A Relation Between Anderson Acceleration and GMRES

Gustaf Lorentzon

Bachelor's thesis

2020:K17

LUND UNIVERSITY

Faculty of Science

Centre for Mathematical Sciences

Numerical Analysis

LUND UNIVERISTY

Abstract

Numerical Analysis
Centre for Mathematical Sciences

A Relation Between Anderson Acceleration and GMRES

by Gustaf Lorentzon

A very common type of problem within mathematics and numerical analysis are fixed-point problems, which can arise as sub-problems of optimization methods, differential equations solvers and much more. The most basic iterative approach for fixed-point problems is fixed-point iteration, special cases of which actually date back as far as the Babylonians, where it was used to to find the square roots of positive numbers. An issue with fixed-point iteration is that it can be very slow, as a consequence, acceleration methods have been developed, which are, not surprisingly, methods for speeding up fixed-point iteration. One of these methods are called Anderson acceleration, which has a very strong relationship with GMRES, an algorithm for solving systems of linear equations, which at first glance, seems completely unrelated. The purpose of this thesis is to investigate the theory behind this relationship and to test it numerically.

Contents

Chapter 1

Introduction

Firstly, this thesis is a review of [3] and unless specified otherwise, all sections up until the numerical implementation are heavily, if not completely, based on [3].

A very common type of problem within mathematics and numerical analysis are fixed-point problems, which can arise as sub-problems of optimization methods, differential equations solvers and much more. We define a fixed-point problem by the following.

Problem 1.1.
Assume we have $g : \mathbb{R}^m \to \mathbb{R}^m$*, then finding* $x \in \mathbb{R}^m$ *such that*

$$x = g(x) \tag{1.1}$$

is a fixed-point problem.

In some cases it is possible to solve this directly using analytical methods, but in other cases this can be extremely difficult or even impossible, therefore it is natural to look for alternative approximation methods. A way of reformulating the fixed-point problem is to define $f(x) \equiv g(x) - x$ and instead formulating it on the equivalent root form; $0 = f(x) = g(x) - x$. This relation allows us to measure how good an approximation is by seeing how close the fixed-point residual; $\|f(x)\|_2$, is to zero, and iterative methods aim to get better

approximations by successively making the fixed-point residual smaller for each iteration. The most basic iterative approach to solving Problem 1.1 is fixed-point iteration, special cases of which actually date date back as far as the Babylonians, where it was used to find the square roots of positive numbers [1].

Algorithm FPI (Fixed-Point Iteration).

Given x_0

for $k = 0, 1, ...$ do;

Set $x_{k+1} = g(x_k)$

An issue with Algorithm FPI is that it does not always converge, and even in the cases where it does converge, it might do so very slowly. A way of dealing with these problems is to use acceleration methods, which can potentially speed up the convergence process and in some cases even decrease the likelihood for divergence. There are many different acceleration methods, but we will put our focus on an algorithm which we refer to as Anderson Acceleration, which we formulate as in [3];

Algorithm AA (Anderson Acceleration).

Given x_0 and $m \geq 1$

Set $x_1 = g(x_0)$

for $k = 1, 2, ...$

Set $m_k = \min\{m, k\}$

Set $F_k = (f_{k-m_k}, ..., f_k)$, where $f_i = g(x_i) - x_i$

Determine $\alpha^{(k)} = (\alpha_0^{(k)}, ..., \alpha_{m_k}^{(k)})^{\mathrm{T}}$ that solves

$$\min_{\alpha^{(k)}=(\alpha_0,...,\alpha_{m_k})^{\mathrm{T}}} \|F_k\alpha\|_2 \text{ such that } \sum_{i=0}^{m_k} \alpha_i = 1. \tag{1.2}$$

Set $x_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i})$

Note that Algorithm AA can be run without truncation by replacing the first line of the for loop by $m_k = k$.

Now we shift our focus to another very common problem within linear algebra known as a system of linear equations.

Problem 1.2.
Assume we have a non-singular matrix $A \in \mathbb{R}^{m\times m}$ and a vector $b \in \mathbb{R}^m$. Then,

$$Ax = b, \tag{1.3}$$

is known as a system of linear equations.

Once again there are many ways of directly solving Problem 1.2, with the most straight-forward approach being to apply Gaussian elimination. But in the case where the m in Problem 1.2 is extremely large, this direct approach might be impossible to execute within a given time frame, in which case one might use an approximation method instead. GMRES is an example of such a method, the details of which are discussed in section 2.1.

What is interesting is that Anderson Acceleration, an algorithm for solving fixed-point problems, and GMRES, an algorithm solving large systems of linear equations, have a very strong relation. To summarize this relation shortly, in the case where g is linear, the algorithms are equivalent in the sense that, under a few additional assumptions, the iterates of either algorithm can be obtained directly from the iterates of the other algorithm. In this thesis we will show this relation theoretically and test it numerically.

Chapter 2

Theory

2.1 GMRES

This entire section is based on the description of GMRES given by [2] and to not litter the text with references it is implied that the information is taken from [2].

2.1.1 Characterization

A non-symmetric system of linear equations is a problem of the type (1.3), where A is non-symmetric, and it is something which can arise from many different problems. In the case where m is extremely large, the system might be impossible to solve within a given time frame by using Gaussian elimination, since it simply takes too many computations.

Instead one might look for iterative methods which approximate the solution and converge to it. If we then perform enough iterations we will get an approximation so close to the exact solution that it is the best approximation given our computer accuracy and its floating point arithmetic. If this is achieved then the approximation will be just as good as finding the exact solution, since the exact solution within floating point arithmetic would then be equal to the approximation.

One way one often goes about this approximation process is to use Krylov subspace methods, one of which is called Generalized Minimized Residual, or GMRES for short, which is described below. Since the main issue of the problem above is that the dimension m of the solution space is too large, finding a way of looking for a solution in a subspace of $\mathbb{R}^m$ with significantly smaller dimension would be very convenient. Krylov subspace methods are a way of doing exactly this.

Definition 2.1.
Assume we have the linear system of equations $Ax = b$ with $A \in \mathbb{R}^{m\times m}$, $b \in \mathbb{R}^m$ and an initial guess for a solution x_0. Then we denote the initial residual by $r_0 = b - Ax_0$ and define the kth Krylov subspace as $\mathcal{K}_k = span\{r_0, Ar_0, ..., A^{k-1}r_0\}$

Given $\mathcal{K}_k$, we can approximate the solution to $Ax = b$ by instead looking for a solution in the restricted solution space $x_0 + \mathcal{K}_k := \{x = x_0 + \hat{x}_k | \hat{x}_k \in \mathcal{K}_k\}$. What characterizes the kth iterate x_k of GMRES is that the norm of the residual $r_k = b - Ax_k$ is minimized, that is, x_k solves

$$\|r_k\|_2 = \min_{x_k \in x_0 + \mathcal{K}_k} \|b - Ax_k\|_2 \tag{2.1}$$

and at every iteration the GMRES algorithm efficiently finds the solution to (2.1). As long as stagnation is not reached, in the sense that $A^{k-1} = I$, the dimension of the solution space $x_0 + \mathcal{K}_k$ will grow, and therefore we hope to get increasingly better solutions. The algorithm terminates when the residual $\|b - Ax_k\|_2$ is lower than a prespecified tolerance, when stagnation is reached, or when a prespecified maximum number of iterations have been reached. Note that in an implementation the number of iterations $k \ll m$, since if the number of dimensions of the solution space $x_0 + \mathcal{K}_k$ of the sub-problem is on the same order as m, the method would just be extremely ineffective.

Another equivalent formulation of the GMRES minimization problem which is more convenient for implementation is finding $\hat{x}_k$ such that

$$\hat{x}_k = \underset{\hat{x}_k \in \mathcal{K}_k}{\arg\min} \|b - A(x_0 + \hat{x}_k)\|_2 = \underset{\hat{x}_k \in \mathcal{K}_k}{\arg\min} \|r_0 - A\hat{x}_k\|_2 , \tag{2.2}$$

or, if we have a that $\{z_1, ..., z_k\}$ is a basis for $\mathcal{K}_k$, then we can use the reformulation $\hat{x}_k = \sum_{i=1}^{k} \alpha_i^{(k)} z_i$ and state the problem as finding $\alpha^{(k)} = (\alpha_1^{(k)}, ..., \alpha_k^{(k)})$ such that

$$\alpha^{(k)} = \underset{\alpha=(\alpha_1,...,\alpha_k)\in\mathbb{R}^k}{\arg\min} \left\| r_0 - A\sum_{i=1}^{k} \alpha_i z_i \right\|_2 . \tag{2.3}$$

The GMRES iterate is then characterized by

$$x_k = x_0 + \hat{x}_k = x_0 + \sum_{i=1}^{k} \alpha_i^{(k)} z_i. \tag{2.4}$$

2.1.2 The GMRES Algorithm

Below we have further details of how GMRES is executed which can be skipped without loss of continuity, the explanation closely follows that of [2].

