$.

Table 3.15: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 0|$ for Anderson acceleration applied to $G_1(x) = x + \arctan(x)$.

x_0	m	K	x_K	$\lvert x_K - 0\rvert$
1.40	1	7	5.00521e−13	5.00521e−13
1.40	2	8	−5.27628e−8	5.27628e−8
1.40	3	9	8.07958e−8	8.07958e−8
1e+5	1	100	5.58427e+15	5.58427e+15
1e+5	2	64	−4.27655e−15	4.27655e−15
1e+5	3	79	−1.75918e−12	1.75918e−12
1e+6	1	150	−1.05964e+10	1.05964e+10
1e+6	2	132	−8.57585e−10	8.57586e−10
1e+6	3	146	7.32556e−10	7.32556e−10

Table 3.16: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 0|$ for Anderson acceleration applied to $G_2(x) = x - \arctan(x)$.

x_0	m	K	x_K	$\lvert x_K - 0\rvert$
1.40	1	6	−1.96331e−17	1.96331e−17
1.40	2	7	8.00309e−13	8.00309e−13
1.40	3	8	−6.49292e−12	6.49292e−12
1e+5	1	100	6.14270e+15	6.14270e+15
1e+5	2	63	−2.70177e−11	2.70177e−11
1e+5	3	78	1.36195e−8	1.36195e−8
1e+6	1	100	−1.05964e+10	1.05964e+10
1e+6	2	71	−7.22633e−18	7.22633e−18
1e+6	3	84	2.75898e−10	2.75898e−10

The second problem, for which Newton's method oscillates and the Picard iteration diverges, is $x = G(x) = x^3 - 3$ with solution $x_* = 1.671699882$.

The first plot of Figure 3.17 and rows of Table 3.17 show that Anderson acceleration does converge towards x_* when Newton's method starts oscillating, namely when $x_0 = -3.00$. The method also converges for bigger values, like $x_0 = 46.0$, as can be seen in the middle plot. However, after this point Anderson starts to fail: for $x_0 = 47.0$ the solution is no longer found.

These results again implicate that Anderson acceleration can improve the performance of the Picard iteration significantly. Furthermore, the method can converge in places where Newton does not.

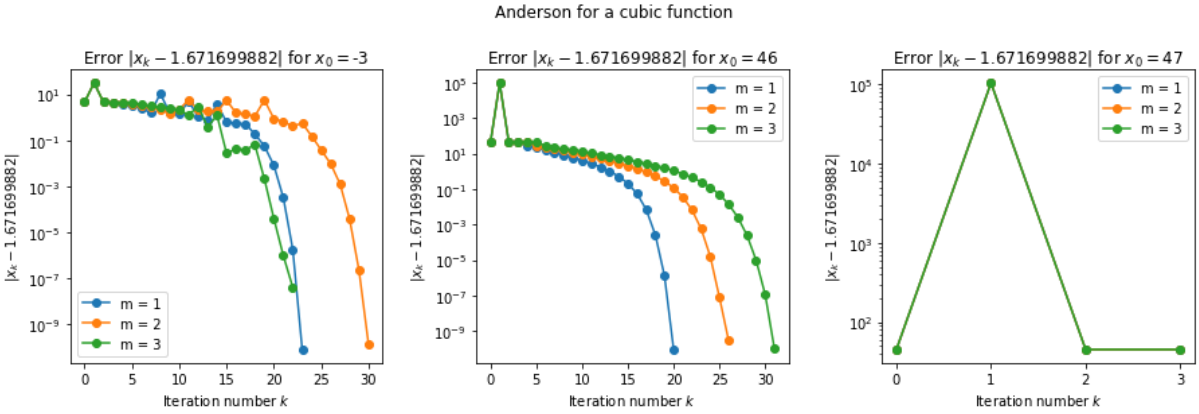

Figure 3.17: Error $|x_k - x_*| = |x_k - 1.671699882|$ per iteration k for Anderson acceleration applied to $G(x) = x^3 - 3$ with starting points $x_0 = -3.00$, $x_0 = 46.0$ and $x_0 = 47.0$.

Table 3.17: Starting points x_0, depth m, total number of iterations done K, final approximations x_K, and final errors $|x_K - x_*| = |x_K - 1.671699882|$ for Anderson acceleration applied to $G(x) = x^3 - 3$.

x_0	m	K	x_K	$\lvert x_K - 1.671699882\rvert$
−3.00	1	23	1.67170e+0	8.26661e−11
−3.00	2	30	1.67170e+0	1.47471e−10
−3.00	3	22	1.67170e+0	4.18748e−8
46.0	1	20	1.67170e+0	9.41829e−11
46.0	2	26	1.67170e+0	3.08048e−10
46.0	3	31	1.67170e+0	1.10712e−10
47.0	1	3	4.70000e+1	4.53283e+1
47.0	2	3	4.70000e+1	4.53283e+1
47.0	3	3	4.70000e+1	4.53283e+1

3.5. Convergence for vector-valued functions

To investigate the convergence of the methods for vector-valued problems, the two dimensional root-finding problem

$$\vec{f}(x_{(1)}, x_{(2)}) = \begin{pmatrix} x_{(1)} + x_{(1)}x_{(2)} + x_{(2)} \\ x_{(1)}^2 - x_{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.21}$$

and the equivalent fixed-point problem

$$\begin{pmatrix} x_{(1)} \\ x_{(2)} \end{pmatrix} = \vec{G}(x_{(1)}, x_{(2)}) = \begin{pmatrix} -x_{(1)}x_{(2)} - x_{(2)} \\ x_{(1)}^2 \end{pmatrix} \tag{3.22}$$

with solution $\vec{x}_* = (0,0)^T$ are considered in the following subsections.

Note that the components of these functions depend on two variables. When the components only depend on one variable, they are just scalar problems, meaning the convergence is as discussed in the previous sections for Newton's method and the Picard iteration. However, Anderson acceleration may behave differently, because of the optimisation step. Therefore, this method is also applied to the fixed-point problem

$$\begin{pmatrix} x_{(1)} \\ x_{(2)} \end{pmatrix} = \vec{G}(x_{(1)}, x_{(2)}) = \begin{pmatrix} -0.5x_{(1)}^2 - 0.5 \\ -0.5x_{(2)}^2 - 0.5 \end{pmatrix} \tag{3.23}$$

with solution $\vec{x}_* = (-1,-1)^T$. The components of this problem are equivalent to (3.9).

3.5.1. Convergence of Newton-Raphson for vector-valued functions

The results of Newton's method applied to problem (3.21) are displayed in Figure 3.18, containing the errors $||\vec{x}_k - (0,0)^T||_2$, and Table 3.18, containing the number of iterations done, final approximations and final errors per starting point. The chosen starting points are $\vec{x}_0 = (0.50, 0.50)^T$, $\vec{x}_0 = (0.50, 1\text{e}{+15})^T$ and $\vec{x}_0 = (1\text{e}{+15}, 0.50)^T$. In the plots, it can be seen that the method converges quadratically for every $\vec{x}_0$, since

$$J(\vec{x}_*) = \begin{pmatrix} 1 + x_{*(2)} & 2x_{*(1)} \\ x_{*(2)} + 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix} \Rightarrow J(\vec{x}_*)^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}. \tag{3.24}$$

It is noticeable that the method needs a lot more iterations for $\vec{x}_0 = (1\text{e}{+15}, 0.50)^T$ than for the other two choices of $\vec{x}_0$. This may be because the second component of $\vec{f}$ is a lot bigger when $x_{0(1)} = 1\text{e}{+15}$ compared to $x_{0(1)} = 0.50$, since it contains the term $x_{(1)}^2$. This implies the performance of Newton's method is sensitive to the choice of the elements of $\vec{x}_0$ when the problem is vector-valued.

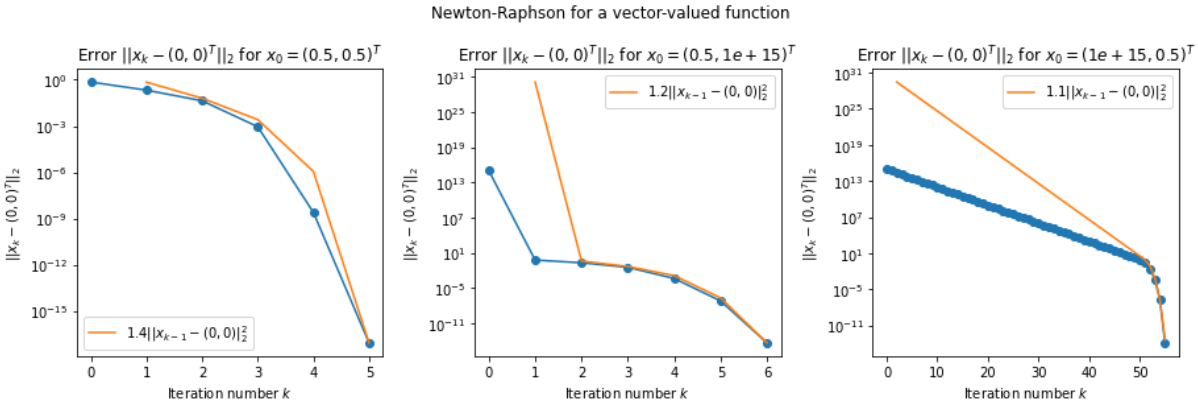

Figure 3.18: Error $||\vec{x}_k - \vec{x}_*||_2 = ||\vec{x}_k - (0,0)^T||_2$ per iteration k for the Newton-Raphson method applied to $\vec{f}(x_{(1)}, x_{(2)}) = (x_{(1)} + x_{(1)}x_{(2)} + x_{(2)}, x_{(1)}^2 - x_{(2)})^T$ with starting points $\vec{x}_0 = (0.50, 0.50)^T$, $\vec{x}_0 = (0.50, 1\text{e}{+15})^T$ and $\vec{x}_0 = (1\text{e}{+15}, 0.50)^T$.

Table 3.18: Starting points $\vec{x}_0$, total number of iterations done K, final approximations $\vec{x}_K$, and final errors $||\vec{x}_K - \vec{x}_*||_2 = ||\vec{x}_K - (0,0)^T||_2$ for the Newton-Raphson method applied to $\vec{f}(x_{(1)}, x_{(2)}) = (x_{(1)} + x_{(1)}x_{(2)} + x_{(2)}, x_{(1)}^2 - x_{(2)})^T$.

$\vec{x}_0$	K	$\vec{x}_K$	$\|\|\vec{x}_K - (0,0)^T\|\|_2$
$(0.50, 0.50)^T$	5	$(5.94154\text{e}{-18}, -5.95953\text{e}{-18})^T$	8.41533e−18
$(0.50, 1\text{e}{+15})^T$	6	$(2.59537\text{e}{-15}, -3.36573\text{e}{-15})^T$	4.25018e−15
$(1\text{e}{+15}, 0.50)^T$	55	$(-5.14795\text{e}{-15}, -1.08239\text{e}{-14})^T$	1.19857e−14

3.5.2. Convergence of Picard for vector-valued functions

The Picard iteration applied to problem (3.22) is extremely sensitive to differences in the elements of $\vec{x}_0$, as can be seen in Figure 3.19 and Table 3.19. When the starting point is $(0.60, 0.60)^T$, the method converges, although the error only reduces every odd step. This is because of the way $x_{(1)}$ and $x_{(2)}$ influence each other. For instance, when $x_{k(1)}$ decreases, $x_{k+1(2)} = x_{k(1)}^2$ also decreases, but when $x_{k(1)}$ increases, $x_{k+1(2)} = x_{k(1)}^2$ also increases.

However, when $x_{0(2)}$ is a little bit bigger, namely 0.65, the method diverges. This is not the case when only $x_{0(1)}$ increases to 0.65. This is the other way around compared to Newton's method. Also, whereas Newton's method converges at a slower pace for certain $\vec{x}_0$, the Picard iteration does not converge at all. From this, it can be concluded that the Picard iteration is even more sensitive to the choice of the elements of $\vec{x}_0$ than Newton's method.

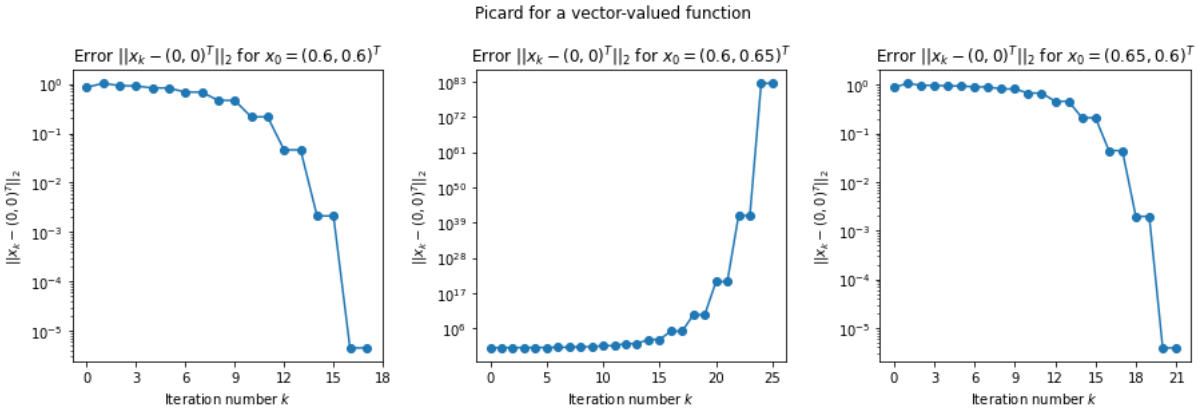

Figure 3.19: Error $||\vec{x}_k - \vec{x}_*||_2 = ||\vec{x}_k - (0,0)^T||_2$ per iteration k for the Picard iteration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-x_{(1)}x_{(2)} - x_{(2)}, x_{(1)}^2)^T$ with starting points $\vec{x}_0 = (0.60, 0.60)^T$, $\vec{x}_0 = (0.60, 0.65)^T$ and $\vec{x}_0 = (0.65, 0.60)^T$.

