

Project Report for Seminar Course in Numerical Analysis, VT23

Junzi Zhang, Brendan O'Donoghue, Stephen Boyd: Globally Convergent
Type-I Anderson Acceleration for Non-Smooth Fixed-Point Iterations

Theo Koppenhöfer

Lund

April 29, 2023

Input : Initial value $x_0 \in \mathbb{R}^n$ and function $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

```
for  $k = 0, 1, \dots$  do  
  | Set  $x_{k+1} = f(x_k)$ .  
end
```

Algorithm 1: Fixed point iteration (original)

Introduction

The following report will summarise and present the main results of the paper [1] as part of the course NUMN27, Seminar in Numerical Analysis. More precisely the report will deal with a modification of the Anderson-Accelleration-I (AA-I) algorithm as described in [1]. For this we start with a motivation of the AA-II algorithm. Then we will discuss the modifications made to the algorithm. After that we will give the main convergence result and finally test the AA-I algorithm in numerical experiments. The report, the python implementation and the corresponding presentation of the topic can be found online under [3].

Motivation of the Anderson-Acceleration algorithm

In science it is a common problem to find a fixed point $x = f(x) \in \mathbb{R}^n$ of a function $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$. An equivalent formulation is to find a zero $x \in \mathbb{R}^n$ of the function $g = \text{Id} - f$. In the following we assume that f indeed has a fixed point and that f is non-expansive, i.e. we have for all $x, y \in \mathbb{R}^n$ that

$$\|f(x) - f(y)\| \leq \|x - y\|.$$

We however also assume that ∇f is unknown which means we cannot use Newton iteration to solve our problem. We also assume that our problem is noisy so that we cannot take finite difference derivatives. If the cost of evaluating f is very high then line search becomes infeasible and if n is very large we are forced to work matrix free. We know that this type of problem can be solved by the fixed point iteration described in algorithm 1.

It can be shown that the fixed point iteration will in this case converge to a fixed point of f . This convergence can be very slow in practice if the Lipshitz-constant of f is close to 1. It is here where the Anderson-Acceleration (AA) methods come in.

The main idea of the AA is to use the information gained from previous function evaluations. To update x_{k+1} we form a weighted average as described in algorithm 2. For simplicity of notation we assume here and in the following that our memory is unlimited.

The General AA leaves the question open of how to choose the $\alpha \in \mathbb{R}^{k+1}$. Since finding a fixed point of f is equivalent to finding a zero of $f = \text{Id} - f$ it seems sensible to require α to minimise

$$\left\| \sum_i \alpha_i g_i \right\|_2.$$

Input: $x_0 \in \mathbb{R}^n$ and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

```

for  $k = 0, 1, \dots$  do
    Set  $f_k = f(x_k)$ .
    Choose  $\alpha = \alpha^k \in \mathbb{R}^{k+1}$  such that  $\sum_i \alpha_i = 1$ .
    Set  $x_{k+1} = \sum_i \alpha_i f_i$ .
end

```

Algorithm 2: General AA (Anderson Acceleration)

Input: $x_0 \in \mathbb{R}^n$ and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

```

for  $k = 0, 1, \dots$  do
    Set  $f_k = f(x_k)$ .
    Set  $g_k = x_k - f_k$ .
    Choose  $\alpha \in \mathbb{R}^{k+1}$  such that  $\sum_i \alpha_i = 1$  and such that  $\alpha$  minimises  $\|\sum_i \alpha_i g_i\|_2$ .
    Set  $x_{k+1} = \sum_i \alpha_i f_i$ .
end

```

Algorithm 3: AA-II

This then yields the AA-II algorithm given in 3.

It is shown in [1] that this update can be brought into the form of a quasi-Newton-like method as shown in algorithm 4.