GMRES is an algorithm which at every iteration efficiently finds the $\hat{x}_k$ which solves the least square problem (2.2). Since $\mathcal{K}_k$ is a k-dimensional subspace of $\mathbb{R}$, the solution space will be of significantly lower dimension for any reasonable number of iterations, but $\hat{x}_k$ is still a $[m \times 1]$ vector. With the intention of representing $\hat{x}_k$ as a $[k \times 1]$ vector, we set up an orthogonal basis for $\mathcal{K}_k$ using the Arnoldi iteration[1]. In the k'th step of the Arnoldi iteration we have the relation

$$AQ_k = Q_{k+1} H_k, \tag{2.5}$$

where the columns of Q_k form an orthonormal basis of $\mathcal{K}_k$, with respect to the Euclidean norm. Thus we can represent any element $\hat{x}_k \in \mathcal{K}_k$ as $\hat{x}_k = Q_k y_k$ with $y_k \in \mathbb{R}^k$ and we have successfully reduced the dimension of the problem since we can rewrite (2.2) into

$$\min_{y_k\in\mathbb{R}^k} \|r_0 - AQ_k y_k\|_2 = \min_{y_k\in\mathbb{R}^k} \|r_0 - Q_{k+1} H_k y_k\|_2 , \tag{2.6}$$

where the second equality follows from (2.5).

Since $r_0 / \|r_0\|_2$ is the first column in Q_{k+1} we have that $r_0 \in \text{img}(Q_{k+1})$ and the second term is in $\text{img}(Q_{k+1})$ by definition, furthermore we have that Q_{k+1} is

[1] [2] provides details of the Arnoldi iteration for the interested reader.

an orthogonal matrix. Thus, multiplying the argument of the norm with Q_{k+1}^{T} from the left will not change the value of the norm and we can rewrite (2.6) into

$$\min_{y_k \in \mathbb{R}^k} \left\| Q_{n+1}^{\mathrm{T}} r_0 - H_k y_k \right\|_2 . \tag{2.7}$$

But since r_0 is orthogonal to all but the first column, by construction, we have that $Q_{n+1}^{\mathrm{T}} r_0 = \|r_0\|_2 e_1$ where e_1 is the first column of the $[k+1] \times [k+1]$ identity matrix. So if we define the function $J : \mathbb{R}^k \to \mathbb{R}$ as

$$J(y) := \left\| \left(\|r_0\|_2 e_1 - H_k y \right) \right\|_2 , \tag{2.8}$$

we can rewrite (2.7) into

$$\min_{y_k \in \mathbb{R}^k} J(y_k) \tag{2.9}$$

and if we recall that $\hat{x}_k = Q_n y_k$, we can conclude that

$$x_k^{\mathrm{GMRES}} = x_0 + Q_k y_k, \tag{2.10}$$

where y_k minimizes $J(y)$. We can formulate the above process as an algorithm [2]:

Algorithm GMRES.

Given x_0

Set $r_0 = b - Ax_0$ and $q_1 = r_0 / \|r_0\|_2$

2. Iterate:

for $j = 1, 2, ..., k, ...,$ until satisfied do;

Set $h_{i,j} = (Aq_j, q_j), i = 1, 2, ..., j,$

$\hat{q}_{j+1} = Aq_j - \sum_{i=1}^{j} h_{i,j} q_i$

$h_{j+1,j} = \|\hat{q}_{j+1}\|_2$, and

$q_{j+1} = \hat{q}_{j+1} / h_{j+1,j}$

Find y_k which minimizes $\min_{y_k \in \mathbb{R}^k} J(y_k)$

Form the approximate solution: $x_k^{\mathrm{GMRES}} = x_0 + Q_k y_k$

---

where $(.\ ,\ .)$ denotes the dot product and q_i denotes the ith column of Q_k.

2.2 Anderson Acceleration and GMRES

In this section we explore some very strong relations between Algorithm AA and GMRES. To begin with we assume throughout this section that Anderson Acceleration is applied to a linear function the form $g(x) := Ax + b$ with $A \in \mathbb{R}^{m\times m}$ and $b \in \mathbb{R}^m$. The logic supporting Algorithm AA can easily be made clear in this case. At iteration k we have that

$$x_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}) = g\Big(\sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}\Big) = g(x_{\min}), \tag{2.11}$$

with $x_{\min} = \sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}$. Furthermore, the fixed-point residual of $x_{\min}$ is

$$g\Big(\sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}\Big) - \sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i} = \sum_{i=0}^{m_k} \alpha_i^{(k)} f_i \tag{2.12}$$

and if we take the norm of this we get precisely what $\alpha^{(k)}$ minimizes in (1.2). Hence, $x_{\min}$ is the weighted average of $\{x_{k-m_k}, ..., x_k\}$ with minimal fixed-point residual, and to obtain the next iterate we plug this into g.

Now that we understand how Algorithm AA works in the linear case, one might ask how this relates to GMRES, which is applied to a system of linear equations, not a fixed-point problem. In Algorithm AA we are looking for a fixed-point x such that

$$Ax + b - x = 0 \tag{2.13}$$

and by rearranging this equation we get

$$b + (A - I)x = 0 \Leftrightarrow (I - A)x = b. \tag{2.14}$$

Thus, finding a fixed-point of a linear function $g(x) = Ax + b$ and solving the linear system of equations $(I - A)x = b$ are equivalent formulations of the same problem.

What we are going to show, is that if GMRES is applied to the system $(I - A)x = b$ and Algorithm AA is applied to $g(x) = Ax+b$, then these are equivalent in the sense that the iterates of each algorithm can be obtained directly from the iterates of the other algorithm. This is true under some assumptions, firstly we

assume that we do not apply truncation in Algorithm AA, that is; $m_k = k$. We also assume both algorithms start at the same initial point, denoted $x_0^{AA} = x_0$ for AA and GMRES respectively. Since GMRES is applied to $(I - A)x = b$, we denote, for each j, the jth iterate of GMRES as x_j^{GMRES} and the residual is defined as $r_j^{\text{GMRES}} := b - (I - A)x_j^{\text{GMRES}}$. The Krylov subspace generated by $(I - A)$ and r_0^{GMRES} will be $\mathcal{K}_j := \text{span}\{r_0^{\text{GMRES}}, (I - A)r_0^{\text{GMRES}}, ..., (I - A)^{j-1}r_0^{\text{GMRES}}\}$. We summarize all the assumptions below, as formulated in [3]:

Assumption 2.2 ([3], Assumption 2.1)**.**

1. $g(x) = Ax + b$ for $A \in \mathbb{R}^{m\times m}$ and $b \in \mathbb{R}^m$
2. Anderson acceleration is not truncated, i.e., $m_k = k$ for each k
3. $(I - A)$ is non-singular
4. GMRES is applied to $(I - A)x = b$ with initial point $x_0 = x_0^{\text{AA}}$

We now move on to the main theorem, as formulated in [3].

Theorem 2.3 ([3], Theorem 2.2)**.**
Suppose that Assumption 2.2 holds, and that, for some $k > 0$, $r_{k-1}^{\text{GMRES}} \neq 0$ *and also* $\left\|r_{j-1}^{\text{GMRES}}\right\|_2 > \left\|r_j^{\text{GMRES}}\right\|_2$ *for each* j *such that* $0 < j < k$. *Then* $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\text{AA}} = x_k^{\text{GMRES}}$ *and* $x_{k+1}^{\text{AA}} = g(x_k^{\text{GMRES}})$.

Before we move on to the proof of the theorem, there are some additional assumptions specified in the theorem that have some implications. Firstly, we see that we do not allow for stagnation in the GMRES algorithm before iteration k, or in other words, the norm of the residual is decreasing with each iterate. At iteration k, we do allow for $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}} \neq 0$, a case which is investigated in Proposition 2.8.

Moreover, the theorem allows for the case $r_k^{\text{GMRES}} = 0$, which is discussed and investigated in remarks 2.5 and 2.7 and Proposition 2.6.

Proof. Firstly, for $i = 1, ..., k$ we define $z_i := x_i^{\text{AA}} - x_0$. Now we prove the theorem in two steps, by proving the two claims below, from which the theorem directly follows.

Claim 1. For $1 \leq j \leq k$, if $\{z_1, ..., z_j\}$ is a basis of $\mathcal{K}_j$, then $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\mathrm{AA}} = x_k^{\mathrm{GMRES}}$ and $x_{k+1}^{\mathrm{AA}} = g(x_k^{\mathrm{GMRES}})$.

Claim 2. For $1 \leq j \leq k$, $\{z_1, ..., z_j\}$ is a basis of $\mathcal{K}_j$.