Table 3.19: Starting points $\vec{x}_0$, total number of iterations done K, final approximations $\vec{x}_K$, and final errors $||\vec{x}_K - \vec{x}_*||_2 = ||\vec{x}_K - (0,0)^T||_2$ for the Picard iteration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-x_{(1)}x_{(2)} - x_{(2)}, x_{(1)}^2)^T$.

$\vec{x}_0$	K	$\vec{x}_K$	$\|\|\vec{x}_K - (0,0)^T\|\|_2$
$(0.60, 0.60)^T$	17	$(-4.51436\text{e}{-6}, 0.00000\text{e}{+0})^T$	$4.51436\text{e}{-6}$
$(0.60, 0.65)^T$	26	$(0.00000\text{e}{+0}, 9.72942\text{e}{+164})^T$	$9.72942\text{e}{+164}$
$(0.65, 0.60)^T$	21	$(-3.88093\text{e}{-6}, 0.00000\text{e}{+0})^T$	$3.88093\text{e}{-6}$

3.5.3. Convergence of Anderson acceleration for vector-valued functions

To start with, Anderson acceleration is applied to problem (3.23) with solution $\vec{x}_* = (-1, -1)^T$. The components only depend on one variable, making it component-wise equivalent to problem (3.9). In Figure 3.20 and Table 3.20, the results are given for the starting points $\vec{x}_0 = (0,0)^T$, $\vec{x}_0 = (1\text{e}{+5}, 1\text{e}{+5})^T$ and $\vec{x}_0 = (0, 1\text{e}{+5})^T$.

The first two plots, in which the elements of the starting points are the same, are similar to the scalar case (Figure 3.7). However, when the elements of $\vec{x}_0$ differ, the results are different, which is apparent from the last plot. The method still converges, but it is remarkable that the errors for $m = 1$ first start growing and then suddenly drop down.

For this problem, $\vec{G}$ grows faster than $\vec{x}$, meaning $||\vec{G}(\vec{x}_k) - \vec{x}_k||_2$ grows as the approximations move further away from $\vec{x}_* = (-1, -1)^T$. Therefore, $\gamma^{(k)}$ starts taking the value 1, as the term

$$\begin{aligned} ||\vec{f}_k - \mathcal{F}_k\gamma||_2 &= ||\vec{f}_k - (\vec{f}_k - \vec{f}_{k-1})\gamma||_2 \\ &= ||(1-\gamma)\vec{f}_k + \gamma\vec{f}_{k-1}||_2 \\ &= ||(1-\gamma)(\vec{G}(\vec{x}_k) - \vec{x}_k) - \gamma(\vec{G}(\vec{x}_{k-1}) - \vec{x}_{k-1})||_2 \end{aligned} \tag{3.25}$$

has to be minimised. Furthermore, when the difference between $\vec{G}(\vec{x}_k)$ and $\vec{G}(\vec{x}_{k-1})$ becomes very large, $\vec{G}(\vec{x}_k) - \vec{G}(\vec{x}_{k-1})$ is set to be $\vec{0}$, because the computer does not have enough precision to represent the actual difference (Goldberg, 1991). When this happens, the approximation becomes $\vec{0}$, because

$$\begin{aligned}\vec{x}_{k+1} &= \vec{G}(\vec{x}_k) - (\mathcal{X}_k + \mathcal{F}_k)\gamma^{(k)} \\ &= \vec{G}(\vec{x}_k) - (\vec{x}_k - \vec{x}_{k-1} + \vec{G}(\vec{x}_k) - \vec{x}_k - (\vec{G}(\vec{x}_{k-1}) - \vec{x}_{k-1})) \\ &= \vec{G}(\vec{x}_k) - (\vec{G}(\vec{x}_k) - G(\vec{x}_{k-1})) \\ &= \vec{G}(\vec{x}_k) - \vec{G}(\vec{x}_k) \\ &= \vec{0}.\end{aligned} \tag{3.26}$$

Once the approximations drop down, the method starts to converge, even though this should not be the case. These results imply that Anderson acceleration is more stable for bigger m, but can still accidentally converge for $m = 1$.

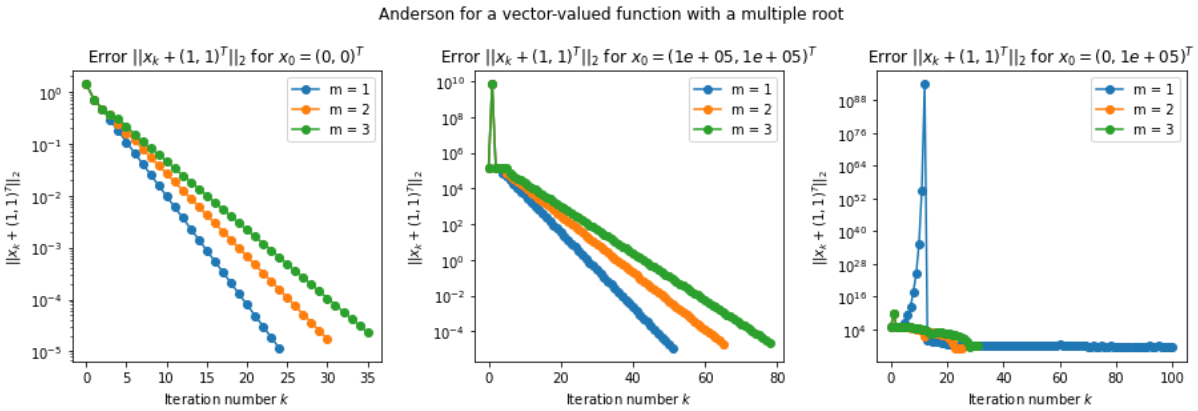

Figure 3.20: Error $||\vec{x}_k - \vec{x}_*||_2 = ||\vec{x}_k + (1,1)^T||_2$ per iteration k for Anderson acceleration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-0.5x_{(1)}^2 - 0.5, -0.5x_{(2)}^2 - 0.5)^T$ with starting points $\vec{x}_0 = (0.00, 0.00)^T$, $\vec{x}_0 = (1e{+}5, 1e{+}5)^T$ and $\vec{x}_0 = (0.00, 1e{+}5)^T$.

Table 3.20: Starting points $\vec{x}_0$, depth m, total number of iterations done K, final approximations $\vec{x}_K$, and final errors $||\vec{x}_K - \vec{x}_*||_2 = ||\vec{x}_K + (1,1)^T||_2$ for Anderson acceleration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-0.5x_{(1)}^2 - 0.5, -0.5x_{(2)}^2 - 0.5)^T$.

$\vec{x}_0$	m	K	$\vec{x}_K$	$\|\|\vec{x}_K + (1,1)^T\|\|_2$
$(0.00, 0.00)^T$	1	24	$(-9.99992e{-}1, -9.99992e{-}1)^T$	1.16499e−5
$(0.00, 0.00)^T$	2	30	$(-9.99988e{-}1, -9.99988e{-}1)^T$	1.73745e−5
$(0.00, 0.00)^T$	3	35	$(-9.99983e{-}1, -9.99983e{-}1)^T$	2.33748e−5
$(1e{+}5, 1e{+}5)^T$	1	51	$(-9.99992e{-}1, -9.99992e{-}1)^T$	1.17362e−5
$(1e{+}5, 1e{+}5)^T$	2	65	$(-9.99985e{-}1, -9.99985e{-}1)^T$	2.08480e−5
$(1e{+}5, 1e{+}5)^T$	3	78	$(-9.99983e{-}1, -9.99983e{-}1)^T$	2.41955e−5
$(0.00, 1e{+}5)^T$	1	100	$(-9.95556e{-}1, -1.00262e{+}0)^T$	5.16060e−3
$(0.00, 1e{+}5)^T$	2	25	$(-1.00003e{+}0, -1.00138e{+}0)^T$	1.38221e−3
$(0.00, 1e{+}5)^T$	3	31	$(-1.00019e{+}0, -1.00976e{+}0)^T$	9.76200e−3

Finally, Anderson acceleration is applied to problem (3.22). In Figure 3.21 and Table 3.21, it can be seen that for $m = 1$ the method starts diverging when $\vec{x}_0 = (1.45, 1.50)^T$ and bigger, but drops down to the solution $\vec{x}_* = (0, 0)^T$. This is what happened for the previous problem as well. For $m = 2$ and $m = 3$, the method starts to fail around $\vec{x}_0 = (1e5, 1e5)^T$.

When comparing these results to the previous subsection, it can be concluded that Anderson acceleration can converge for more choices of $\vec{x}_0$ compared to the Picard iteration when the problem is vector valued. However, convergence should theoretically not always be the case for $m = 1$.

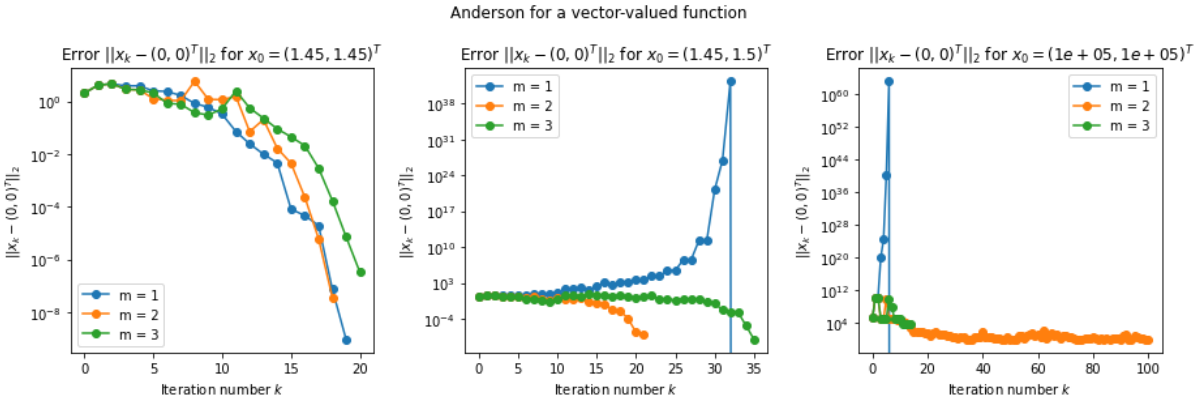

Figure 3.21: Error $||\vec{x}_k - \vec{x}_*||_2 = ||\vec{x}_k - (0,0)^T||_2$ per iteration k for Anderson acceleration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-x_{(1)}x_{(2)} - x_{(2)}, x_{(1)}^2)^T$ with starting points $\vec{x}_0 = (1.45, 1.45)^T$, $\vec{x}_0 = (1.45, 1.50)^T$ and $\vec{x}_0 = (1e{+}5, 1e{+}5)^T$.

Table 3.21: Starting points $\vec{x}_0$, depth m, total number of iterations done K, final approximations $\vec{x}_K$, and final errors $||\vec{x}_K - \vec{x}_*||_2 = ||\vec{x}_K - (0,0)^T||_2$ for Anderson acceleration applied to $\vec{G}(x_{(1)}, x_{(2)}) = (-x_{(1)}x_{(2)} - x_{(2)}, x_{(1)}^2)^T$.

$\vec{x}_0$	m	K	$\vec{x}_K$	$\|\|\vec{x}_K - (0,0)^T\|\|_2$
$(1.45, 1.45)^T$	1	19	$(9.26663e{-}10, -1.44760e{-}12)^T$	9.26664e−10
$(1.45, 1.45)^T$	2	18	$(-1.36212e{-}8, 3.42822e{-}8)^T$	3.68891e−8
$(1.45, 1.45)^T$	3	20	$(-1.33348e{-}7, 2.99942e{-}7)^T$	3.28248e−7
$(1.45, 1.50)^T$	1	33	$(0.00000e{+}0, 0.00000e{+}0)^T$	0.00000e+0
$(1.45, 1.50)^T$	2	21	$(8.07927e{-}9, -1.23152e{-}7)^T$	1.23417e−7
$(1.45, 1.50)^T$	3	35	$(-4.62998e{-}9, 8.72754e{-}9)^T$	9.87961e−9
$(1e{+}5, 1e{+}5)^T$	1	7	$(0.00000e{+}0, 0.00000e{+}0)^T$	0.00000e+0
$(1e{+}5, 1e{+}5)^T$	2	100	$(-5.88555e{-}1, -4.11610e{-}1)^T$	7.18206e−1
$(1e{+}5, 1e{+}5)^T$	3	14	$(-1.99823e{+}0, 4.11215e{+}3)^T$	4.11215e+3

3.6. Comparison of the algorithms

In this final section, the conclusions from the previous sections are summarised and the discussed methods are compared to each other.