Now one could expect H_k to be an approximate inverse of $\nabla f(x_k)$. Indeed one can show

Proposition 1 (Approximate inverse Jacobian). *H_k minimises $\|H_k - \text{Id}\|_F$ under the multi-secant condition $H_k S_k = Y_k$.*

Proof. See [1]. □

The good Broyden method approximates the Jacobian rather than its inverse and tends to yield better results. This motivates to choose B_k to be a minimiser of $\|B_k - \text{Id}\|_F$

Input: $x_0 \in \mathbb{R}^n$ and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Set $x_1 = f(x_0)$.

```

for  $k = 0, 1, \dots$  do
    Set  $g_k = g(x_k)$ .
    Construct  $S_k = [x_1 - x_0 \quad \dots \quad x_k - x_{k-1}] \in \mathbb{R}^{n \times k}$  and
     $Y_k = [g_1 - g_0 \quad \dots \quad g_k - g_{k-1}] \in \mathbb{R}^{n \times k}$ .
    Set  $H_k = \text{Id} + (S_k - Y_k)(Y_k^\top Y_k)^{-1} Y_k^\top \in \mathbb{R}^{n \times n}$ .
    Set  $x_{k+1} = x_k - H_k g_k$ .
end

```

Algorithm 4: AA-II (reformulated)

Input : $x_0 \in \mathbb{R}^n$ and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Set $x_1 = f(x_0)$

for $k = 0, 1, \dots$ **do**

 Set $g_k = g(x_k)$.

 Construct S_k from x_0, \dots, x_k and Y_k from g_0, \dots, g_k .

 Set $B_k = \text{Id} + (Y_k - S_k)(S_k^\top S_k)^{-1} S_k^\top \in \mathbb{R}^{n \times n}$.

 Set $H_k = B_k^{-1}$.

 Set $x_{k+1} = x_k - H_k g_k$.

end

Algorithm 5: AA-I

under the condition $B_k Y_k = S_k$. One can then show that

$$B_k = \text{Id} + (Y_k - S_k) \left(S_k^\top S_k \right)^{-1} S_k^\top.$$

which yields the AA-I algorithm 5.

Modifications to AA-I

The AA-I algorithm as stated in 5 has some apparent problems. For one, the approach is not matrix-free. This will be fixed by a rank-1 update formula for matrices B_k and later H_k . It may also occur in a step that the matrix H_k is not well-defined. This may occur if B_k itself is not well-defined or is singular. The well-definedness will be resolved by the Powell-type regularisation and the restarting of the iteration. The restarting of the iteration will also yield an algorithm that does not require unlimited memory. Lastly, this algorithm does not have the convergence property of the fixed point iteration algorithm. This will be resolved by the safeguarding of steps.

One can show that

Proposition 2 (Rank-1 update for B_k). *We have*

$$B_k = B_{k-1} + \frac{(y_{k-1} - B_{k-1} s_{k-1}) \hat{s}_{k-1}^\top}{\hat{s}_{k-1}^\top s_{k-1}}$$

where $y_{k-1} = g_k - g_{k-1}$, $B_0 = \text{Id}$ and

$$\hat{s}_{k-1} = s_{k-1} - \sum_{j=0}^{k-2} \frac{\hat{s}_j^\top s_{k-1}}{\|\hat{s}_j\|^2} \hat{s}_j$$

is the Gram-Schmidt orthogonalisation of $s_{k-1} = x_k - x_{k-1}$.

Proof. See [1]. □

Input : $x_0 \in \mathbb{R}^n$ and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.
Set $B_0 = \text{Id}$ and $x_1 = f(x_0)$.
for $k = 0, 1, \dots$ **do**
 Set $g_k = g(x_k)$.
 Set $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$ and $\hat{s}_{k-1} = s_{k-1} - \sum_{i=0}^{k-2} \frac{\hat{s}_i^\top s_{k-1}}{\|\hat{s}_i\|^2} s_i$.
 Set $B_k = B_{k-1} + \frac{(y_{k-1} - B_{k-1} s_{k-1}) \hat{s}_{k-1}^\top}{\hat{s}_{k-1}^\top s_{k-1}}$.
 Set $H_k = B_k^{-1}$.
 Set $x_{k+1} = x_k - H_k g_k$.
end

Algorithm 6: AA-I (rank-1 update)

Well-definedness of H_k : Powell-type regularisation

To fix the singularity of B_k we use Powell-type regularisation.