We begin with proving Claim 1. Firstly, as defined in Algorithm AA, we have

$$f_0 = g(x_0) - x_0 = Ax_0 + b - x_0 = b + (A - I)x_0 = b - (I - A)x_0 = r_0^{\mathrm{GMRES}}, \quad (2.15)$$

where everything follows from definitions. For $i = 1, ..., k$ we have that

$$\begin{aligned} f_i &= g(x_i^{\mathrm{AA}}) - x_i^{\mathrm{AA}} \\ &= Ax_i^{\mathrm{AA}} + b - x_i^{\mathrm{AA}} \\ &= A(z_i + x_0) + b - (z_i + x_0) \\ &= b + Ax_0 - x_0 + Az_i - z_i \\ &= b - (I - A)x_0 - (I - A)z_i \\ &= r_0^{\mathrm{GMRES}} - (I - A)z_i, \end{aligned} \quad (2.16)$$

where in the third equality we substitute $x_i^{AA} = z_i + x_0$ from the definition of z_i and the rest is simply rearranging. From (2.15) and (2.16) we get that, for the F_j of Algorithm AA and any $\alpha = (\alpha_0, ..., \alpha_k)^{\mathrm{T}}$, we have

$$\begin{aligned} F_j\alpha = \sum_{i=0}^{j} \alpha_i f_i &= \alpha_0 f_0 + \sum_{i=1}^{j} \alpha_i f_i \\ &= \alpha_0 r_0^{\mathrm{GMRES}} + \sum_{i=1}^{j} \alpha_i (r_0^{\mathrm{GMRES}} - (I - A)z_i) \\ &= \sum_{i=0}^{j} \alpha_i r_0^{\mathrm{GMRES}} - \sum_{i=1}^{j} \alpha_i (I - A)z_i \\ &= r_0^{\mathrm{GMRES}} \big(\sum_{i=0}^{j} \alpha_i \big) - (I - A) \sum_{i=1}^{j} \alpha_i z_i, \end{aligned} \quad (2.17)$$

where we use (2.15) and (2.16) in the third equality.

From this we will be able to show that $\alpha^{(j)} = (\alpha_0^{(j)}, ..., \alpha_j^{(j)})^{\mathrm{T}}$ with $\alpha_0^{(j)} = 1 - \sum_1^j \alpha_i^{(j)}$ solves

$$\min_{\alpha = (\alpha_0, ..., \alpha_j)^{\mathrm{T}}} \|F_j \alpha\|_2 \text{ such that } \sum_{i=0}^{j} \alpha_i = 1 \quad (2.18)$$

if and only if $(\alpha_1^{(j)}, ..., \alpha_j^{(j)})$ solves

$$\min_{(\alpha_1,...,\alpha_j)^{\mathrm{T}}} \left\| r_0^{\mathrm{GMRES}} - (I - A) \sum_{i=1}^{j} \alpha_i z_i \right\|_2 . \tag{2.19}$$

The logic behind the equivalence is as follows, from (2.17), we see that (2.18) is equivalent to the problem

$$\min_{\alpha=(\alpha_0,...,\alpha_j)^{\mathrm{T}}} \left\| r_0^{\mathrm{GMRES}} \big(\sum_{i=0}^{j} \alpha_i \big) - (I - A) \sum_{i=1}^{j} \alpha_i z_i \right\|_2 \text{ such that } \sum_{i=0}^{j} \alpha_i = 1, \tag{2.20}$$

which is the same as

$$\min_{\alpha=(\alpha_0,...,\alpha_j)^{\mathrm{T}}} \left\| r_0^{\mathrm{GMRES}} - (I - A) \sum_{i=1}^{j} \alpha_i z_i \right\|_2 \text{ such that } \sum_{i=0}^{j} \alpha_i = 1. \tag{2.21}$$

In the above argument, we can see that α_0 does not affect the norm, so we can use this degree of freedom to remove the constraint by setting $\alpha_0 = 1 - \sum_{i=1}^{j} \alpha_i$. From this we get the equivalence of the solution of (2.18) and (2.19).

We now use the assumption of Claim 1 that $\{z_1, .., z_j\}$ is a basis for $\mathcal{K}_j$. From this we see that (2.19) is in fact a GMRES minimization problem on the form (2.3), where $(\alpha_1, ..., \alpha_j)$ are the coefficients of the basis elements of the Krylov subspace. A consequence of this is that the solution $\alpha^{(j)} = (\alpha_0^{(j)}, ..., \alpha_j^{(j)})^{\mathrm{T}}$ to (2.18), also satisfies the the following relation

$$\begin{aligned} \sum_{i=0}^{j} \alpha_i^{(j)} x_i^{AA} &= \alpha_0^{(j)} x_0^{AA} + \sum_{i=1}^{j} \alpha_i^{(j)} x_i^{AA} \\ &= \alpha_0^{(j)} x_0^{AA} + \sum_{i=1}^{j} \alpha_i^{(j)} (x_0 + z_i) \\ &= x_0 \Big(\sum_{i=0}^{j} \alpha_i^{(j)} \Big) + \sum_{i=1}^{j} \alpha_i^{(j)} z_i \\ &= x_0 + \hat{x}_j^{\mathrm{GMRES}} = x_j^{\mathrm{GMRES}}, \end{aligned} \tag{2.22}$$

where the last equality follows from $(\alpha_1^{(k)}, ..., \alpha_j^{(k)})^{\mathrm{T}}$ being the solution to (2.19). This proves the first part of Claim 1.

Now we apply g on both sides of (2.22) to show the second statement of

Claim 1

$$g(x_j^{\text{GMRES}}) = g\Big(\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{AA}\Big) = \sum_{i=0}^{j} \alpha_i^{(j)} g(x_i^{AA}) = x_{j+1}^{\text{AA}}, \tag{2.23}$$

where the second equality follows from the linearity of g and the third follows from the definition of the next iterate in Algorithm AA. From this, Claim 1 has been proven and we move on to proving Claim 2.

We prove Claim 2 by induction over j and thus begin with the basic case of z_1 being a basis for $\mathcal{K}_1 = \text{span}\{r_0^{\text{GMRES}}\}$. It follows from the definition of z_1 that

$$z_1 = x_1^{\text{AA}} - x_0 = g(x_0) - x_0 = r_0^{\text{GMRES}} \tag{2.24}$$

and what is left to show is that r_0^{GMRES} is non-zero. From the assumption of the theorem we have that $r_{k-1}^{\text{GMRES}} \neq 0$ and $\left\|r_{j-1}^{\text{GMRES}}\right\|_2 > \left\|r_j^{\text{GMRES}}\right\|_2$ for $j = 1, .., k-1$, thus, we have

$$\left\|r_0^{\text{GMRES}}\right\|_2 > \left\|r_1^{\text{GMRES}}\right\|_2 > ... > \left\|r_{k-1}^{\text{GMRES}}\right\|_2 > 0. \tag{2.25}$$

Therefore $z_1 = \{r_0^{\text{GMRES}}\}$ is a basis for $\mathcal{K}_1$ and for the basic case $k = 1$ the theorem is proven.

Now we assume $k > 1$ and use that, for some $0 < j < k$, $\{z_1, ..., z_j\}$ is a basis for $\mathcal{K}_j$ as an induction hypothesis to prove that $\{z_1, ..., z_{j+1}\}$ is a basis for $\mathcal{K}_{j+1}$. We do this by first proving that $z_{j+1} \in \mathcal{K}_{j+1}$ and then that $z_{j+1} \notin \mathcal{K}_j$. Firstly, we have that

$$\begin{aligned} z_{j+1} &= x_{j+1}^{\text{AA}} - x_0 g(x_j^{\text{GMRES}}) - x_0 = \\ &= Ax_j^{\text{GMRES}} + b - x_0 + x_j^{\text{GMRES}} - x_j^{\text{GMRES}} \\ &= b - (I - A)x_j^{\text{GMRES}} - x_0 + x_j^{\text{GMRES}} \\ &= r_j^{\text{GMRES}} - x_0 + x_j^{\text{GMRES}} \\ &= r_j^{\text{GMRES}} + \hat{x}_j^{\text{GMRES}}, \end{aligned} \tag{2.26}$$

where the second equality follows from Claim 1. Moreover, by the induction hypothesis, $\hat{x}_j^{\text{GMRES}}$ can be represented by the basis $\{z_1, ..., z_j\}$, so we get

$$z_{j+1} = r_j^{\text{GMRES}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_j, \tag{2.27}$$

for the $(\alpha_1^{(j)}, ..., \alpha_j^{(j)})^{\mathrm{T}}$ which minimize (2.19), since $\hat{x}_j^{\mathrm{GMRES}} = \sum_{i=1}^{j} \alpha_i^{(j)} z_j$ for this $\alpha_j^{(j)}$ by the characterization in (2.3). From the definition we have that

$$\begin{aligned} r_j^{\mathrm{GMRES}} &= b - (I - A)x_j^{\mathrm{GMRES}} = b - (I - A)(x_0 + \hat{x}_j^{\mathrm{GMRES}}) \\ &= b - (I - A)x_0 - (I - A)\hat{x}_j^{\mathrm{GMRES}} \\ &= r_0^{\mathrm{GMRES}} - (I - A)\hat{x}_j^{\mathrm{GMRES}}. \end{aligned} \tag{2.28}$$