To start with, the three algorithms perform the same for linear problems: the exact solution is found after one iteration.

For the Newton-Raphson method it is easy to say for which starting points x_0 the method converges to a certain solution and what the rate of convergence is when the function is 'nice', like quadratic functions. However, there are cases where Newton's method is very sensitive to the choice of x_0. The method can diverge or oscillate between values for certain starting points.

Banach's theorem gives certain starting points for which the Picard iteration will converge. However, the problem has to be a contraction in order to apply this theorem, which can lead to strict bounds. Although

the method may convergence when x_0 is not given by Banach's theorem, there is no certainty that this is the case. There are situations where the Picard iteration does not converge to the solution for any starting point and is outperformed by Newton's method. Furthermore, the rate of convergence is usually linear, whereas Newton's method often has quadratic convergence.

Furthermore, Anderson acceleration can converge for choices of x_0 for which the Picard iteration and Newton's method do not. This is a big advantage of this method, as the choice of the starting point is less restrictive. However, when there are multiple solutions, it is hard to say which solution the method will converge to. Besides this, the method can improve the convergence rate of Picard, even though it is still linear.

Finally, the performance of the methods for vector-valued functions is similar to that for scalar problems. However, Newton's method, the Picard iteration and Anderson acceleration with $m = 1$ are quite sensitive to differences in the elements of $\vec{x}_0$.

Although there is not one method that is strictly the best in every situation, Anderson acceleration can converge for more choices of $\vec{x}_0$ than the other methods, making it more stable. The converge rates can also be better than those of the Picard iteration. This implies it might be preferable to use this method when not much is known about the problem and its solution.

4

Root-finding algorithms applied to the incompressible Navier-Stokes equations

In this chapter, Newton's method, the Picard iteration and Anderson acceleration are applied to the incompressible Navier-Stokes equations (1.1). In each iteration of these algorithms, the problem is solved using a Nutils implementation of the finite element method. This method, and the parameters it is influenced by, is introduced in Section 4.1. In Section 4.2, the parameters, stopping criteria and collected data of the implementations are discussed. The resulting convergence rates are examined in Section 4.3, followed by the solutions in Section 4.4 and the timings in Section 4.5. Finally, in Section 4.6, the methods are compared to each other.

4.1. The finite element method

With the finite element method, the domain Ω of a problem is divided into elements e_i. The solution $\vec{u}$ is then approximated by a linear combination $\vec{\tilde{u}}$ of n functions $\vec{\phi}_j$ that are defined on these elements (Lyu, 2022), so

$$\vec{\tilde{u}}(\vec{x}) = \sum_{j=0}^{n} a_j \vec{\phi}_j(\vec{x}). \tag{4.1}$$

The functions $\vec{\phi}_j$ are called basis functions and are usually piecewise polynomials of degree at most p (Ern and Guermond, 2021), meaning the approximation is in the finite-dimensional space

$$S_n := \{f \in C^k(\bar{\Omega}) : f|_{e_i} \in \mathbb{P}_p, i = 1, \dots, n\}. \tag{4.2}$$

Since the basis functions are known, the problem has to be solved for the coefficients a_i.

As an example, the interval $[0, 1]$ is divided into four elements $e_1, \dots, e_4$ in Figure 4.1. The nodes $x_0, \dots, x_4$ are equally spaced.

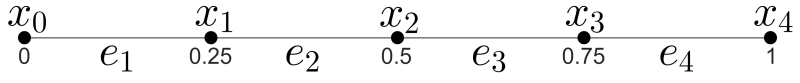

Figure 4.1: The interval $[0, 1]$ divided into four equally large elements.

One way to define the basis functions is by B-splines (De Boor, 1972). These splines are used in the Nutils implementation (Van Zwieten et al., 2022) and can have different orders. In Figure 4.2, a basis of order one is given. The functions ϕ_i are only non-zero on one or two elements. A basis of order two is displayed in Figure 4.3. Here, the functions are non-zero on one, two or three elements, so there is one more basis function compared to the linear case. As the amount of basis functions grows, the number of coefficients a_j to solve for also grows. Therefore, the problem gets bigger as the number of elements or the degree of the basis functions grows.

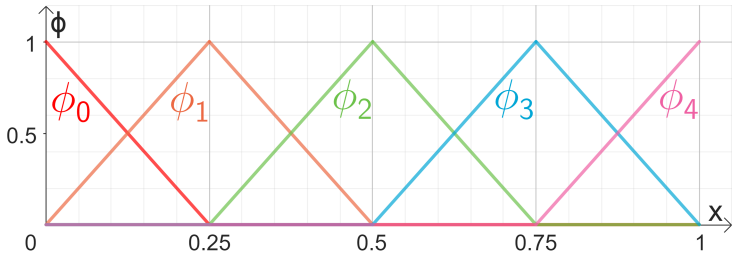

Figure 4.2: Linear spline basis functions ϕ_j with $j = 0, \dots, 4$ for four elements.

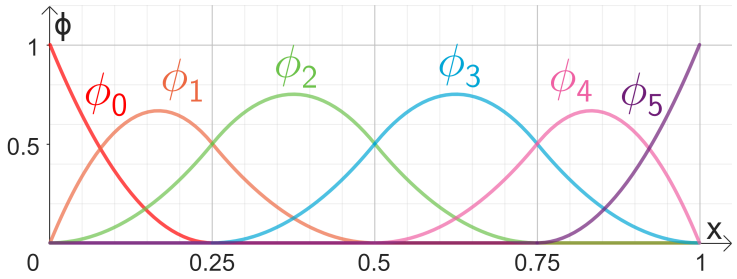

Figure 4.3: Quadratic spline basis functions ϕ_j with $j = 0, \dots, 5$ for four elements.

In Figure 4.4, an example of a linear combination of linear basis splines with coefficients $a = (0.5, 1.5, 1.75, 1.25, 1)$ is shown. The example in Figure 4.5 is a linear combination of quadratic splines with coefficients $a = (0.5, 1.5, 1.75, 1.25, 1, 1)$. The combination of quadratic splines is smoother than the combination of linear splines. Furthermore, the values of the linear example in the nodes are equal to the coefficients, because the basis functions are equal to one in the nodes. For the quadratic example, this is only the case in the first and last node.

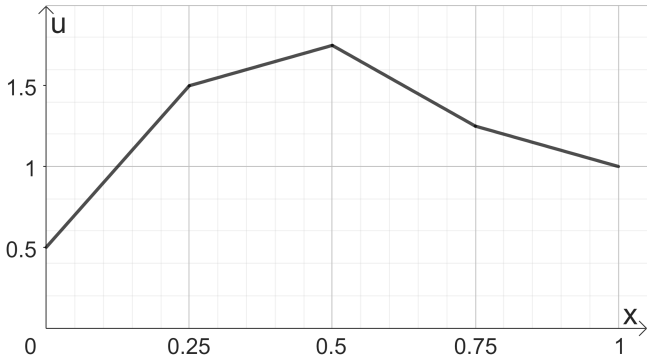

Figure 4.4: Example function u for four elements with $a = (0.5, 1.5, 1.75, 1.25, 1)$ and linear spline basis functions ϕ_j with $j = 0, \dots, 4$.

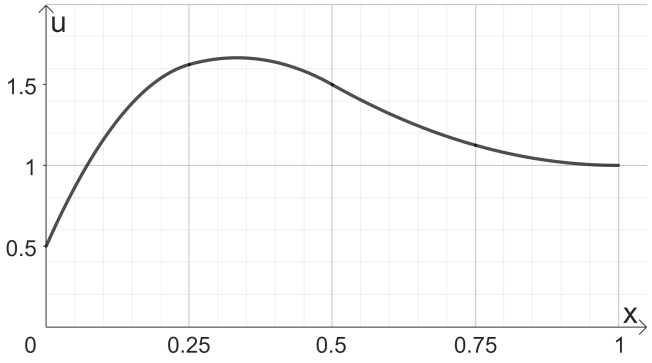

Figure 4.5: Example function u for four elements with $a = (0.5, 1.5, 1.75, 1.25, 1, 1)$ and quadratic spline basis functions ϕ_j with $j = 0, \dots, 5$.

4.2. Implementations

The three root-finding algorithms are applied to the incompressible Navier-Stokes equations (1.1) for the following parameters:

- $\mathrm{Re} = 5\mathrm{e}{+2}, 1\mathrm{e}{+3}, 1.5\mathrm{e}{+3}, 2\mathrm{e}{+3}, 2.5\mathrm{e}{+3}, 5\mathrm{e}{+3}$;
- number of elements in each dimension $= 16, 32, 64$;
- degree of basis functions $= 2, 3, 4$;
- depth m of Anderson $= 1, 2, 3, 4$.

The starting point is chosen to be the zero vector in all situations.

The stopping criteria for the iterations are similar to Chapter 3, namely:

- the residual, $||\vec{f}(\vec{x}_k)||_2$ for Newton or $||\vec{G}(\vec{x}_k) - \vec{x}_k||_2$ for Picard and Anderson, is smaller than tolerance $\delta = 1\mathrm{e}{-10}$;
- the difference between successive approximations, $||\vec{x}_{k+1} - \vec{x}_k||_2$, is smaller than $\epsilon = 1\mathrm{e}{-10}$;
- k exceeds $N = 250$;
- $||\vec{x}_k||_2$ exceeds $L = 1\mathrm{e}{+100}$.

In order to examine the performance of the methods for different combinations of parameters, the following data is collected:

- the errors per iteration;
- the final approximations;
- the timings of solving the linear system in Anderson acceleration;
- the timings of all extra steps that Anderson acceleration takes;
- the timings of Newton-Raphson.

The problems can get quite big, especially when the number of elements, and therefore the number of basis functions, increases. Besides this, there are many combinations of parameters. Therefore, the implementations are run on the DelftBlue high-performance computer (2024). The codes and outputs can be found in https://gitlab.tudelft.nl/kwdijkstra/bsc-jasmijn. The results are discussed in the following sections.

4.3. Convergence rates

In this section, the convergence rates of the methods are studied.

In Figure 4.6, 4.7 and 4.8, the errors per iteration are displayed for basis functions of degree 3 and all number of elements for $\mathrm{Re} = 5\mathrm{e}{+2}, 2\mathrm{e}{+3}$ and $5\mathrm{e}{+3}$ respectively. Figures corresponding to other basis functions and Reynolds numbers can be found in Appendix A.1, A.2 and A.3. Note that the iterations continue after the error has passed tolerance $\delta = 1\mathrm{e}{-10}$. This may be because these are the errors given by the Nutils implementation, and not the residuals $||\vec{f}(\vec{x}_k)||_2$ or $||\vec{G}(\vec{x}_k) - \vec{x}_k||_2$.

The first noticeable thing in these figures is that Newton's method converges a lot faster than the other methods for Reynolds numbers below $2\mathrm{e}{+3}$. However, for higher values of Re the method does not converge anymore. The Picard iteration does converge for higher Reynolds numbers, but can also fail for certain choices of the number of elements. Besides this, the convergence rate can become very slow, as is apparent from the middle plot of Figure 4.8. This is less so the case for Anderson acceleration, which has much better convergence rates in this same plot and the plot on the right.

Furthermore, there are instances where Picard and Anderson start oscillating, especially for large Reynolds numbers. The effect of this lessens as the depth of Anderson increases. For instance, in the left plot of Figure 4.8 it is visible that the error for $m = 4$ overall still decreases, and this eventually seems to happen for $m = 3$ as well, whereas this is not the case for $m = 1$ and $m = 2$. This implies that increasing the depth can make the method more stable.

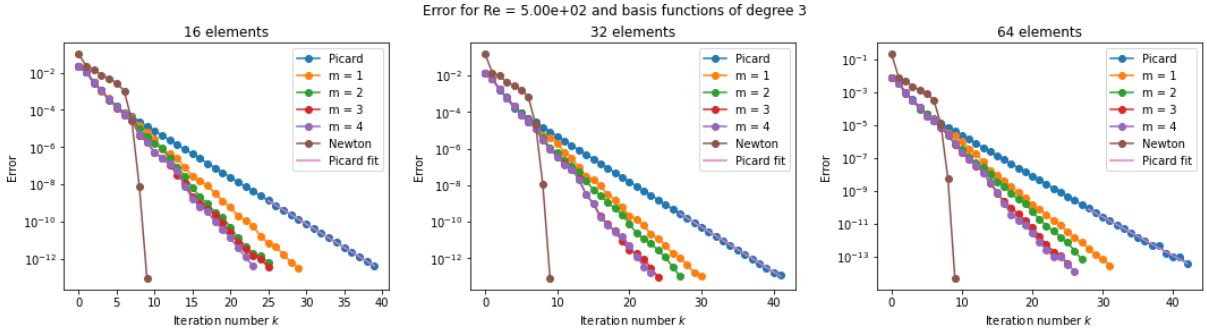

Figure 4.6: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+2}$ and basis functions of degree 3.