Input : $x^0 \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\bar{\theta} \in (0, 1)$.
Set $B_0 = \text{Id}$ and $x_1 = f(x_0)$.
for $k = 0, 1, \dots$ **do**
 Set $g_k = g(x_k)$, $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.
 Set $\hat{s}_{k-1} = s_{k-1} - \sum_{i=0}^{k-2} \frac{\hat{s}_i^\top s_{k-1}}{\|\hat{s}_i\|^2} s_i$.
 Choose θ_{k-1} in dependence of $\bar{\theta}$.
 Set $\tilde{y}_{k-1} = \theta_{k-1} y_{k-1} - (1 - \theta_{k-1}) g_{k-1}$.
 Set $B_k = B_{k-1} + \frac{(\tilde{y}_{k-1} - B_{k-1} s_{k-1}) \hat{s}_{k-1}^\top}{\hat{s}_{k-1}^\top s_{k-1}}$.
 Set $H_k = B_k^{-1}$.
 Set $x_{k+1} = x_k - H_k g_k$.
end

Algorithm 7: AA-I with Powell-type regularisation

Well-definedness of H_k , memory usage: Restarting iteration

If $\hat{s}_k = 0$ the update

$$B_k = B_{k-1} + \frac{(\tilde{y}_{k-1} - B_{k-1} s_{k-1}) \hat{s}_{k-1}^\top}{\hat{s}_{k-1}^\top s_{k-1}}$$

is ill-defined. This occurs in algorithm 7 e.g. for $k > n$ as then $\hat{s}_k = 0$ by linear dependence. Hence we restart the algorithm with x_k as the new starting point if

- $k = m + 1$ for some fixed $m \in \mathbb{N}$ or
- $\|\hat{s}_{k-1}\| < \tau \|s_{k-1}\|$ for some fixed $\tau \in (0, 1)$.

It can be shown that B_k is then well-defined.

Input : $x^0 \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $m \in \mathbb{N}$ and $\bar{\theta}, \tau \in (0, 1)$

Set $B_0 = \text{Id}$, $x_1 = f(x_0)$ and $m_0 = 0$.

for $k = 0, 1, \dots$ **do**

Set $g_k = g(x_k)$, $m_k = m_{k-1} + 1$, $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

Set $\hat{s}_{k-1} = s_{k-1} - \sum_{i=k-m_k}^{k-2} \frac{\hat{s}_i^\top s_{k-1}}{\|\hat{s}_i\|^2} s_i$.

if $m_k = m + 1$ **or** $\|\hat{s}_{k-1}\| < \tau \|s_{k-1}\|$ **then**

 Set $m_k = 0$, $\hat{s}_{k-1} = s_{k-1}$ and $B_{k-1} = \text{Id}$.

end

Choose θ_{k-1} in dependence of $\bar{\theta}$.

Set $\tilde{y}_{k-1} = \theta_{k-1} y_{k-1} - (1 - \theta_{k-1}) g_{k-1}$.

Set $B_k = B_{k-1} + \frac{(\tilde{y}_{k-1} - B_{k-1} s_{k-1}) \hat{s}_{k-1}^\top}{\hat{s}_{k-1}^\top s_{k-1}}$.

Set $H_k = B_k^{-1}$.

Set $x_{k+1} = x_k - H_k g_k$.

end

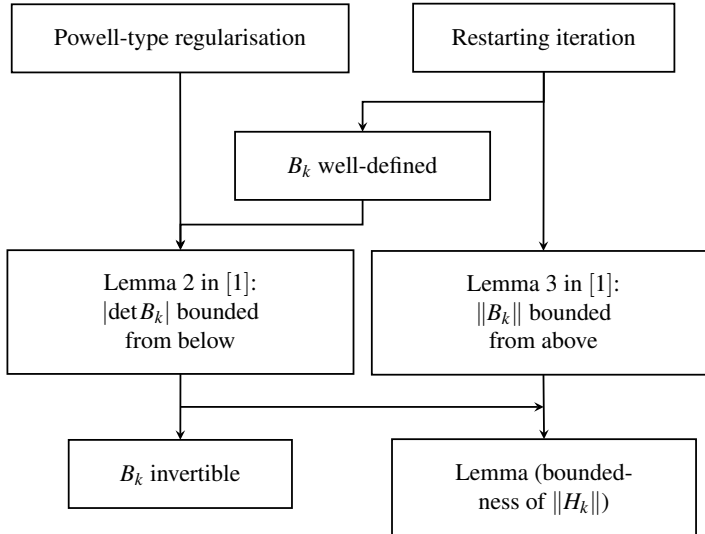
Algorithm 8: AA-I with Powell-type regularisation and Restarting