Moreover, we have that $r_0^{\mathrm{GMRES}} \in \mathcal{K}_{j+1}$ and $(I - A)\hat{x}_j^{\mathrm{GMRES}} \in \mathcal{K}_{j+1}$, which implies $r_j^{\mathrm{GMRES}} \in \mathcal{K}_{j+1}$. By the induction hypothesis the second term of (2.27) is also in $\mathcal{K}_{j+1}$, thus, we conclude that $z_{j+1} \in \mathcal{K}_{j+1}$. Furthermore, since $\left\|r_{j-1}^{\mathrm{GMRES}}\right\|_2 > \left\|r_j^{\mathrm{GMRES}}\right\|_2 \geq \left\|r_{k-1}^{\mathrm{GMRES}}\right\|_2 > 0$, it follows from Lemma 2.4 that $r_j^{\mathrm{GMRES}} \notin \mathcal{K}_j$. From this and (2.27) we can conclude that z_{j+1} cannot depend linearly on $\{z_1, ..., z_j\}$, which implies that $\{z_1, ..., z_{j+1}\}$ is a basis for $\mathcal{K}_{j+1}$. Thus, the induction proof of Claim 2 is done and the theorem is proven. □

To complete the proof of Theorem 2.3, we also need to prove the following supportive lemma.

Lemma 2.4 ([3], Lemma 2.4)**.**
Suppose that GMRES is applied to $Mx = b$ with M non-singular. If $\left\|r_{j-1}^{GMRES}\right\|_2 > \left\|r_j^{GMRES}\right\|_2 > 0$ for some $j > 0$, then $r_j^{\mathrm{GMRES}} \notin \mathcal{K}_j$.

Proof. For the sake of simplifying the proof, we define $K_0 \equiv \{0\}$. For $\ell \geq 0$, denote $r_\ell := r_\ell^{\mathrm{GMRES}}$ and $x_\ell := x_\ell^{\mathrm{GMRES}}$ and the orthogonal projection onto $(M\mathcal{K}_\ell)^\perp$ by π_ℓ.

To begin with, we prove that $\pi_j r_{j-1} = r_j$ by induction over $\ell = 1, ..., j$. We have that

$$\begin{aligned} \|r_\ell\|_2 &= \min_{\hat{x}_\ell \in \mathcal{K}_\ell} \|b - M(x_0 + \hat{x}_\ell)\|_2 = \min_{\hat{x}_\ell \in \mathcal{K}_\ell} \|r_0 - M\hat{x}_\ell\|_2 \\ &= \min_{y_\ell \in M\mathcal{K}_\ell} \|r_0 - y_\ell\|_2 , \end{aligned} \tag{2.29}$$

from which it follows by the minimizing property that; $y_\ell = Mx_\ell$ is the orthogonal projection of r_0 on $M\mathcal{K}_\ell$ and r_ℓ is the orthogonal projection of r_0 on the orthogonal complement $(M\mathcal{K}_\ell)^\perp$, or in other words

$$\pi_\ell r_0 = r_\ell. \tag{2.30}$$

The base case $\ell = 1$ follows directly from this, so we make the induction hypothesis that; for some k such that $1 < k \leq j$, we have that $\pi_\ell r_{\ell-1} = r_\ell$ for all $1 \leq \ell < k$ and we need to show that this implies $\pi_k r_{k-1} = r_k$. We begin by observing that the induction hypothesis implies that

$$\pi_k r_{k-1} = \pi_k \pi_{k-1} r_{k-2} = ... = \pi_k ... \pi_0 r_0. \tag{2.31}$$

Since M is nonsingular, we have $M\mathcal{K}_{\ell-1} \subset M\mathcal{K}_\ell$, and consequently $(M\mathcal{K}_\ell)^\perp \subset (M\mathcal{K}_{\ell-1})^\perp$, for $\ell = 1, ..., j$. Therefore, we are dealing with orthogonal projections to successively smaller nested subspaces and it follows that

$$\pi_k ... \pi_0 r_0 = \pi_k r_0. \tag{2.32}$$

As a consequence of (2.30), (2.31) and (2.32), we have that $\pi_k r_{k-1} = \pi_k r_0 = r_k$, which completes the induction and proves that $\pi_j r_{j-1} = r_j$.

With this property proven we go on to finish the proof by contradiction. From the assumptions of the Lemma we have that $r_j \neq 0$ for some $j > 0$. Then if we assume $r_j \in \mathcal{K}_j$, we get that $r_j \in \mathcal{K}_j \cap (M\mathcal{K}_j)^\perp \subset \mathcal{K}_j \cap (M\mathcal{K}_{j-1})^\perp$. Moreover, we have that $\mathcal{K}_j = \text{span}\{r_0, Mr_0, ..., M^{j-1}r_0\}$ and $M\mathcal{K}_{j-1} = \text{span}\{Mr_0, ..., M^{j-1}r_0\}$. If we let $\{q_1, ..., q_j\}$ and $\{q_2, ..., q_j\}$ be orthogonal bases for $\mathcal{K}_j$ and $M\mathcal{K}_{j-1}$ respectively, so that q_1 is the additional dimension of $\mathcal{K}_j$, then we see that q_1 is the only dimension shared by $\mathcal{K}_j$ and $(M\mathcal{K}_{j-1})^\perp$. Consequently, $\mathcal{K}_j \cap (M\mathcal{K}_{j-1})^\perp$ is a one-dimensional subspace, which contains both r_{j-1} and r_j, which implies that $r_j = \lambda r_{j-1}$ for some $\lambda \in \mathbb{R}$. But then $r_j = \pi_j r_j = \lambda \pi_j r_{j-1} = \lambda r_j$ by the relation proved above. Since $r_j \neq 0$ we have that $\lambda = 1$, which implies $r_j = r_{j-1}$ and $\|r_j\|_2 = \|r_{j-1}\|_2$, which contradicts an assumption of the Lemma and we can conclude that $r_j \notin \mathcal{K}_j$. □

Remark 2.5.

If in addition to the assumptions of Theorem 2.3 we have that $r_k^{\text{GMRES}} = 0$, that is, the GMRES algorithm finds an exact solution after k iterations, then

$$\begin{aligned} & r_k^{\text{GMRES}} = b - (I - A)x_k^{\text{GMRES}} = 0 \\ \Leftrightarrow\ & b + Ax_k^{\text{GMRES}} = x_k^{\text{GMRES}} \\ \Leftrightarrow\ & g(x_k^{\text{GMRES}}) = x_k^{\text{GMRES}}. \end{aligned} \tag{2.33}$$

Moreover, by Theorem 2.3, we have that in this case $x_{k+1}^{\mathrm{AA}} = g(x_k^{\mathrm{GMRES}}) = x_k^{\mathrm{GMRES}}$, which is a fixed-point, and thus the AA algorithm would find an exact solution and terminate after $k+1$ steps.

We now derive the following proposition which shows us some further implications in the case where $r_k^{\mathrm{GMRES}} = 0$.

Proposition 2.6 ([3], Proposition 2.6)**.**
Suppose that the assumptions of Theorem 2.3 hold. Then rank $F_k \geq k$, and rank $F_k = k$ if and only if $r_k^{GMRES} = 0$.

Proof. We use the relation (2.17) developed in the proof of Theorem 2.3, with $j = k$, and we see that $F_k\alpha = 0$ with $\alpha = (\alpha_0, ..., \alpha_k)^{\mathrm{T}}$ if and only if

$$(I - A)\sum_{i=1}^{k} \alpha_i z_i = r_0^{\mathrm{GMRES}}\big(\sum_{i=0}^{k} \alpha_i\big), \tag{2.34}$$

from which it follows that

$$(I - A)\sum_{i=1}^{k} \alpha_i z_i = 0 \Leftrightarrow r_0^{\mathrm{GMRES}}\big(\sum_{i=0}^{k} \alpha_i\big) = 0 \Leftrightarrow \sum_{i=0}^{k} \alpha_i = 0, \tag{2.35}$$

where the last equivalence follows from $r_0^{\mathrm{GMRES}} \neq 0$. Furthermore, since $(I - A)$ is of full rank and $\{z_1, ..., z_k\}$ are linearly independent, we have that

$$(I - A)\sum_{i=1}^{k} \alpha_i z_i = 0 \Leftrightarrow \alpha = 0. \tag{2.36}$$