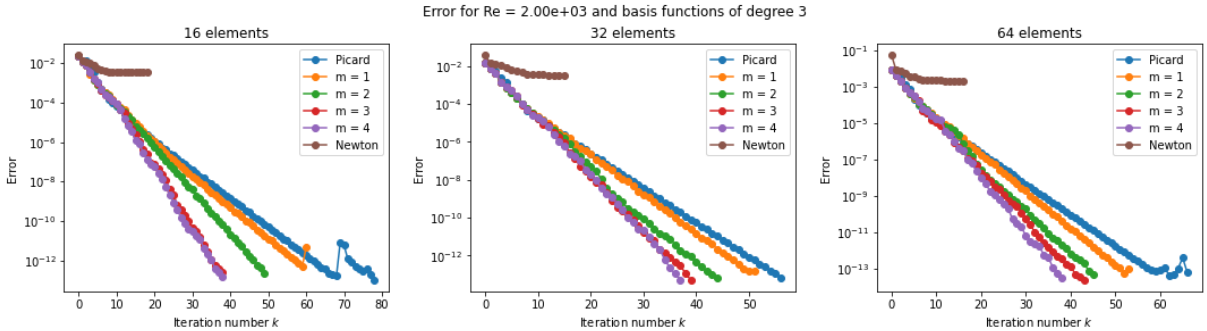

Figure 4.7: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and basis functions of degree 3.

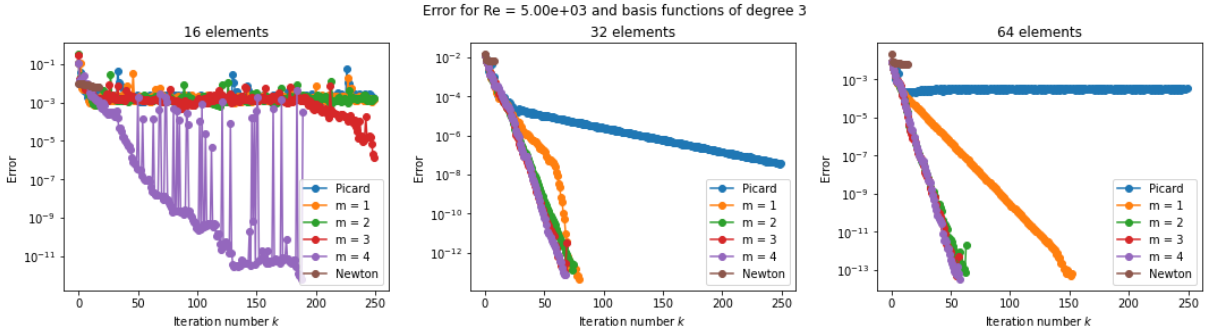

Figure 4.8: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+3}$ and basis functions of degree 3.

In order to examine the convergence rates of Picard and Anderson more easily, the slopes of the plots are estimated using linear regression on the last fifteen iterations. In Figure 4.6, the fitted lines are shown for Picard as an example. The absolute values of these slopes are shown in Figure 4.9, 4.10 and 4.11 for basis functions of degree 2, 3 and 4 respectively. Instances where the slope is positive are not shown, as the method is then diverging.

In these figures, it can be seen that the convergence rates overall decrease as the Reynolds number increases. It can also happen that a method does not converge for certain values of Re. Furthermore, Anderson acceleration generally has better convergence rates than Picard, and increasing the depth can improve the rates as well. There are also cases where Picard does not converge, but Anderson does. These results are in line with those found in Chapter 3. Note, however, that only the final fifteen iterations are considered and that there can be oscillations in the errors. This may cause the results to be flawed.

There is not one optimal method and combination of parameters that follows from these findings. Generally, Newton's method might be the best choice for small Reynolds numbers and Anderson acceleration with a depth of 3 or 4 for large Reynolds numbers.

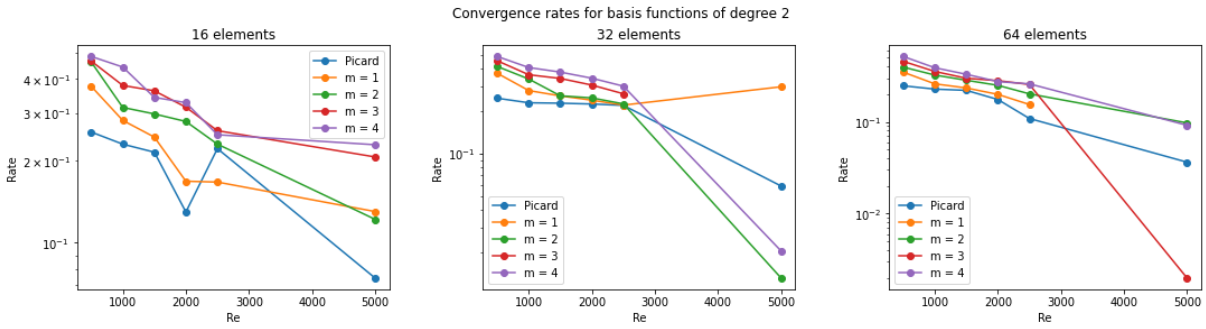

Figure 4.9: Convergence rates per Reynolds number for the Navier-Stokes equations with basis functions of degree 2.

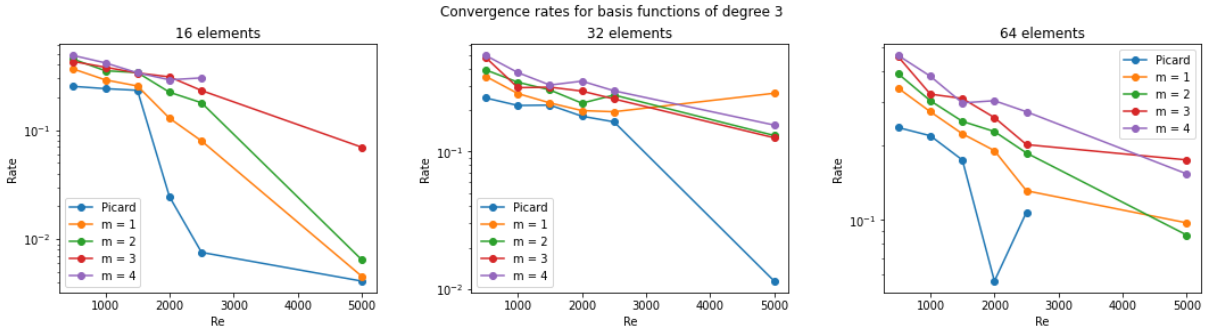

Figure 4.10: Convergence rates per Reynolds number for the Navier-Stokes equations with basis functions of degree 3.

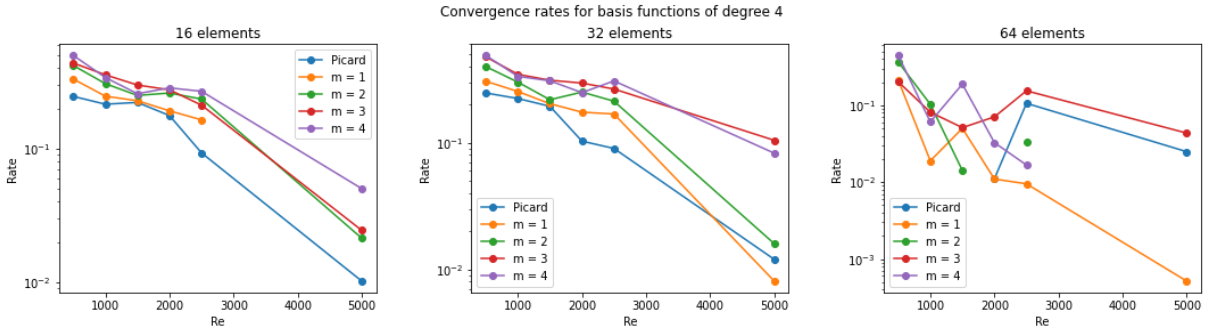

Figure 4.11: Convergence rates per Reynolds number for the Navier-Stokes equations with basis functions of degree 4.

4.4. Solutions

In this section, it is examined how the solutions are affected by different choices of the parameters Re, number of elements, and degree of the basis functions.

As mentioned in the introduction, the solutions contain a circular motion due to the horizontal movement on the top boundary. How the velocity changes over space is dependent on the Reynolds number. Since fluids with low Reynolds numbers are relatively viscous, the velocity gradually changes over the domain. The convective term plays a bigger role for fluids with high Reynolds numbers, resulting in the velocity changing more quickly.

Besides this, at high Reynolds numbers, the term $\Delta\vec{u}(\vec{x},t)$ does not have much importance inside of the domain. Since this is the highest order derivative, one of the boundary conditions is 'neglected' on the interior (Veldman, 2012). However, the term gains importance in a thin boundary layer close to the solid boundaries. Therefore, the velocity can be high close to the boundaries and drop down to zero in this small layer in order to satisfy the boundary conditions (Mestel, n.d.).

These behaviours are visible in Figure 4.12, showing the solutions for 32 elements with basis functions of degree 3 for $\mathrm{Re} = 5\mathrm{e}{+2}, 2\mathrm{e}{+3}$ and $5\mathrm{e}{+3}$. Dark parts indicate high velocity and lighter parts indicate lower velocity. These solutions are given by Anderson acceleration with depth 4, which converges for all these situations.

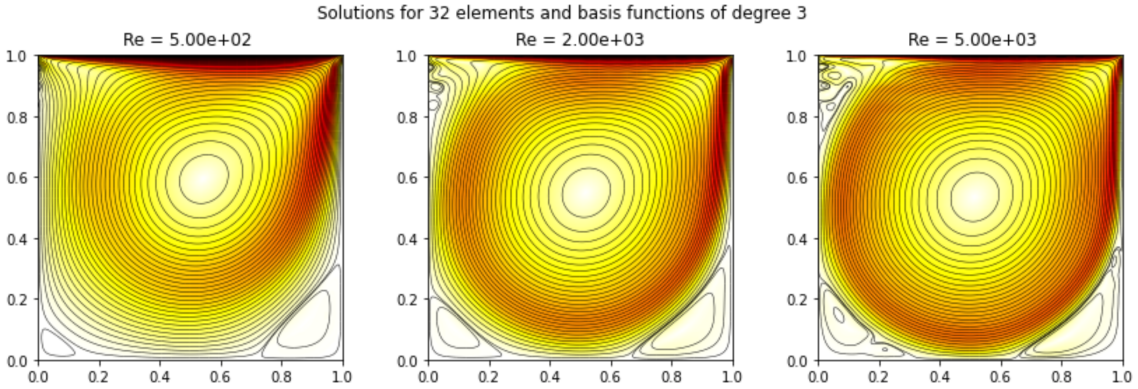

Figure 4.12: Solutions to the Navier-Stokes equations with 32 elements and basis functions of degree 3.

In Figure 4.13, solutions for $\mathrm{Re} = 2\mathrm{e}{+3}$ with basis functions of degree 3 are shown for 16, 32 and 64 elements. The solutions become smoother as the number of elements grows. This is because the number of basis functions increases as well, allowing for more detail in the solutions. However, there is not much difference between 32 and 64 elements.

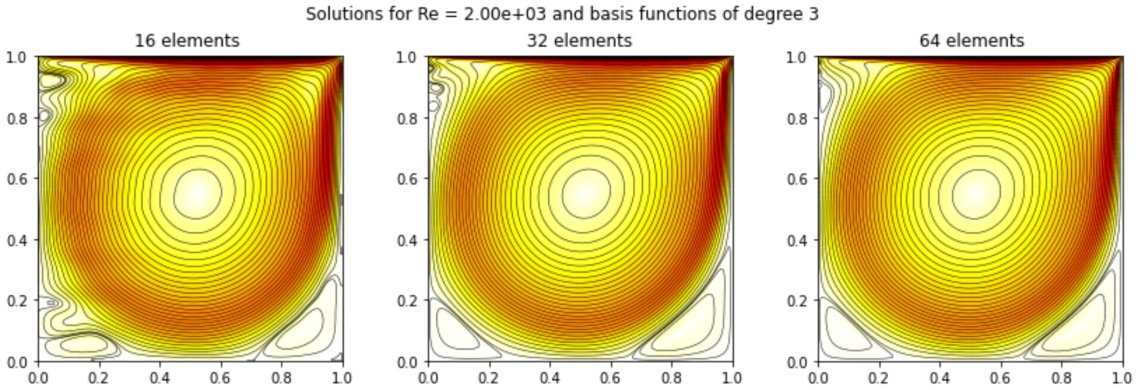

Figure 4.13: Solutions to the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and basis functions of degree 3.

Finally, Figure 4.14 contains solutions for $\mathrm{Re} = 2\mathrm{e}{+3}$ and 32 elements with basis functions of degree 2, 3 and 4. There do not seem to be many differences between these solutions. This indicates that changing the number of elements has more effect than changing the degree of the basis functions.