One can then show

Lemma 3 (bound on $\|H_k\|_2$). *In algorithm 8 we have that H_k is well-defined and there exists a constant $c_1 = c_1(m, n, \bar{\theta}, \tau) > 0$ such that*

$$\|H_k\|_2 \leq c_1.$$

Proof. See [1, Corollary 4]. □



Computational efficiency: Rank-1 update for H_k

From the Sherman-Morrison formula one can obtain

Proposition 4 (Rank-1 update for H_k). *We have*

$$H_k = H_{k-1} + \frac{(s_{k-1} - H_{k-1}y_{k-1})\hat{s}_{k-1}^\top H_{k-1}}{\hat{s}_{k-1}^\top H_{k-1}y_{k-1}}$$

Input : $x^0 \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $m \in \mathbb{N}$ and $\bar{\theta}, \tau \in (0, 1)$

Set $H_0 = \text{Id}$, $x_1 = f(x_0)$ and $m_0 = 0$.

for $k = 0, 1, \dots$ **do**

 Set $g_k = g(x_k)$, $m_k = m_{k-1} + 1$, $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

 Set $\hat{s}_{k-1} = s_{k-1} - \sum_{i=k-m_k}^{k-2} \frac{\hat{s}_i^\top s_{k-1}}{\|\hat{s}_i\|^2} s_i$.

if $m_k = m + 1$ **or** $\|\hat{s}_{k-1}\| < \tau \|s_{k-1}\|$ **then**

 | Set $m_k = 0$, $\hat{s}_{k-1} = s_{k-1}$ and $H_{k-1} = \text{Id}$.

end

 Choose θ_{k-1} in dependence of $\bar{\theta}$.

 Set $\tilde{y}_{k-1} = \theta_{k-1}y_{k-1} - (1 - \theta_{k-1})g_{k-1}$.

 Set $H_k = H_{k-1} + \frac{(s_{k-1} - H_{k-1}\tilde{y}_{k-1})\hat{s}_{k-1}^\top H_{k-1}}{\hat{s}_{k-1}^\top H_{k-1}\tilde{y}_{k-1}}$.

 Set $x_{k+1} = x_k - H_k g_k$.

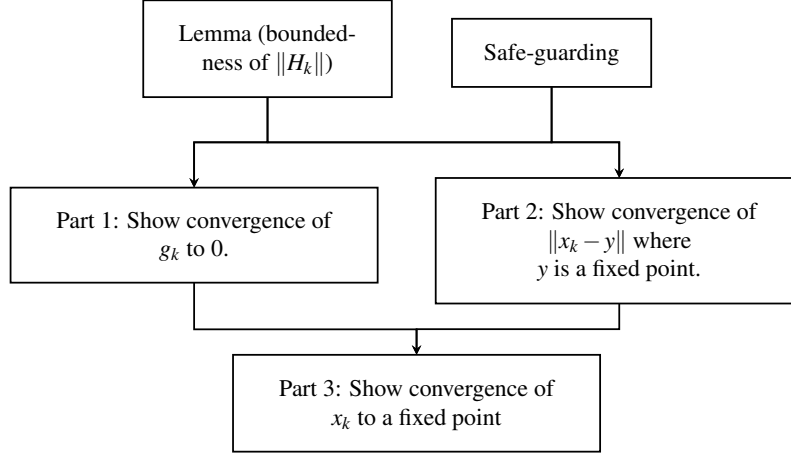
end

Algorithm 9: AA-I with Powell-type regularisation and Restarting

Convergence: Safeguarding steps

To guarantee the decrease in $\|g_k\|$ one can interleave the AA-I steps with Krasnosel'skii-Mann (KM) steps which are given by

$$x_{k+1} = (1 - \alpha)x_k + \alpha f_k$$



for some fixed $\alpha \in (0, 1)$.