Therefore, when considering non-trivial solutions to $F_k\alpha = 0$, that is, solutions where $\alpha \neq 0$, it follows from (2.35) and (2.36) that $\sum_{i=0}^{k} \alpha_i = \lambda$ for some $\lambda \in \mathbb{R}\backslash\{0\}$. Now suppose we have two non-trivial solutions $\alpha = (\alpha_0, ..., \alpha_k)^{\mathrm{T}}$ and $\bar{\alpha} = (\bar{\alpha}_0, ..., \bar{\alpha}_k)^{\mathrm{T}}$ to $F_k\alpha = 0$ such that

$$\sum_{i=0}^{k} \alpha_i = \lambda \text{ and } \sum_{i=0}^{k} \bar{\alpha}_i = \bar{\lambda}. \tag{2.37}$$

Then from (2.34) and (2.37) we have

$$\begin{cases} (I - A)\sum_{i=1}^{k} \alpha_i z_i = \lambda r_0^{\mathrm{GMRES}} \\ (I - A)\sum_{i=1}^{k} \bar{\alpha}_i z_i = \bar{\lambda} r_0^{\mathrm{GMRES}} \end{cases} \tag{2.38}$$

which is equivalent to

$$\begin{aligned}
&\frac{(I-A)}{\lambda}\sum_{i=1}^{k}\alpha_i z_i = r_0^{\text{GMRES}} = \frac{(I-A)}{\bar{\lambda}}\sum_{i=1}^{k}\bar{\alpha}_i z_i \\
\Leftrightarrow &\frac{(I-A)}{\lambda}\sum_{i=1}^{k}\alpha_i z_i - \frac{(I-A)}{\bar{\lambda}}\sum_{i=1}^{k}\bar{\alpha}_i z_i = 0 \\
\Leftrightarrow &\frac{(I-A)}{\lambda\bar{\lambda}}\Big(\bar{\lambda}\sum_{i=1}^{k}\alpha_i z_i - \lambda\sum_{i=1}^{k}\bar{\alpha}_i z_i\Big) = 0 \\
\Leftrightarrow &\frac{(I-A)}{\lambda\bar{\lambda}}\sum_{i=1}^{k} z_i(\bar{\lambda}\alpha_i - \lambda\bar{\alpha}_i) = 0.
\end{aligned} \tag{2.39}$$

Consequently, from the same argument as before that $(I-A)$ is full rank and $\{z_1, ..., z_k\}$ is linearly independent, we have

$$\bar{\lambda}\alpha - \lambda\bar{\alpha} = 0 \Leftrightarrow \alpha = (\lambda/\bar{\lambda})\alpha_i, \tag{2.40}$$

which means any two nontrivial solutions depend linearly on each other and therefore $\text{Dim}(\text{Ker}(F_k))$ is at most 1. From the definition of F_k we have that it has $k+1$ columns, which implies rank $F_k \geq k$.

What is left to prove is that rank $F_k = k$ if and only if $r_k^{\text{GMRES}} = 0$. We have that rank $F_k = k$ if and only if there exists an $\alpha = (\alpha_0, ..., \alpha_k)^{\text{T}}$ such that $\sum_{i=0}^{k}\alpha_i = \lambda > 0$ and $F_k\alpha = 0$. If we define $\alpha^{(k)} = (\alpha_0^{(k)}, ..., \alpha_k^{(k)})^{\text{T}} = (1/\lambda)\alpha$ then $\sum_{i=0}^{k}\alpha_i^{(k)} = 1$ and if we plug this into (2.34) we get

$$\begin{aligned}
&(I-A)\sum_{i=1}^{k}\alpha_i^{(k)} z_i = r_0^{\text{GMRES}} \\
\Leftrightarrow\ &r_0^{\text{GMRES}} - (I-A)\sum_{i=1}^{k}\alpha_i^{(k)} z_i = 0,
\end{aligned} \tag{2.41}$$

which is the GMRES minimization problem in (2.19). Thus, it follows by definition that $r_k^{\text{GMRES}} = 0$ if and only if (2.41) holds and consequently, $r_k^{\text{GMRES}} = 0 \Leftrightarrow$ rank $F_k = k$. □

Remark 2.7.

According to Proposition 2.6, when $r_k^{\text{GMRES}} = 0$ we get that $F_k \in \mathbb{R}^{m\times(k+1)}$ is not of full rank. But since (2.19) is a least square problem with a unique

solution, which is equivalent to (2.18) under the assumptions of Theorem 2.3, we have that (2.18) has a unique solution, which implies that x_{k+1}^{AA} can still be determined uniquely.

As previously mentioned, the assumptions of Theorem 2.3 allow for stagnation of GMRES in the kth step, that is; $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}} \neq 0$ and in the following proposition we investigate what implications this has on Algorithm AA.

Proposition 2.8 ([3], Proposition 2.8)**.**
Suppose that the assumptions of Theorem 2.3 hold and that $r_{k-1}^{GMRES} = r_k^{GMRES} \neq 0$*, then* $x_{k+1}^{AA} = x_k^{AA}$.

Proof. For $j = k-1$ we know that (2.19) has a unique solution $\alpha^{(k-1)} = (\alpha_1^{(k-1)}, ..., \alpha_{k-1}^{(k-1)})^{\text{T}}$. Furthermore, since $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}}$ we have that the unique solution of (2.19) for $j = k$ is $\alpha^{(k)} = (\alpha_1^{(k-1)}, ..., \alpha_{k-1}^{(k-1)}, 0)^{\text{T}}$ since for this $\alpha^{(k)}$ we have

$$\begin{aligned} r_k^{\text{GMRES}} &= r_0^{\text{GMRES}} - (I - A)\sum_{i=1}^{k} z_i \alpha_i^{(k)} \\ &= r_0^{\text{GMRES}} - (I - A)\sum_{i=1}^{k-1} z_i \alpha_i^{(k-1)} = r_{k-1}^{\text{GMRES}}. \end{aligned} \tag{2.42}$$

With this $\alpha^{(k)}$ and Theorem 2.3 we get

$$\begin{aligned} x_{k+1}^{\text{AA}} &= g(x_k^{\text{GMRES}}) = g\Big(\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\text{AA}}\Big) \\ &= \sum_{i=0}^{k-1} \alpha_i^{(k-1)} g(x_i^{\text{AA}}) = x_k^{\text{AA}}. \end{aligned} \tag{2.43}$$

□

From Proposition 2.8 we see that stagnation of GMRES in the kth step implies that $x_{k+1}^{\text{AA}} = x_k^{\text{AA}}$, meaning stagnation of Algorithm AA in the $(k+1)$th step. This in itself does not seem too bad since GMRES does not break down upon stagnation and can continue making progress, but we will see that this is not the case for Algorithm AA.

It follows by definition that $f_{k+1} = g(x_{k+1}^{\mathrm{AA}}) - x_{k+1}^{\mathrm{AA}} = g(x_k^{\mathrm{AA}}) - x_k^{\mathrm{AA}} = f_k$, meaning the last two columns of F_{k+1} are the same. This means that the least square problem (1.2) is rank deficient for F_{k+1} and lacks a unique solution. An approach to deal with this problem could be to add further specification of how to pick $\alpha^{(k+1)}$, so that x_{k+2}^{AA} can be uniquely determined, but we will see that this would be pointless.

In iteration k, the unique solution to (1.2) is $\alpha^{(k)} = (\alpha_0^{(k)}, ..., \alpha_k^{(k)})^{\mathrm{T}}$. In iteration $k+1$, the argument of the norm in (1.2) can be rewritten into

$$\begin{aligned} F_{k+1}\alpha^{(k+1)} &= \alpha_0^{(k+1)} f_0 + ... + \alpha_k^{(k+1)} f_k + \alpha_{k+1}^{(k+1)} f_{k+1} \\ &= \alpha_0^{(k+1)} f_0 + ... + (\alpha_k^{(k+1)} + \alpha_{k+1}^{(k+1)}) f_k. \end{aligned} \tag{2.44}$$

Therefore, any solution to (1.2) in iteration $k+1$ satisfies $\alpha^{(k+1)} = (\alpha_0^{(k)}, ..., \alpha_{k-1}^{(k)}, (1-\lambda)\alpha_k^{(k)}, \lambda\alpha_k^{(k)})^{\mathrm{T}}$ for some $\lambda \in \mathbb{R}$. From this and the fact that $x_{k+1}^{\mathrm{AA}} = x_k^{\mathrm{AA}}$ we get the next iterate

$$\begin{aligned} x_{k+2}^{\mathrm{AA}} &= \sum_{i=0}^{k+1} \alpha_i^{(k+1)} g(x_i^{AA}) \\ &= \sum_{i=0}^{k-1} \alpha_i^{(k)} g(x_i^{AA}) + (1-\lambda)\alpha_k^k g(x_k^{\mathrm{AA}}) + \lambda\alpha_k^k g(x_{k+1}^{\mathrm{AA}}) \\ &= \sum_{i=0}^{k-1} \alpha_i^{(k)} g(x_i^{AA}) + ((1-\lambda)+\lambda) g(x_k^{\mathrm{AA}})\alpha_k^k \\ &= \sum_{i=0}^{k} \alpha_i^{(k)} g(x_i^{AA}) = x_{k+1}^{\mathrm{AA}} \qquad \forall \lambda \in \mathbb{R}, \end{aligned} \tag{2.45}$$

or in other words, no matter how we specify the solution of (1.2) for iterate $k+1$, we have that $x_k^{\mathrm{AA}} = x_{k+1}^{\mathrm{AA}} = x_{k+2}^{\mathrm{AA}}$. From this we see that $f_{k+2} = f_{k+1} = f_k$ and repeating this process shows that $x_k^{\mathrm{AA}} = x_{k+1}^{\mathrm{AA}} = x_{k+2}^{\mathrm{AA}} = ...$, meaning that if stagnation occurs, Algorithm AA can make no further progress.