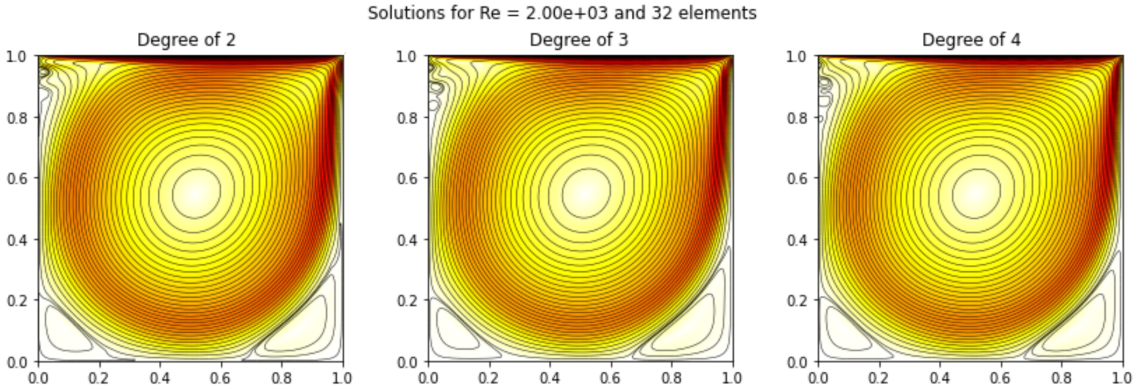

Figure 4.14: Solutions to the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and 32 elements.

4.5. Timings

As a final step, the amount of time the different methods take for different parameters is looked at in this section.

Firstly, it is examined how much more expensive Anderson acceleration is compared to the Picard iteration. The ratios between the amount of time the extra optimisation steps of Anderson take on average and the amount of time it takes to solve the linear system (so just a Picard iteration) on average are displayed in Figure 4.15. This is done for different depths, Reynolds numbers and numbers of elements with basis functions of degree 3. The results for basis functions of degree 2 and 4 are similar and can be found in Appendix A.4.

In the figure, it can be seen that Anderson acceleration becomes more expensive as the depth m grows, because then the least squares problem gets larger. However, the ratios remain of an order 10^{-3} or 10^{-4}, which means the extra steps barely take any time compared to the linear iteration. The ratios also do not differ much for different Reynolds numbers. This implies that, although Anderson acceleration gets more expensive as the depth increases, the difference with respect to the Picard iteration is small.

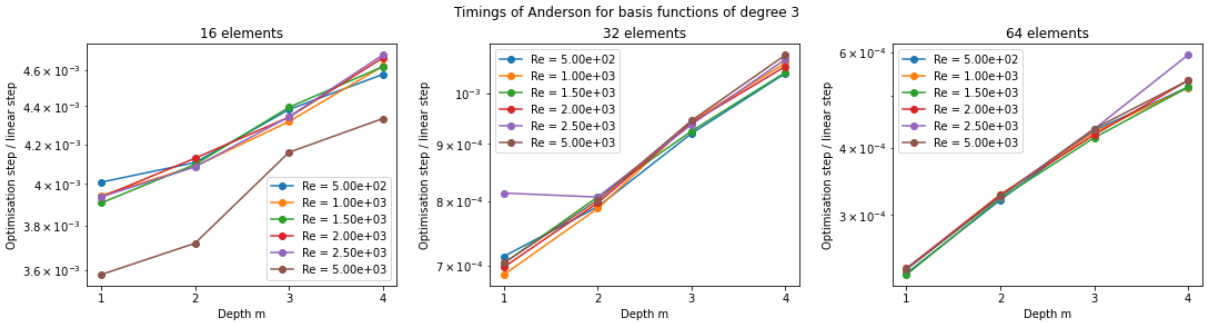

Figure 4.15: Ratio timings of linear iteration to timings of additional steps for Anderson per depth m for the Navier-Stokes equations with basis functions of degree 3.

In Figure 4.16, the bar plots indicate the average timings of Picard and Anderson for different values of Re and number of elements with basis functions of degree 3. Note that the extra time that Anderson takes is negligible on this scale. The boxplots correspond to the timings of Newton's method. A boxplot is chosen, because the timings can differ, since the Jacobian does not have to be computed in every iteration of the implementation. When a boxplot is not present, Newton's method does not converge.

The results for basis functions of degree 2 and 4 are similar and are given in Appendix A.5. The main difference is that the time increases as the degree increases, because the number of basis functions, and therefore the size of the system, grows.

From these figures, it can be concluded that Newton-Raphson usually takes longer per iteration compared to the other methods. This is expectable, as computing the Jacobian matrix is computationally expensive. There could be exceptions for 64 elements, but the medians of Newton are still higher than the average time of the linear iterations. Furthermore, all methods take longer when the number of elements is increased, because then the system gets bigger.

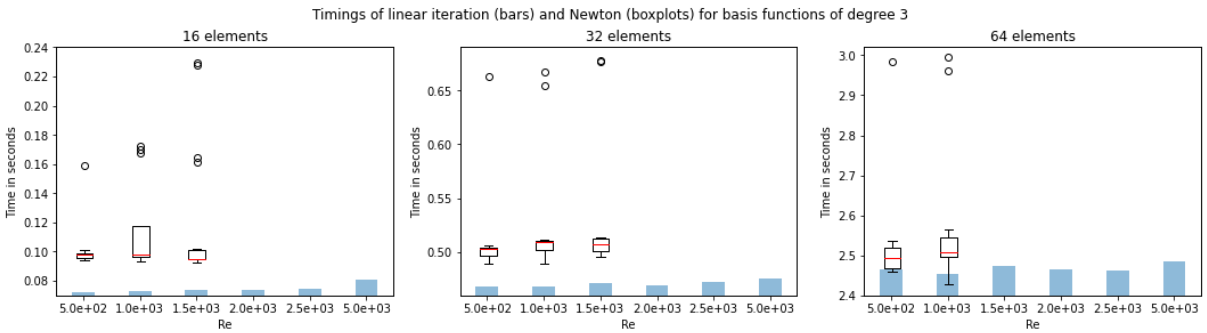

Figure 4.16: Average timings of linear iterations and timings of the iterations of Newton-Raphson per Reynolds number for the Navier-Stokes equations with basis functions of degree 3.

At last, the total times the methods take for different Reynolds numbers and number of elements are shown in Figure 4.17, 4.18 and 4.19 for basis functions of degree 2, 3 and 4 respectively. Similarly to the convergence rates, instances where the estimated slope is positive are not plotted.

Here, it can be concluded that Newton's method is the fastest method when it converges, with a small exception for 16 elements and degree 3. Although the method takes longer per iteration compared to Picard and Anderson, not many iterations are needed, because of the fast convergence. This results in the total time being lower. Besides this, Anderson acceleration generally takes less time as the depth increases, even though it was shown that the costs of the optimisation steps increase. This is because the convergence rates get better as the depth grows, and therefore the method needs less iterations. Thus, the cheapest choice would be Newton's method for small Reynolds numbers and Anderson acceleration with depth 3 or 4 for high Reynolds numbers.

Furthermore, the total time increases as the number of elements or the degree of the basis functions, and thus the size of the system, grows. It was already shown with the bar plots and boxplots that the time per iteration increases, and this carries over to the total time.

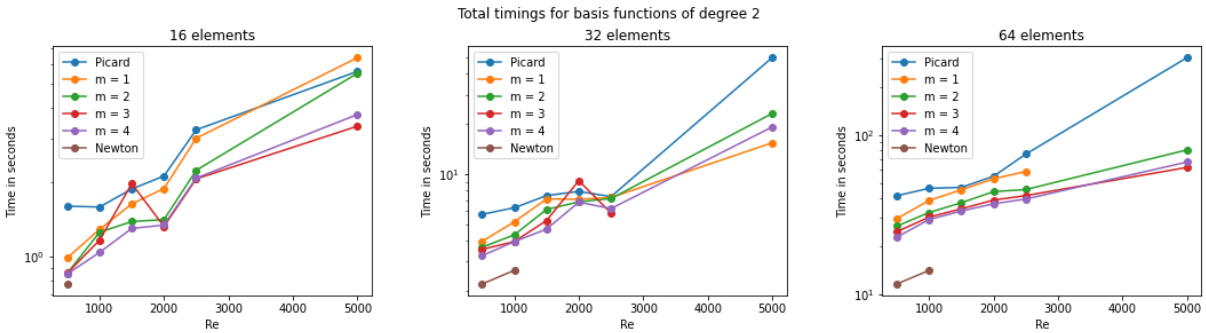

Figure 4.17: Total timings per Reynolds number for the Navier-Stokes equations with basis functions of degree 2.

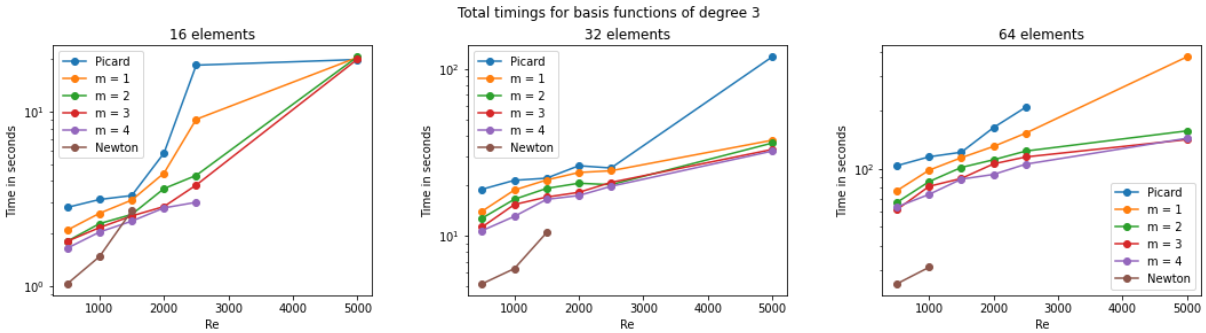

Figure 4.18: Total timings per Reynolds number for the Navier-Stokes equations with basis functions of degree 3.

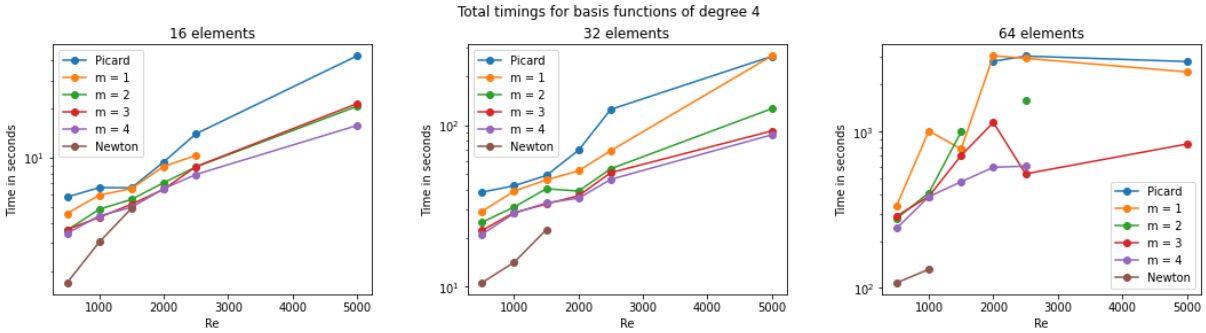

Figure 4.19: Total timings per Reynolds number for the Navier-Stokes equations with basis functions of degree 4.

4.6. Comparison of the algorithms

From the previous sections, the same conclusions can be drawn regarding Anderson acceleration as in Chapter 3. The method can converge in situations where the Newton-Raphson method and the Picard iteration do not, which is the case for high Reynolds numbers. Furthermore, Anderson can increase the convergence rate of Picard.

There is not one depth m of Anderson acceleration that follows as the optimal choice. Generally, $m = 3$ and $m = 4$ have better convergence rates and are more stable than $m = 1$ and $m = 2$, as they are less affected by oscillations and often still converge overall. It is also not that much more computationally expensive per iteration to choose a higher value of m. In total, this actually takes less time, as less iterations are needed due to the faster convergence.

For low Reynolds numbers, the optimal choice would be Newton's method. Although this method is more expensive than the other methods per iteration, it has better convergence rates and therefore needs less iterations and in turn less time in total. The method might still be useful for high Reynolds numbers. As shown in Chapter 3, Newton's method is very sensitive to the starting point. Therefore, convergence may occur when a better starting point is chosen. This can be done by doing a few iterations with Anderson acceleration to get closer to the solution, and using the approximation that follows from this as a starting point for Newton.

Finally, for this particular problem, choosing 32 or 64 elements results in solutions that are visibly smoother than when choosing 16 elements, but 64 elements leads to much more computational time. The degree of the basis functions does not have much effect on the solutions, but can affect the performance of the methods. For instance, increasing the degree to from 3 to 4 can cause a lot of oscillations for 64 elements. The systems, and thus the computational time, also grow when the degree increases. These conclusions, however, may not hold for different problems and thus no recommendations are made regarding the parameters of the finite element method.