Input : $x^0 \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $m \in \mathbb{N}$, $\bar{\theta}$, τ , $\alpha \in (0, 1)$ and safe-guarding constants $D, \varepsilon > 0$
 Set $H_0 = \text{Id}$, $x_1 = \bar{x}_1 = f(x_0)$, $m_0 = n_{AA} = 0$ and $\bar{U} = \|g_0\|_2$.
for $k = 0, 1, \dots$ **do**
 Set $g_k = g(x_k)$, $m_k = m_{k-1} + 1$, $s_{k-1} = \bar{x}_k - x_{k-1}$ and $y_{k-1} = g(\bar{x}_k) - g_{k-1}$.
 Set $\hat{s}_{k-1} = s_{k-1} - \sum_{i=k-m_k}^{k-2} \frac{\hat{s}_i^\top s_{k-1}}{\|\hat{s}_i\|^2} s_i$.
 if $m_k = m + 1$ **or** $\|\hat{s}_{k-1}\| < \tau \|s_{k-1}\|$ **then**
 Set $m_k = 0$, $\hat{s}_{k-1} = s_{k-1}$ and $H_{k-1} = \text{Id}$.
 end
 Choose θ_{k-1} in dependence of $\bar{\theta}$.
 Set $\tilde{y}_{k-1} = \theta_{k-1} y_{k-1} - (1 - \theta_{k-1}) g_{k-1}$.
 Set $H_k = H_{k-1} + \frac{(s_{k-1} - H_{k-1} \tilde{y}_{k-1}) s_{k-1}^\top H_{k-1}}{\hat{s}_{k-1}^\top H_{k-1} \tilde{y}_{k-1}}$ and $\tilde{x}_{k+1} = x_k - H_k g_k$.
 if $\|g_k\| \leq D \bar{U} (n_{AA} + 1)^{-(1+\varepsilon)}$ **then**
 Set $x_{k+1} = \tilde{x}_{k+1}$ and $n_{AA} = n_{AA} + 1$.
 else
 Set $x_{k+1} = (1 - \alpha)x_k + \alpha f_k$.
 end
end

Algorithm 10: AA-I with Powell-type regularisation, restarting and safeguarding

Convergence result

Theorem 5 (Convergence). *Let x_k be generated by algorithm 10 then x_k converges to a fixed point of f .*

Proof. The proof follows [1, Theorem 6]. In the first part we use lemma ?? on the boundedness of $\|H_k\|$ and the safe-guarding step to show the convergence of g_k to 0. In part 2 we also use lemma ?? and the safe-guarding to show convergence of $a_k = \|x_k - y\|^2$ to some $a \in \mathbb{R}$. Here y denotes a fixed point of f . In the third part we use parts 1 and 2 to show that x_k converges to a fixed point.

Part 1. We partition $\mathbb{N} = K_{AA} \sqcup K_{KM}$ where $K_{AA} = \{m_0, m_1, \dots\}$ denote the indices k where the algorithm chose an AA-step (a) and $K_{KM} = \{n_0, n_1, \dots\}$ where the algorithm chose a KM-step (b).

if $\|g_k\| \leq D\bar{U}(n_{AA} + 1)^{-(1+\varepsilon)}$ **then**
 Set $x_{k+1} = \tilde{x}_{k+1}$ and $n_{AA} = n_{AA} + 1$. (a)

else
 Set $x_{k+1} = (1 - \alpha)x_k + \alpha f_k$. (b)

end

Algorithm 11: The two cases for x_{k+1} .

Let y be a fixed point of f . We distinguish the cases

case (a) if $k_i \in K_{AA}$ then

$$\begin{aligned} \|x_{k_i+1} - y\| &\leq \|x_{k_i} - y\| + \|H_{k_i}g_{k_i}\| \\ &\leq \|x_{k_i} - y\| + c_1\|g_k\| \\ &\leq \|x_{k_i} - y\| + c_2(i+1)^{-(1+\varepsilon)} \end{aligned} \tag{1}$$

for some constant $c_2 > 0$.

case (b) if $l_i \in K_{KM}$ then one can show (see [1, Theorem 6])

$$\|x_{l_i+1} - y\|^2 \leq \|x_{l_i} - y\|^2 - \alpha(1 - \alpha)\|g_{l_i}\|^2. \tag{2}$$

Here one uses the non-expansiveness of f and that y is a fixed point.