Moreover, if two successive iterates are nearly the same so that $x_{k+1}^{\mathrm{AA}} \approx x_k^{\mathrm{AA}}$, then we expect ill-conditioning of the least squares problem (1.2) in floating point arithmetic. Therefore, a practical implementation of the algorithm should terminate in either of these scenarios. This is a numerical weakness of Algorithm AA not prevalent in GMRES, which does not break down upon this type of

stagnation.

Additionally, if GMRES would stagnate in the sense that $(I-A)^{k-1} = I$ then it would follow that $\mathcal{K}_k = \text{span}\{r_0^{\text{GMRES}}, ..., r_0^{\text{GMRES}}(I-A)^{k-2}, r_0^{\text{GMRES}}\} = \mathcal{K}_{k-1}$, implying $r_k = r_{k-1}$ and hence Algorithm AA would also stagnate.

2.2.1 A system with a preconditioner

In this subsection we consider problem (1.3) with a preconditioner. A preconditioner is commonly used to speed up the process of solving a system of linear equations, and is typically an easily obtained approximation of the inverse of A, multiplied from the left to improve the system while using an iterative process such as GMRES.

Definition 2.9.

Assume we have a system of linear equation as in Problem 1.3 and that $M^{-1} \in \mathbb{R}^{m\times m}$ is a non-singular matrix, then

$$M^{-1}Ax = M^{-1}b \tag{2.46}$$

is a left preconditioned system with the preconditioner M.

If we also use the the matrix splitting $A = M - N$ with $M, N \in \mathbb{R}^{m\times m}$ and M is non-singular, then we can reformulate the problem

$$\begin{aligned} &M^{-1}Ax = M^{-1}b \Leftrightarrow M^{-1}(M-N)x = M^{-1}b \\ &\Leftrightarrow (I - M^{-1}N)x = M^{-1}b \Leftrightarrow M^{-1}Nx + M^{-1}b = x. \end{aligned} \tag{2.47}$$

It is now a fixed-point problem with $g(x) \equiv (M^{-1}N)x + M^{-1}b$. In the following corollary of Theorem 2.3, we use this to show that Anderson Acceleration applied to this g without truncation is, in the same sense as previously, equivalent to GMRES applied to the left preconditioned system $M^{-1}Ax = M^{-1}b$. We summarize our assumptions below.

Assumption 2.10 ([3], Proposition 2.9)**.**

1. $A = M - N$, where both $A \in \mathbb{R}^{m\times m}$ and $M \in \mathbb{R}^{m\times m}$ are non-singular.
2. $g(x) = M^{-1}Nx + M^{-1}b$ for $b \in \mathbb{R}^n$
3. Anderson acceleration is not truncated, i.e., $m_k = k$ for each k
4. GMRES is applied to the left preconditioned system $M^{-1}Ax = M^{-1}b$ with initial point $x_0 = x_0^{\mathrm{AA}}$.

Our notation is the same as before except $r_j^{\mathrm{GMRES}} \equiv M^{-1}b - M^{-1}Nx_j^{\mathrm{GMRES}}$.

Corollary 2.11 ([3], Corollary 2.10)**.**
Suppose that Assumption 2.10 holds and that, for some $k > 0$, $r_{k-1}^{\mathrm{GMRES}} \neq 0$ *and also* $\left\|r_{j-1}^{\mathrm{GMRES}}\right\|_2 > \left\|r_j^{\mathrm{GMRES}}\right\|_2$ *for each* j *such that* $0 < j < k$*. Then* $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\mathrm{AA}} = x_k^{\mathrm{GMRES}}$ *and* $x_{k+1}^{\mathrm{AA}} = g(x_k^{\mathrm{GMRES}})$.

Proof. We know that $A = M - N$ is non-singular, therefore multiplying by the non-singular matrix M cannot reduce the dimension of the image space and we get that $M^{-1}A = M^{-1}(M - N) = (I - M^{-1}N)$ is non-singular. If we now replace A and b of Theorem 2.3 with $\hat{A} := M^{-1}N$ and $\hat{b} := M^{-1}b$ respectively, then we see that all the assumptions are satisfied. □

Due to the numerical weakness discussed at the end of Section 2.2, Algorithm AA is not recommended as a substitute to preconditioned GMRES, but the option does exist.

2.2.2 AA with the Least Square Problem on Unconstrained Form

Before moving on to testing the relation between Anderson Acceleration and GMRES numerically, we note that there are several equivalent forms of the least squares problem (1.2), which can have different advantages. In the implementation later, an unconstrained form is used, so that the QR algorithm can

be used to solve the system, since this was recommended in [3]. We will use the equivalent form of (1.2);

$$\min_{\gamma=(\gamma,\dots,\gamma_{m_k-1})^{\mathrm{T}}} \|f_k - \mathcal{F}_k\gamma\|_2\,, \tag{2.48}$$

where $\mathcal{F}_k = (\Delta f_{k-m_k}, ..., \Delta f_{k-1})$ with $\Delta f_i = f_{i+1} - f_i$ for each i, and where α and γ are related by

$$\begin{cases} \alpha_0 = \gamma_0 \\ \alpha_i = \gamma_i - \gamma_{i-1} \text{ for } 1 \leq i \leq m_k - 1 \\ \alpha_{m_k} = 1 - \gamma_{m_k-1} \end{cases}. \tag{2.49}$$

It follows from the lemma below that they are indeed equivalent.

Lemma 2.12.
In iteration k of Algorithm AA, $\alpha^{(k)}$ solves (1.2) if and only if $\gamma^{(k)}$ solves (2.48), where $\alpha^{(k)}$ and $\gamma^{(k)}$ are related by (2.49).

Proof. Firstly, we have that the argument of the norm in (2.48) can be rewritten as follows

$$\begin{aligned} &f_k - \mathcal{F}_k\gamma = f_k - \sum_{i=0}^{m_k-1} \gamma_i(\Delta f_{k-m_k+i}) = \\ &f_k - \left(\sum_{i=0}^{m_k-1} \gamma_i f_{k-m_k+i+1} - \sum_{i=0}^{m_k-1} \gamma_i f_{k-m_k+i}\right) = \\ &f_k - \left(\sum_{i=1}^{m_k} \gamma_{i-1} f_{k-m_k+i} - \sum_{i=0}^{m_k-1} \gamma_i f_{k-m_k+i}\right) = \\ &f_k - f_k\gamma_{m_k-1} + \gamma_0 f_{k-m_k} + \left(\sum_{i=1}^{m_k-1} (\gamma_i - \gamma_{i-1})(f_{k-m_k+i})\right) = \\ &\quad (1-\gamma_{m_k-1})f_k + \gamma_0 f_{k-m_k} + \left(\sum_{i=1}^{m_k-1} (\gamma_i - \gamma_{i-1})(f_{k-m_k+i})\right) \\ &\quad = \sum_{i=0}^{m_k} \alpha_i(f_{k-m_k+i}) = F_k\alpha, \end{aligned} \tag{2.50}$$

where we used relation (2.49) in the second to last equality. From this it follows that $\gamma^{(k)}$ solves (2.48) if and only if $\alpha^{(k)}$ solves

$$\min_{\alpha=(\alpha_0,\dots,\alpha_k)^{\mathrm{T}}} \|F_k\alpha\|_2 \text{ such that (2.49) holds.} \tag{2.51}$$