5

Conclusion and discussion

The aim of this paper was to implement Anderson acceleration for the incompressible Navier-Stokes benchmark problem and compare its performance to the Picard iteration and Newton's method. This was done by first implementing the algorithms for scalar-valued and vector-valued problems, and then extending them to the Navier-Stokes equations. In this final chapter, the main conclusions are discussed.

To start with, for scalar-valued problems, vector-valued problems and the Navier-Stokes equations, Anderson acceleration can convergence in situations where the Picard iteration and Newton's method do not. Therefore, it can be concluded that Anderson acceleration is more stable than the other methods. Besides this, the convergence rates of Picard can be improved by Anderson, which is especially noticeable for high Reynolds numbers.

There is not one optimal depth m of Anderson acceleration that follows from the experiments. Overall, the convergence rates are better and the method is more stable when $m = 3$ and $m = 4$ compared to $m = 1$ and $m = 2$. These higher values also take less computational time, as the better convergence rates result in less iterations that are needed.

Despite the long computational time that Newton's method needs per iteration, the method is the cheapest option for low Reynold's numbers. This is because it converges really quickly, and thus needs less iterations and in turn less total time to converge. In Chapter 3 it was shown that the method is very sensitive to the choice of the starting point. Therefore, different starting points may improve the performance of Newton for high Reynolds numbers. A better starting point could for instance be found by using the result of few iterations of Anderson acceleration. This is something that could be researched further.

Besides the different methods, different parameters of the finite element method were considered. For this problem, it followed that dividing the domain in 32 or 64 elements in each direction results in solutions that are visibly smoother than when choosing 16 elements. However, much more computational time is needed for 64 elements. For a high number of elements, oscillations may occur in the errors when the degree of the basis functions is 4. These conclusions, however, may not hold for different problems and thus no recommendations are made regarding the parameters of the finite element method.

Finally, it should be noted that the convergence rates were calculated as the slope of the last fifteen iterations of the methods, which contained oscillations in some instances. Therefore, the results and the conclusions drawn from them may not be perfect.

Bibliography

Anderson, D. (1965). Iterative Procedures for Nonlinear Integral Equations. *Journal of the ACM*, *12*(4), 547–560. https://doi-org.tudelft.idm.oclc.org/10.1145/321296.321305

Berinde, V. (2007). *Iterative Approximation of Fixed Points* (Second edition). Springer.

Boyer, F., & Fabrie, P. (2013). *Mathematical Tools for the Study of the Incompressible Navier-Stokes Equations and Related Models*. Springer.

De Boor, C. (1972). On Calculating with B-Splines. *Journal of Approximation Theory*, *6*(1), 50–62. 10.1016/0021-9045(72)90080-9

Dennis, J., & Schnabel, R. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM edition). Prentice-Hall, Inc.

DHPC. (2024). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2

Ern, A., & Guermond, J.-L. (2021). *Finite Elements I: Approximation and Interpolation*. Springer.

Goldberg, D. (1991). What Every Computer Scientist Should Know About Floating-Point Arithmetic. *ACM Computing Surveys*, *23*(1), 5–48. https://doi.org/10.1145/103162.103163

Kamel, A., Haraz, E., & Hanna, S. (2020). Numerical simulation of three-sided lid-driven square cavity. *Wiley Engineering Reports*, *2*(4). https://doi.org/10.1002/eng2.12151

Kichigin. (2024). Researchers created a tornado-like vortex in superfluid helium. https://www.newscientist.com/article/2412676-scientists-created-a-giant-quantum-vortex-that-mimics-a-black-hole/

Lay, S. (2014). *Analysis with an Introduction to Proof* (Fifth edition). Pearson.

Łukaszewicz, G., & Kalita, P. (2016). *Navier-Stokes Equations: An Introduction with Applications*. Springer.

Lyu, Y. (2022). *Finite Element Method: Element Solutions*. Springer.

Mestel, J. (n.d.). M3A10 Viscous Flow: Boundary Layer Theory. https://www.ma.ic.ac.uk/~ajm8/M3A10/

Pollock, S., Rebholz, L., & Xiao, M. (2019). Anderson-Accelerated Convergence of Picard Iterations for Incompressible Navier-Stokes Equations. *SIAM*, *57*(2), 615–637. https://doi.org/10.1137/18M1206151

Van Kan, J., Segal, A., & Vermolen, F. (2014). *Numerical methods in Scientific Computing* (Second edition). TU Delft OPEN Publishing.

Van Zwieten, J., Van Zwieten, G., & Hoitinga, W. (2022). The Nutils Book. https://nutils.org/tutorial-bases.html

Veldman, A. (2012). *Boundary Layers in Fluid Dynamics*. University of Groningen.

Walker, H., & Ni, P. (2011). Anderson Acceleration for Fixed-point Iterations. *SIAM*, *49*(4), 1715–1735. https://doi.org/10.1137/10078356X

Younsi, R. (2012). *Navier-Stokes Equations: Properties, Description and Applications*. Nova Science Publishers, Inc.

A

Additional figures Chapter 4

This appendix contains figures that are additional to Chapter 4. In Appendix A.1, A.2 and A.3, the errors per iteration for the three algorithms with different parameters can be found for basis functions of degree 2, 3 and 4 respectively. Timings of the optimisation steps of Anderson compared to timings of solving the linear system are shown in Appendix A.4 for different parameters. The last section, A.5, contains timings of the linear iteration and Newton's method.

A.1. Errors for basis functions of degree 2

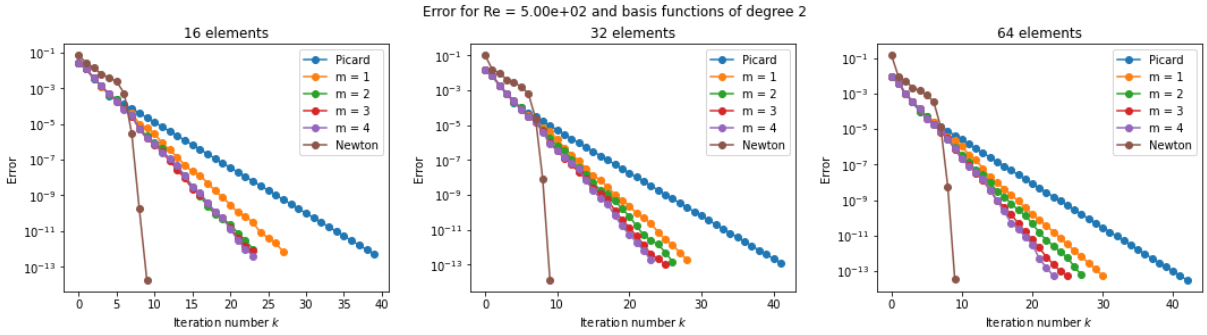

Figure A.1: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+2}$ and basis functions of degree 2.

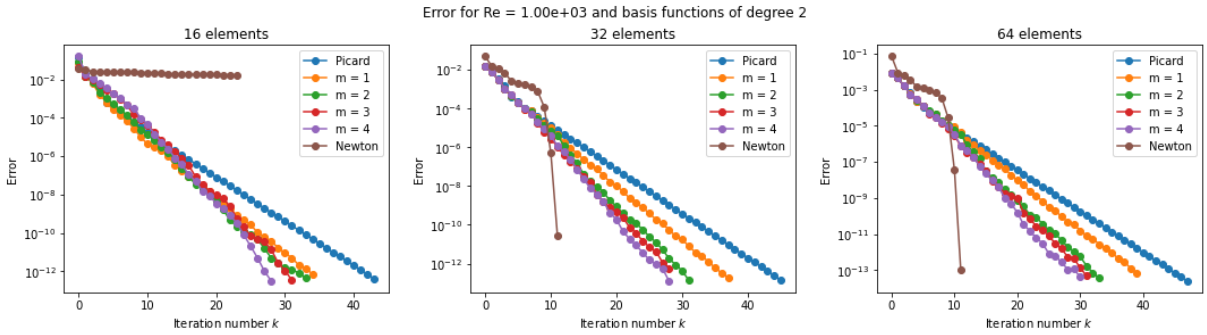

Figure A.2: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1\mathrm{e}{+3}$ and basis functions of degree 2.

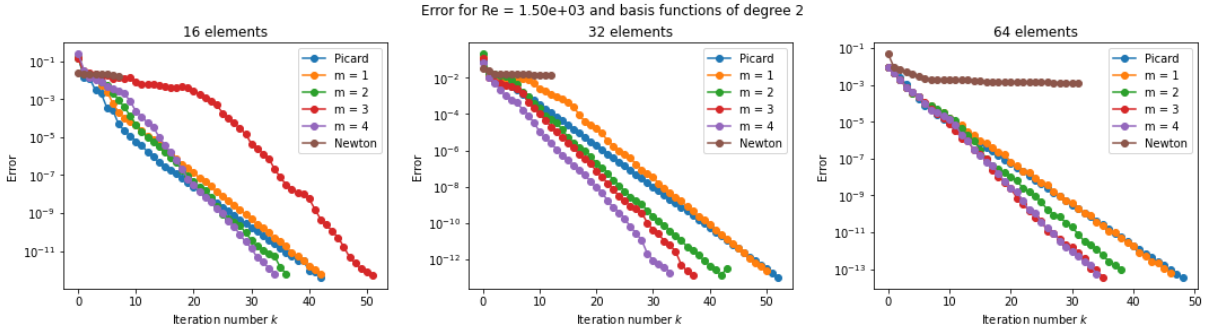

Figure A.3: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1.5\mathrm{e}{+3}$ and basis functions of degree 2.

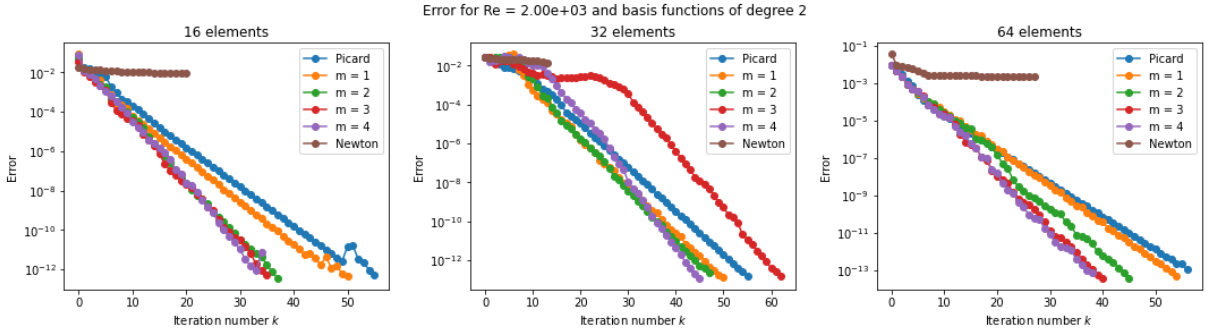

Figure A.4: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and basis functions of degree 2.

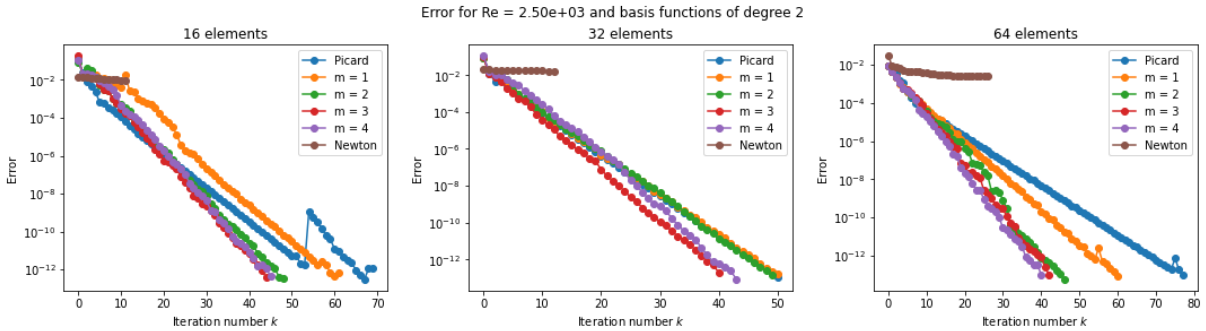

Figure A.5: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2.5\mathrm{e}{+3}$ and basis functions of degree 2.

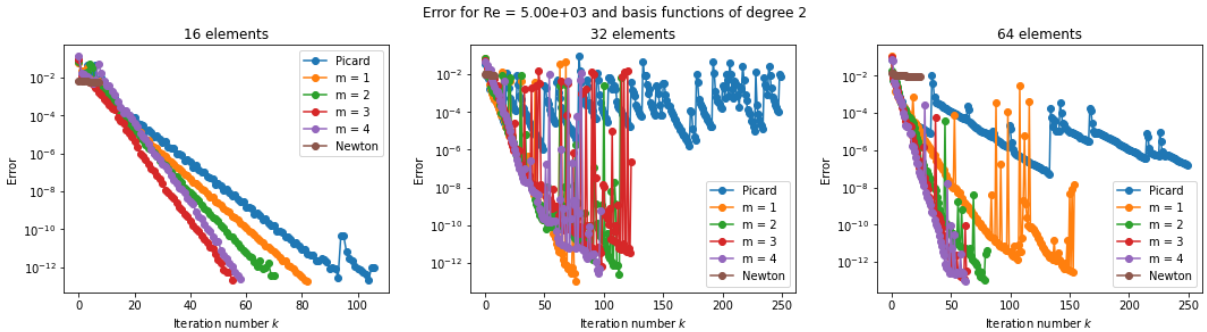

Figure A.6: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+3}$ and basis functions of degree 2.