Hence we have in any case

$$\|x_k - y\| \leq \|x_0 - y\| + c_2 \sum_i (i+1)^{-(1+\varepsilon)} = c_3 < \infty.$$

It then follows that

$$\begin{aligned} a_{k_i+1} &= \|x_{k_i+1} - y\|^2 \stackrel{(1),(2)}{\leq} \left(\|x_{k_i} - y\| + c_2(i+1)^{-(1+\varepsilon)} \right)^2 \\ &\leq \underbrace{\|x_{k_i} - y\|^2}_{=a_{k_i}} + \underbrace{c_2^2(i+1)^{-2(1+\varepsilon)} + 2c_2 \overbrace{\|x_{k_i} - y\|}^{\leq c_3} (i+1)^{-(1+\varepsilon)}}_{=b_{k_i}} \\ &= a_{k_i} + b_{k_i} \end{aligned} \tag{3}$$

and thus

$$\alpha(1 - \alpha) \sum_i \|g_{l_i}\|^2 \stackrel{(2)}{\leq} \sum_i a_{l_i} - a_{l_i+1} \stackrel{(3)}{\leq} a_0 + \sum_k b_k < \infty.$$

We therefore have $\lim_i \|g_{l_i}\| = 0$. It also follows from $\|g_{k_i}\| \leq D\bar{U}(i+1)^{-(1+\varepsilon)}$ that $\lim_i \|g_{k_i}\| = 0$. As all subsequences of g_k converge to 0 we thus have that g_k converges to 0.

Part 2. Let now a_{k_i} and a_{l_i} be subsequences such that $k_i \leq l_i$ and

$$a_{k_i} \xrightarrow{j \rightarrow \infty} \liminf_k a_k = \underline{a} \quad \text{and} \quad a_{l_i} \xrightarrow{j \rightarrow \infty} \limsup_k a_k = \bar{a}.$$

It then follows that

$$a_{l_i} - a_{k_i} = \sum_{k=k_i}^{l_i-1} a_{k+1} - a_k \stackrel{(3)}{\leq} \sum_{k=k_i}^{\infty} b_k$$

and when taking $l_i \rightarrow \infty$

$$\bar{a} - a_{k_i} \leq \sum_{k=k_i}^{\infty} b_k$$

and then taking $k_i \rightarrow \infty$ we get

$$\bar{a} - \underline{a} \leq 0.$$

Thus

$$\limsup_k a_k = \bar{a} \leq \underline{a} = \liminf_k a_k$$

and so $a_k = \|x_k - y\|^2$ converges to some $a \in \mathbb{R}$.

Part 3. Let k_i and l_i now be convergent subsequences of x_k which converge to x and \tilde{x} respectively. Since by continuity of g

$$\|g(x)\| = \lim_i \|g(x_{k_i})\| \stackrel{\text{part 1}}{=} 0$$

we have that x is a fixed point and analogously \tilde{x} is too. Now we have

$$\|x_{k_i}\|^2 = \|x_{k_i} - y\|^2 - \|y\|^2 + 2y^\top x_{k_i}$$

and by part 2 we obtain when taking $i \rightarrow \infty$

$$\|x\|^2 = a - \|y\|^2 + 2y^\top x$$

Analogously we obtain

$$\|\tilde{x}\|^2 = a - \|y\|^2 + 2y^\top \tilde{x}$$

which implies

$$2y^\top (x - \tilde{x}) = \|x\|^2 - \|\tilde{x}\|^2.$$

As x and \tilde{x} are fixed points it follows for $y \in \{x, \tilde{x}\}$ that

$$x^\top (x - \tilde{x}) = \tilde{x}^\top (x - \tilde{x})$$

and further

$$(x - \tilde{x})^\top (x - \tilde{x}) = 0.$$

We thus have $x = \tilde{x}$. We have shown that two convergent subsequences of x_k have the same limit and hence x_k is convergent and the limit must be a fixed point of f . □