Furthermore, we have from (2.49) that for $2 \leq j \leq m_k - 1$

$$\gamma_j = \alpha_j + \gamma_{j-1} = \alpha_j + (\alpha_{j-1} + \gamma_{j-2}) = ... = \sum_{i=0}^{j} \alpha_j \tag{2.52}$$

and using the above relation we see that (2.49) implies $\alpha_{m_k} = 1 - \gamma_{m_k-1} = 1 - \sum_{i=0}^{m_k-1} \alpha_i$, from which it follows that $\sum_{i=0}^{m_k} \alpha_i = 1$. Therefore, $\gamma^{(k)}$ solves (2.48) if and only if $\alpha^{(k)}$ solves

$$\min_{\alpha=(\alpha_0,\ldots,\alpha_k)^{\mathrm{T}}} \|F_k\alpha\|_2 \text{ such that } \sum_{i=0}^{m_k} \alpha_i = 1, \tag{2.53}$$

which is the same as (1.2). □

Lastly, the next iterate of Algorithm AA can be obtained directly from $\gamma^{(k)}$ by

$$x_{k+1} = g(x_k) - \sum_{i=0}^{m_k-1} \gamma_i^{(k)} [g(x_{k-m_k+i+1}) - g(x_{k-m_k+i})], \tag{2.54}$$

since

$$\begin{aligned}
& g(x_k) - \sum_{i=0}^{m_k-1} \gamma_i^{(k)} [g(x_{k-m_k+i+1}) - g(x_{k-m_k+i})] \\
=& g(x_k) + \sum_{i=0}^{m_k-1} \gamma_i^{(k)} g(x_{k-m_k+i}) - \sum_{i=1}^{m_k} \gamma_{i-1}^{(k)} g(x_{k-m_k+i}) \\
=& g(x_k) - \gamma_{m_k-1}^{(k)} g(x_k) + \gamma_0^{(k)} g(x_{k-m_k}) + \sum_{i=1}^{m_k-1} (\gamma_i^{(k)} - \gamma_{i-1}^{(k)}) g(x_{k-m_k+i}) \\
=& \alpha_{m_k}^{(k)} g(x_k) + \alpha_0^{(k)} g(x_{k-m_k}) + \sum_{i=1}^{m_k-1} \alpha_i^{(k)} g(x_{k-m_k+i}) \\
=& \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}) = x_{k+1},
\end{aligned} \tag{2.55}$$

where we use (2.49) in the third equality.

Chapter 3

Numerical Implementation

3.1 The Finite Difference Method and Implicit Euler

In this section we consider the differential equation

$$\begin{cases} U_t(x,t) = U_{xx}(x,t) + f(x), \quad x \in [0,1], \quad t > 0 \\ U(0,t) = U(1,t) = 0 \\ U(x,0) = U^0(x) \end{cases}, \tag{3.1}$$

for some given functions f and U^0, where the subscripts indicate with respect to what variable the derivative is taken. For some choices of f and U^0 it is rather simple to solve this problem analytically, but for other choices it can be extremely hard or even impossible, therefore it is natural to look for ways to find an approximate solution. In this section we will place our focus on the finite difference method, a way of approximating $U(x,t)$ through discretization, which is described below.

Firstly, we discretize the interval $[0,1]$ on the x-axis by only considering a vector of function evaluations of the points $\{x_0, ..., x_n\}$, where $x_i = i/n$ for $0 \leq i \leq n$, which we will denote with an underline. We also discretize t by only

considering $t_\ell = \ell\Delta t$ for $\ell \geq 0$, for some time step Δt, where a superscript will indicate which time step is considered. With the boundary condition $U(0,t) = U(1,t) = 0$, we can simplify by disregarding the endpoints and only consider $\{x_1, ..., x_{n-1}\}$. From this we get the discretization of the first line in (3.1):

$$\underline{U}_t^\ell = \underline{U}_{xx}^\ell + \underline{f}, \quad \ell \geq 0, \tag{3.2}$$

where $\underline{U}_t^\ell = (U_t(x_1, t_\ell), ..., U_t(x_{n-1}, t_\ell))^{\mathrm{T}}$, $\underline{U}_{xx}^\ell = (U_{xx}(x_1, t_\ell), ..., U_{xx}(x_{n-1}, t_\ell))^{\mathrm{T}}$ and $\underline{f} = (f(x_1), ..., f(x_{n-1}))^{\mathrm{T}}$.

Secondly, we approximate the first derivative of U with respect to x with the difference quotient

$$U_x(x_i, t_\ell) \approx \frac{U(x_{i+1}, t_\ell) - U(x_i, t_\ell)}{\Delta x}, \quad \begin{cases} 0 \leq i \leq n-1, \\ t_\ell \geq 0 \end{cases}, \tag{3.3}$$

where $\Delta x = 1/n$. By applying a similar process, we can get

$$U_{xx}(x_i, t_\ell) \approx \frac{U(x_{i+1}, t_\ell) - 2U(x_i, t_\ell) + U(x_{i-1}, t_\ell)}{\Delta x^2}, \quad \begin{cases} 1 \leq i \leq n-1, \\ t_\ell \geq 0 \end{cases}. \tag{3.4}$$

Moreover, the initial condition $U(0,t) = U(1,t) = 0$, implies $U_{xx}(x_1, t) = \frac{U(x_2,t_\ell) - 2U(x_1,t_\ell)}{\Delta x^2}$ and $U_{xx}(x_{n-1}, t_\ell) = \frac{-2U(x_{n-1},t_\ell) + U(x_{n-2},t_\ell)}{\Delta x^2}$. This gives us the following approximate discretization of U_{xx} in $\{x_1, ..., x_{n-1}\}$

$$\underline{U}_{xx}^\ell \approx A\underline{U}^\ell, \tag{3.5}$$

where

$$A = \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \vdots \\ 0 & 1 & -2 & \ddots & \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & & 1 & -2 \end{bmatrix}, \qquad A \in \mathbb{R}^{(n-1)\times(n-1)}. \tag{3.6}$$

and $\underline{U}^\ell = (U(x_1, t_\ell), ..., U(x_{n-1}, t_\ell))^{\mathrm{T}}$. This lets us approximate (3.2) by

$$\underline{U}_t^\ell - A\underline{U}^\ell = \underline{f}, \quad \ell \geq 0. \tag{3.7}$$

With this approximation we can apply the implicit Euler method over t, which leads to the the following iterative process for finding approximations of $\underline{U}^\ell$ at time t_ℓ

$$\underline{U}^{\ell+1} = \underline{U}^\ell + \Delta t(A\underline{U}^{\ell+1} + \underline{f}), \quad \ell \geq 0, \tag{3.8}$$

where $\underline{U}^0 = (U^0(x_1), ..., U^0(x_{n-1}))^{\mathrm{T}}$.

On closer inspection we can see that (3.8) is a fixed-point problem and by expanding and sorting the terms, we get the equivalent form

$$\underline{U}^{\ell+1} = (\Delta tA)\underline{U}^{\ell+1} + (\underline{U}^\ell + \Delta t\underline{f}). \tag{3.9}$$

Moreover, if at time t_ℓ, we let $M := \Delta tA$, $b := (\underline{U}^\ell + \Delta t\underline{f})$ and $g(u) \equiv Mu + b$, then we get the problem on the form (1.1)

$$\underline{U}^{\ell+1} = g(\underline{U}^{\ell+1}), \tag{3.10}$$

where g is a linear function. Furthermore, we also get the equivalent system of linear equations

$$(I - M)\underline{U}^{\ell+1} = b, \quad u \in \mathbb{R}^{n-1}. \tag{3.11}$$

Consequently, we can not only apply Anderson acceleration to problem (3.10), but also GMRES to the equivalent problem (3.11).

3.1.1 Implementation

Since the primary purpose of (3.8) to us was to test the relation between Algorithm AA and GMRES on a common problem, we only consider the case $\ell = 0$, since all following iterates could be obtained by solving a problem of the same form.

For problem (3.8), the three solution methods, fixed-point iteration, Anderson Acceleration, and GMRES, were all implemented in Python as methods of a class called FDM. The three inputs when creating an object of the class FDM were n, f and U^0. In the initialization of FDM, the following was created as

attributes of the class.

$$\begin{cases} M = \Delta t A \\ b = (\underline{U}^{\ell} + \underline{f}) \\ g(x) \equiv Mx + b \end{cases} \tag{3.12}$$

Additionally, three empty lists, called AAnormRes, fixPnormRes and GMRESnormRes were created as attributes, with the intention to store the relative norm of the residual; $\|r_k\|_2 = \|g(u_k) - u_k\|_2 / \|g(u_k)\|_2$, for iteration $k = 1, 2, ...$, where u_k was the iterate of the respective algorithms.

Inputs for all methods of the class were an initial guess of the fixed-point $\underline{U}^1$, and optional inputs "tol" for the tolerance, Δt for the size of the time step and "it" for maximum number of iterations, with default values 10^{-15}, 10^{-6} and 100 respectively. An additional input unique to Anderson Acceleration was m for truncation, with default value -1, which represented truncation being inactive.

Regarding how the three methods were implemented, GMRES was imported from the library scipy.sparse.linalg and the method in the class simply ran GMRES by providing the necessary inputs. Anderson acceleration was implemented as in Algorithm AA, except that (1.2) was replaced with the unconstrained form (2.48) and the update of the iterates was given by (2.54). Fixed-point iteration simply followed Algorithm FPI.