A.2. Errors for basis functions of degree 3

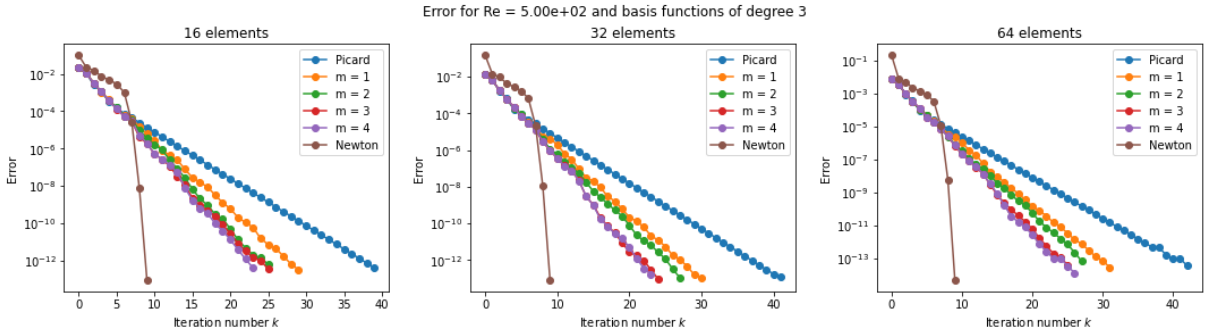

Figure A.7: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+2}$ and basis functions of degree 3.

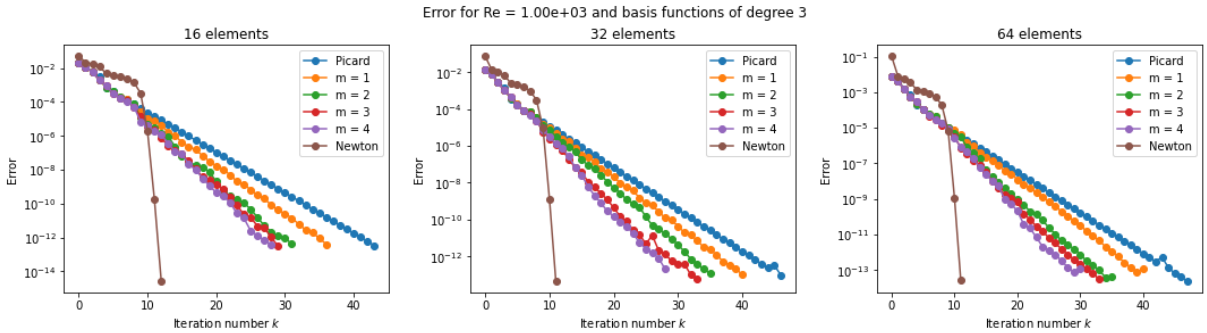

Figure A.8: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1\mathrm{e}{+3}$ and basis functions of degree 3.

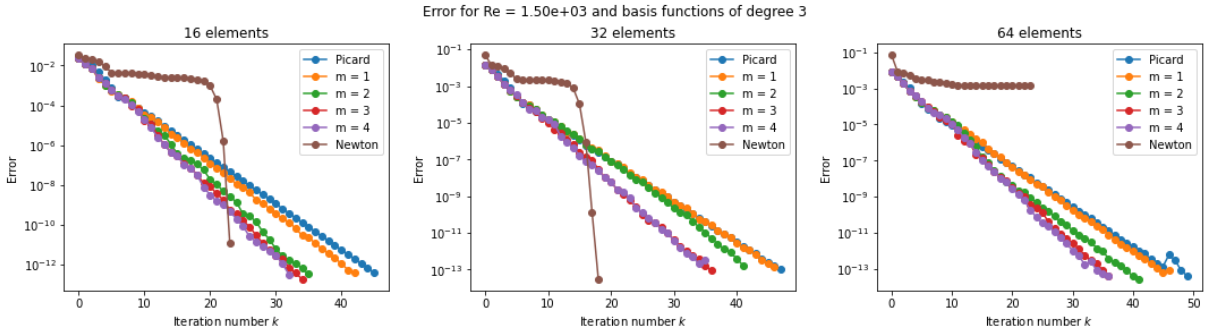

Figure A.9: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1.5\mathrm{e}{+3}$ and basis functions of degree 3.

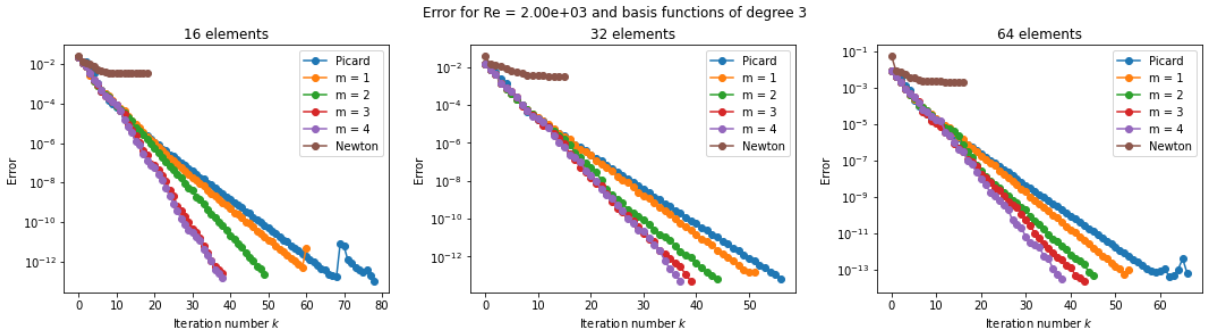

Figure A.10: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and basis functions of degree 3.

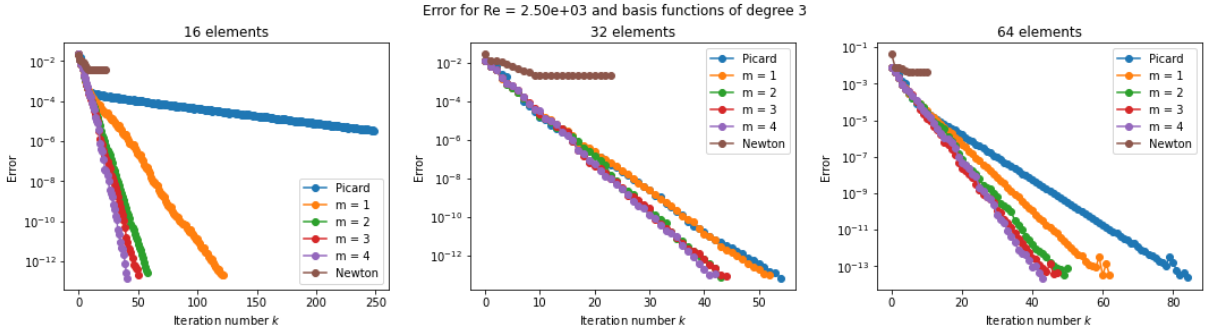

Figure A.11: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2.5\mathrm{e}{+3}$ and basis functions of degree 3.

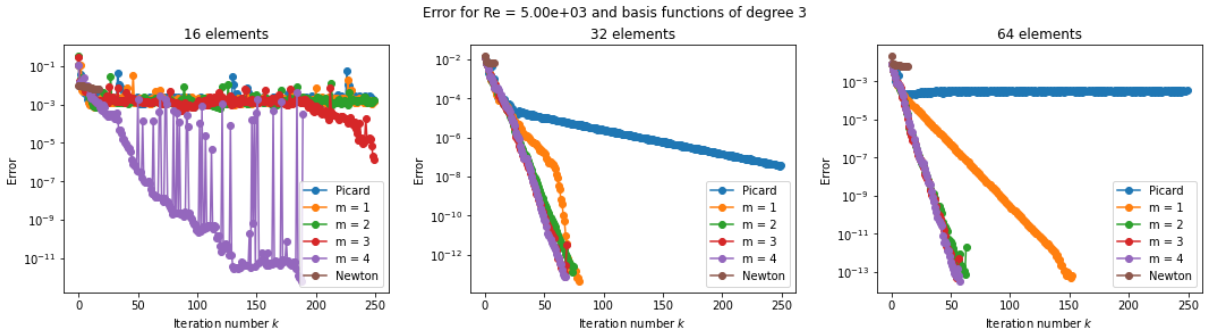

Figure A.12: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+3}$ and basis functions of degree 3.

A.3. Errors for basis functions of degree 4

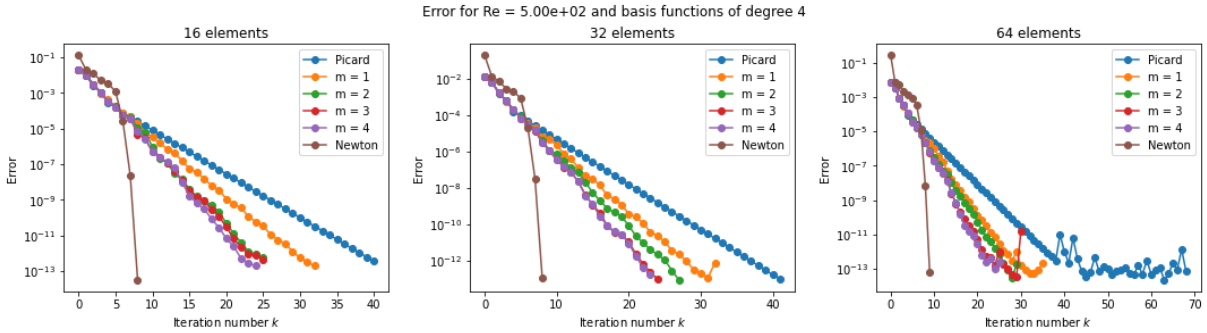

Figure A.13: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+2}$ and basis functions of degree 4.

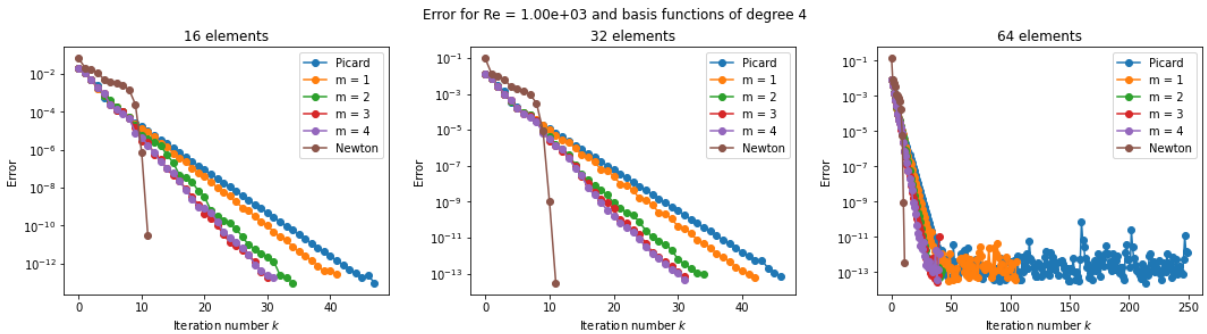

Figure A.14: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1\mathrm{e}{+3}$ and basis functions of degree 4.

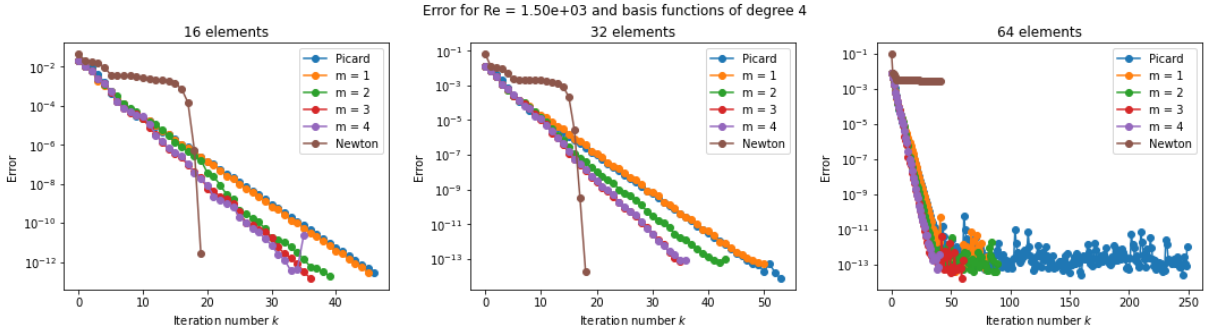

Figure A.15: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 1.5\mathrm{e}{+3}$ and basis functions of degree 4.