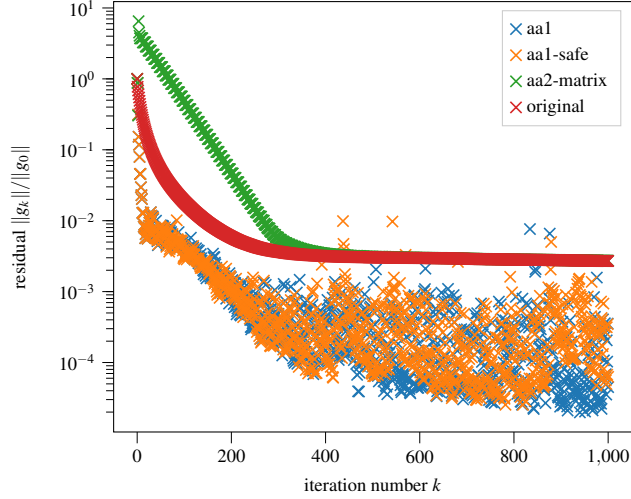


Figure 1: Residual norms for the elastic net regression problem.

Numerical experiments

As part of the project some numerical experiments from [1] were replicated. Thus the obtained results are the same as in [1]. The aim of the experiments is to test the numerical performance of the algorithms. For this f is chosen so that the Lipschitz constant is close to 1. This is precisely the type of problem for which the AA algorithm was developed.

Elastic net regression

In the first experiment f originates from an elastic net regression problem and is motivated in [1, Section 5.1f]. Specifically one obtains

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad x \mapsto S_{\alpha\mu/2} \left(x - \alpha \left(A^\top (Ax - b) + \frac{\mu}{2} x \right) \right)$$

with shrinkage operator

$$S_\kappa(x) = (\text{sgn}(x_i)(|x_i| - \kappa)_+)_{i=1}^n$$

and $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and some $\alpha, \mu \in \mathbb{R}$. Here we choose the parameters as in [1, Section 5.2]. In particular we choose $m = 500$ and $n = 1000$ and A , b and x^0 to be randomly generated.

The results for the different methods can be seen in Figure 1. Here the method 'original' is the fixed point iteration, i.e. algorithm ???. The method 'aa1-safe' is the AA-I algorithm with Powell-type regularisation, restarting and safeguarding. The 'aa2-matrix' algorithm is an implementation of the AA-II algorithm where 'matrix' indicates that the implementation is not matrix-free. We see little difference between the 'aa1' and the

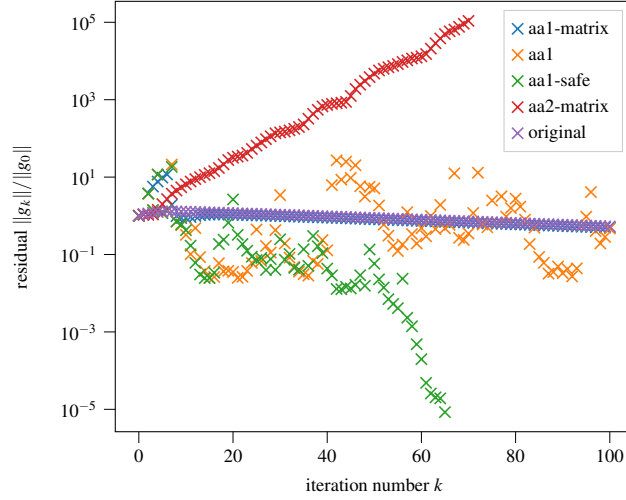


Figure 2: Residual norms for the Markov decision process problem.

'aa1-safe' methods. Both methods however outperform the 'original' and 'aa2-matrix' methods for this problem.