All of the algorithms checked whether $\|r_k\|_2 <$ tol, at the end of every iteration to determine whether or not convergence had been reached, in which case they could declare success. Moreover, $\|r_k\|_2$ was stored in AAnormRes, fixPnormRes and GMRESnormRes for the respective algorithm. Furthermore, the methods for both Anderson Acceleration and GMRES returned all of the iterates and additionally, unique to the method for Anderson acceleration, was that it also returned the list of the solutions γ_k to (2.48).

3.2 Results

Problem (3.8) at time step $\ell = 0$ was solved using the methods of the class FDM, for two different pairs of U^0 and f, first

$$\begin{cases} f(x) \equiv \sin^2(x) \\ U^0(x) \equiv \sin(\pi x) \end{cases} \tag{3.13}$$

and then

$$\begin{cases} f(x) \equiv \cos^2(x) \\ U^0(x) \equiv x^3 - \frac{3}{2}x^2 + \frac{1}{2}x \end{cases}, \tag{3.14}$$

where the initial guess for $\underline{U}^1 = u_0$ was chosen as the zero vector, note that both functions satisfy the initial conditions. Both cases were tested with both $n = 500$ and $n = 1000$. Anderson acceleration was implemented on these cases both with and without truncation. For "tol", Δt and "it", the default values 10^{-15}, 10^{-6} and 100 were used respectively. Plots of the relative norm of the residuals against iteration number for the methods implemented on the 4 cases can be seen in Figures, 3.1, 3.2, 3.3, and 3.2.

Lastly, the returns from the methods for Anderson acceleration and GMRES were stored and then γ_k was translated to α_k using relation (2.49). Theorem 2.3 could then be directly tested and for $n = 500$, the relations held true to an order of about 10^{-16} for the first 10 iterations, which is very close to machine precision, but when iteration 20 had been reached, the relations were on the order of 10^{-14}, likely due to round-off errors. For $n = 1000$, the relations held true to an order of about 10^{-16} for the first few iterations, but at iteration 10 the inaccuracy had already reached an order of about 10^{-14}, which is not too surprising, since we expect worse approximation errors when dealing with larger matrices.

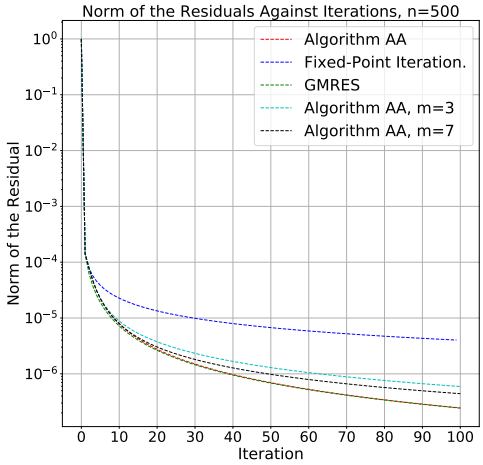

Figure 3.1: The relative norm of the residuals at each iteration for different algorithms implemented on (3.8) for $\ell = 0$. Here we have $n = 500$, and the conditions (3.13). Algorithm AA can be seen three times, since it was implemented both without truncation and with m set to 3 and 7.

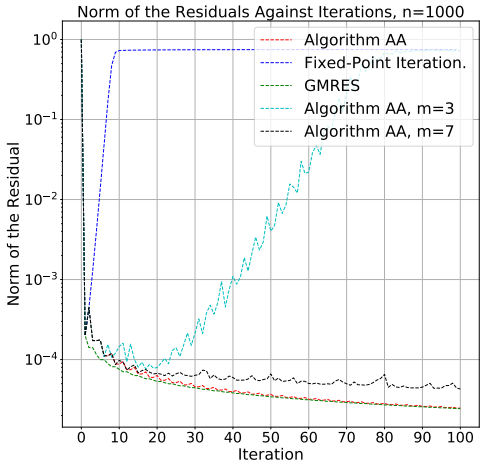

Figure 3.2: The relative norm of the residuals at each iteration for different algorithms implemented on (3.8) for $\ell = 0$. Here we have $n = 1000$, and the conditions (3.13). Algorithm AA can be seen three times, since it was implemented both without truncation and with m set to 3 and 7.

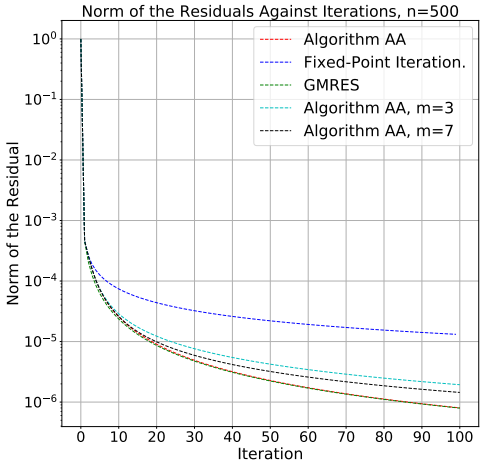

Figure 3.3: The relative norm of the residuals at each iteration for different algorithms implemented on (3.8) for $\ell = 0$. Here we have $n = 500$, and the conditions (3.14). Algorithm AA can be seen three times, since it was implemented both without truncation and with m set to 3 and 7.

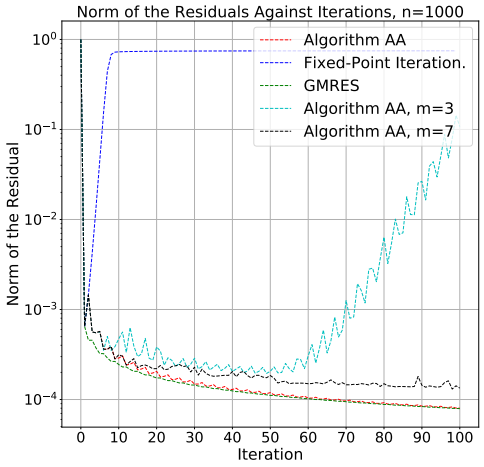

Figure 3.4: The relative norm of the residuals at each iteration for different algorithms implemented on (3.8) for $\ell = 0$. Here we have $n = 1000$, and the conditions (3.14). Algorithm AA can be seen three times, since it was implemented both without truncation and with m set to 3 and 7.

3.3 Discussion

Firstly, the theory of Section 2.2, more specifically Theorem 2.3, was supported by the numerical implementation since the relations of the theorem held true very close to machine precision.

Secondly, we observe Figures 3.1, 3.2, 3.3 and 3.4. We see that in all of the examples, the algorithms terminate either because of divergence or because the maximum number of iterations was reached, but since this is a system without a preconditioner, GMRES is expected to converge very slowly and therefore these graphs seem reasonable.

In all of the figures, we can observe the close relationship between the iterates of Algorithm AA and GMRES and see that the algorithms had nearly identical convergence rates, which was far superior to that of fixed-point iteration. This is great since one of the main issues of fixed-point iteration is that the convergence rate can be too slow.

An important observation is that in Figures 3.2 and 3.4, the iterations of GMRES seem to be very close to stagnation, but surprisingly, the stopping criterion $\left\|x_{k+1}^{\mathrm{AA}} - x_k^{\mathrm{AA}}\right\|_2 < \epsilon$, implemented to safeguard against the numerical weakness mentioned at the end of Section 2.2, was never close to being reached.

In the same figures, we can see that Anderson acceleration was convergent while fixed-point iteration was not, which shows potential to deal with the other main weakness of fixed-point iteration. However, whether or not Anderson Acceleration can relieve fixed-point iteration of its issues in any nonlinear cases, is beyond the scope of this thesis.

As for truncated Anderson acceleration, we see that it converged significantly faster than fixed-point iteration in Figures 3.1 and 3.3, even for small choices of m. In Figures 3.2 and 3.4 we can see that it only seemed convergent for $m = 7$. However, to figure whether or not it is actually convergent in these cases, one would have to run many more iterations, since we can see that for $m = 3$ it appears convergent for the first 20-50 iterations, but then diverges.

To summarize, we have shown that in the linear case, Anderson acceleration is, under some additional assumptions, equivalent to GMRES. Consequently, Anderson acceleration is somewhat of a generalization of GMRES to nonlinear cases. Moreover, we implemented Anderson acceleration numerically and were able to obtain iterates for which the relations held true very close to machine precision.

Bibliography

[1] O. Kosheleva. "Babylonian method of computing the square root: Justifications based on fuzzy techniques and on computational complexity". *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*. 2009, pp. 1–6. DOI: 10.1109/NAFIPS.2009.5156463.

[2] Youcef Saad and Martin H Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". *SIAM J. Sci. Stat. Comput.* 7.3 (1986), pp. 856–869.

[3] Homer Walker and Peng Ni. "Anderson Acceleration for Fixed-Point Iterations". *SIAM J. Numerical Analysis* 49 (2011), pp. 1715–1735. DOI: 10.2307/23074353.

Bachelor's Theses in Mathematical Sciences 2020:K17

ISSN 1654-6229

LUNFNA-4033-2020

Numerical Analysis

Centre for Mathematical Sciences

Lund University

Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/