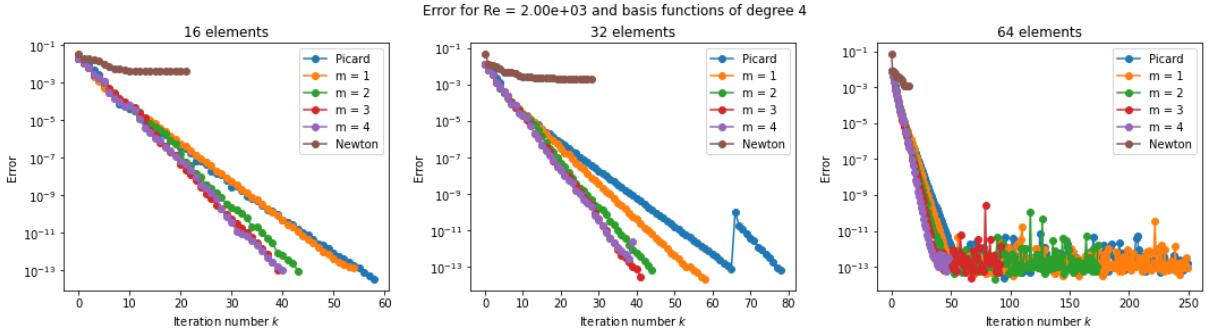

Figure A.16: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2\mathrm{e}{+3}$ and basis functions of degree 4.

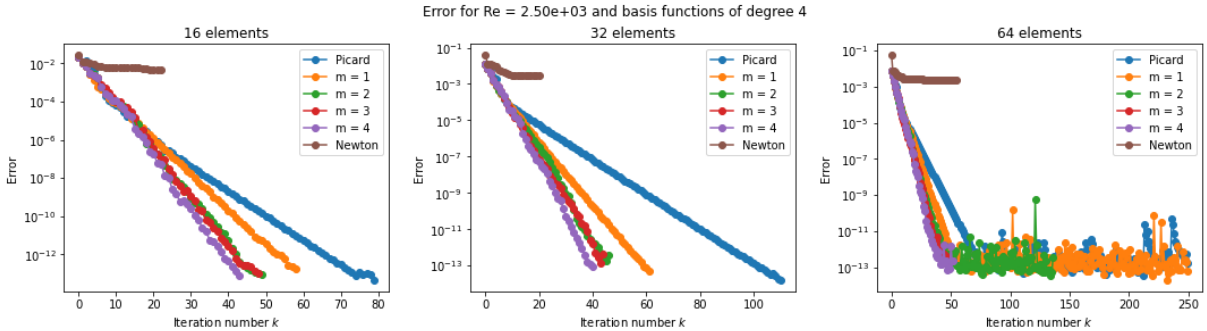

Figure A.17: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 2.5\mathrm{e}{+3}$ and basis functions of degree 4.

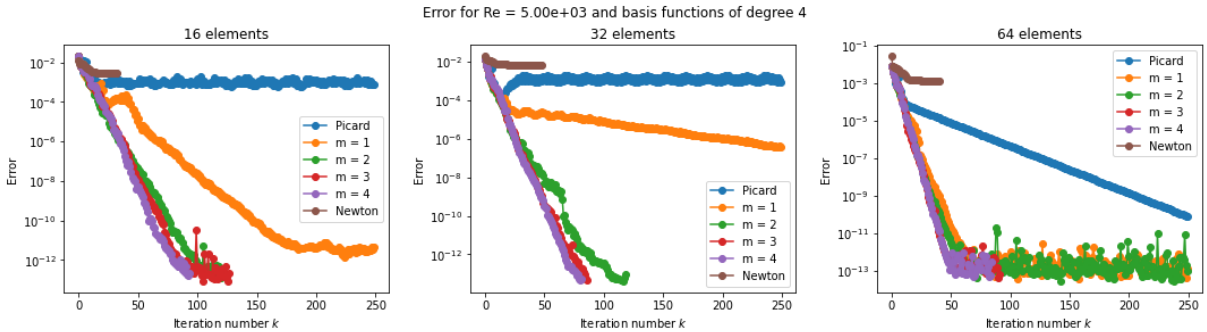

Figure A.18: Error per iteration k for the Navier-Stokes equations with $\mathrm{Re} = 5\mathrm{e}{+3}$ and basis functions of degree 4.

A.4. Timings of Anderson acceleration

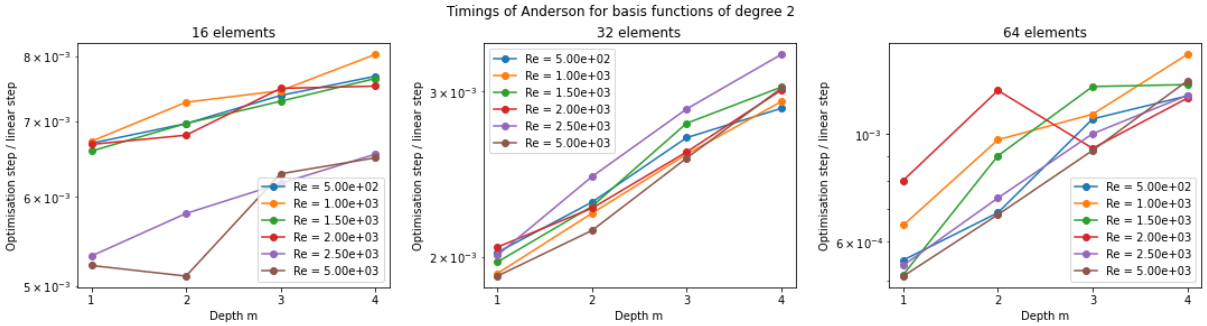

Figure A.19: Ratio timings of linear iteration to timings of additional steps for Anderson per Reynolds number for the Navier-Stokes equations with basis functions of degree 2.

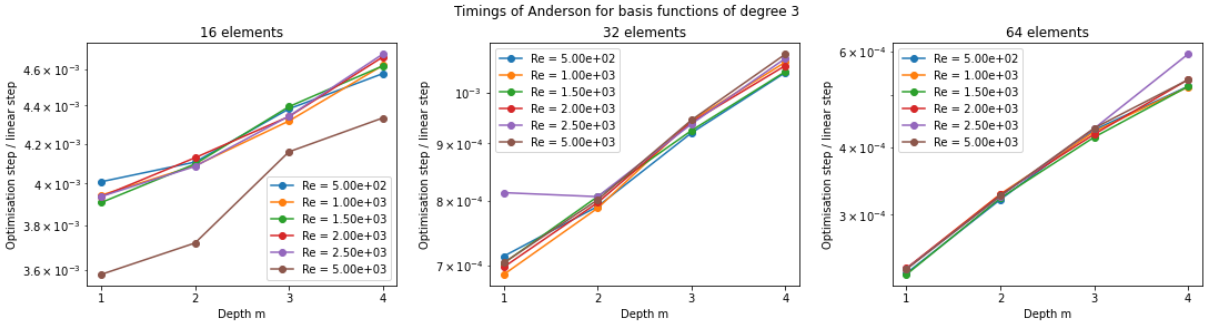

Figure A.20: Ratio timings of linear iteration to timings of additional steps for Anderson per Reynolds number for the Navier-Stokes equations with basis functions of degree 3.

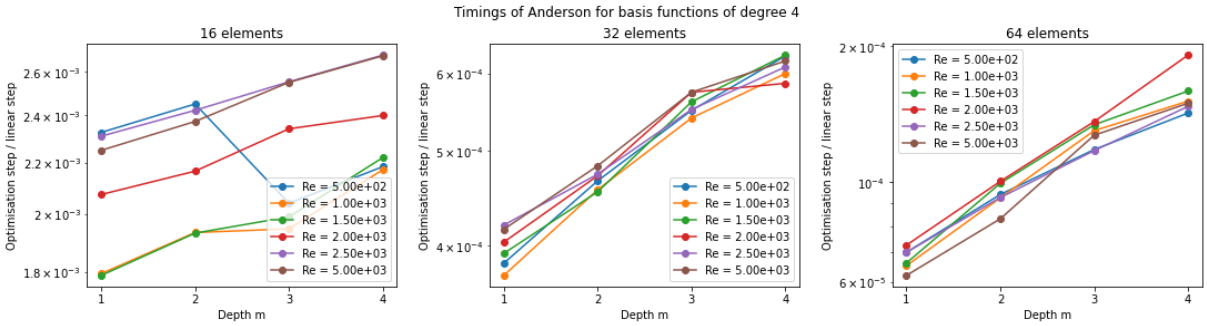

Figure A.21: Ratio timings of linear iteration to timings of additional steps for Anderson per Reynolds number for the Navier-Stokes equations with basis functions of degree 4.

A.5. Timings of linear iteration and Newton-Raphson

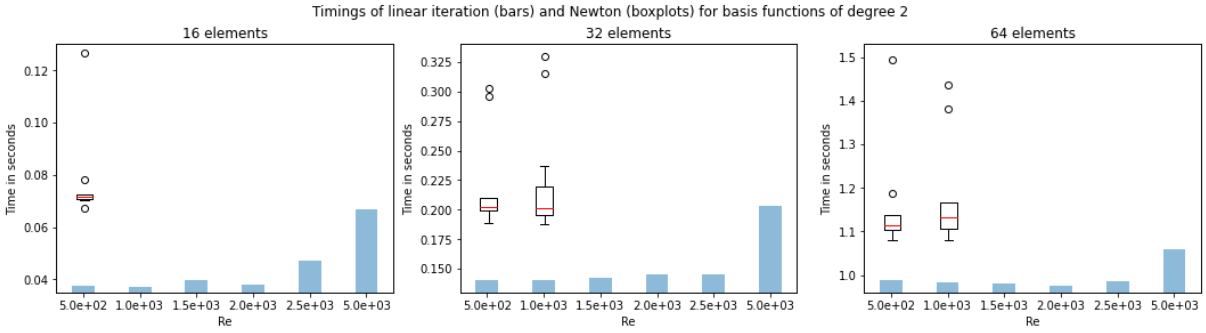

Figure A.22: Timings of linear iteration and timings of Newton-Raphson per Reynolds number for the Navier-Stokes equations with basis functions of degree 2.

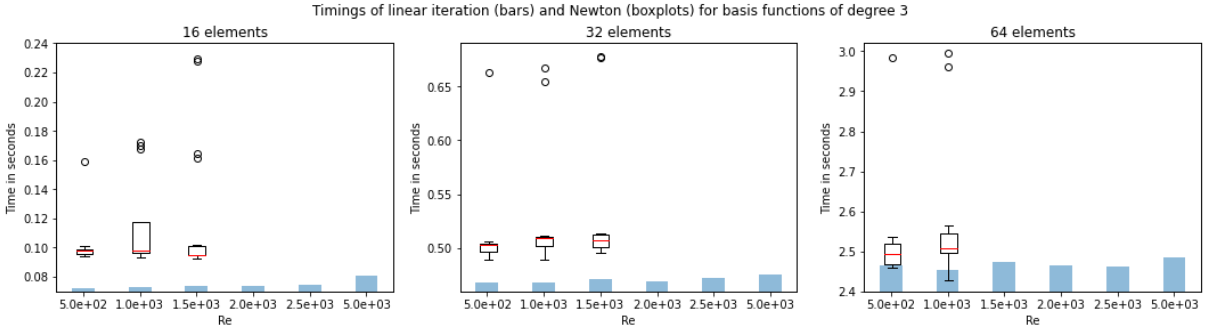

Figure A.23: Timings of linear iteration and timings of Newton-Raphson per Reynolds number for the Navier-Stokes equations with basis functions of degree 3.

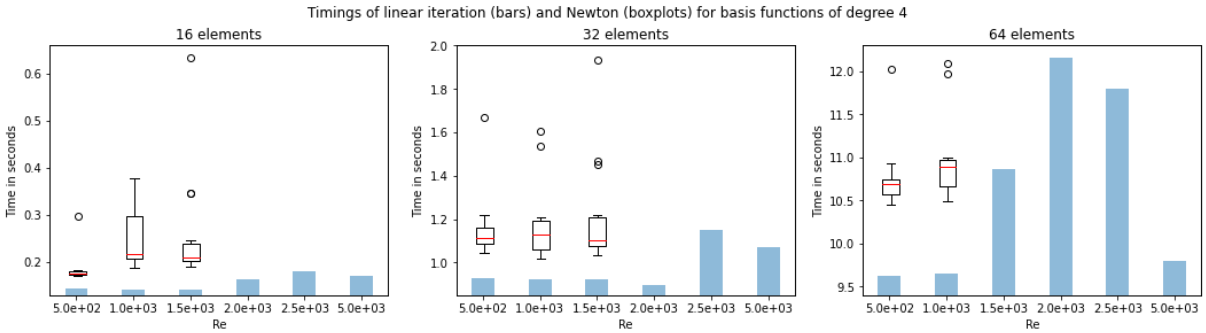

Figure A.24: Timings of linear iteration and timings of Newton-Raphson per Reynolds number for the Navier-Stokes equations with basis functions of degree 4.