Markov decision process

In a second experiment f originates from a random Markov decision process which is motivated in [1, Section 5.1f]. Here our aim is to find a fixed point of the Bellman operator

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad x \mapsto \left(\max_a \left(R_{sa} + \gamma \sum_{s'} P_{sas'} x_{s'} \right) \right)_{s=1}^n$$

with some $R \in \mathbb{R}^{S \times A}$, $P \in \mathbb{R}^{S \times A \times A}$ and $\gamma \in \mathbb{R}$. Here the parameter γ determines the Lipschitz-constant of f . Again, we choose the parameters as in [1, Section 5.2]. In particular we have $n = 1000$, $A = 200$, $S = 300$ and A and R to be randomly generated.

The performance for the various methods can be seen in figure 2. In contrast to the 'aa1-matrix' method the 'aa2-matrix' method does not converge here. In this problem the fixed point iteration ('original') converges very slowly. The 'aa1-safe' method outperforms all others. This confirms numerically that the 'aa1-safe' algorithm can deal with the problems it was specifically designed for and for which the fixed point iteration fails.

In figure 3 we see how the performance of the 'aa1-safe' method depends on the memory parameter m . In particular one sees that the algorithm performs best for this problem with a parameter of $m \approx 10$. We also see that increasing the parameter m does not necessarily improve performance of the method as in this plot the choice $m = 50$ performs worst.

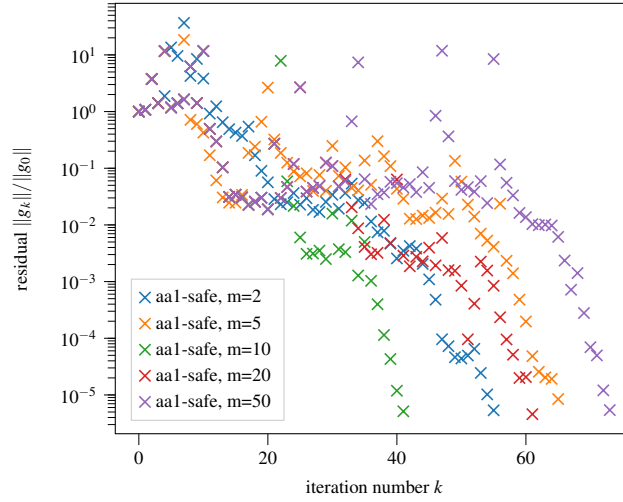


Figure 3: Residual norms for the Markov decision process problem.

Summary

The AA-I algorithm is specifically tailored to find the fixed point of a function f which is expensive to evaluate, noisy, has an unknown gradient and where the dimension n is large. The main idea of the AA-I and AA-II algorithms is to generalise the fixed point iteration by setting $x_{k+1} = \sum_i \alpha_i f_i$ for some clever choice of $\alpha = \alpha^k \in \mathbb{R}^{k+1}$. The AA-I algorithm one obtains requires some modifications. More specifically, one applies Powell-type regularisation and a restarting of the iteration for well-definedness. One builds in a mechanism for safeguarding of the steps for convergence and one uses a rank-1 update formula to make the implementation matrix-free. One can then show the convergence of the algorithm under the assumption that f is non-expansive and that there exists an algorithm. The numerical experiments then show that the AA-I algorithm with the modifications outperforms the fixed point iteration for the problems tested.

Bibliography

Main source

- [1] J. Zhang, B. O’Donoghue, and S. Boyd, “Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations,” *SIAM J. Optim.*, vol. 30, no. 4, pp. 3170–3197, 2020, ISSN: 1052-6234. DOI: 10.1137/18M1232772. [Online]. Available: <https://doi-org.ludwig.lub.lu.se/10.1137/18M1232772>.

Other sources

- [2] H.-r. Fang and Y. Saad, “Two classes of multisecant methods for nonlinear acceleration,” *Numer. Linear Algebra Appl.*, vol. 16, no. 3, pp. 197–221, 2009, ISSN: 1070-5325. DOI: 10.1002/nla.617. [Online]. Available: <https://doi-org.ludwig.lub.lu.se/10.1002/nla.617>.
- [3] numerics-seminar-VT23, *Github repository to the project*. Online, 2023. [Online]. Available: <https://github.com/TheoKoppenhoefer/numerics-seminar-VT23>.