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

Theo Koppenhöfer

Lund

April 16, 2023

Table of contents

The problem setting

Problem (find fixed point)

Find a fixed point $x \in \mathbb{R}^n$ *of* $f : \mathbb{R}^n \to \mathbb{R}^n$*, i.e.* $x = f(x)$*.*

or equivalently

Problem (find zero)

Find a zero $x \in \mathbb{R}^n$ *of* $g = \mathrm{Id} - f$*, i.e.* $0 = g(x)$*.*

We also assume

- f has a fixed point.
- f is nonexpansive, i.e. $\|f(x) - f(y)\| \leq \|x - y\|$.
- n is large $\rightarrow$matrix-free
- ∇f is unknown $\rightarrow$no Newton
- noisy problem $\rightarrow$no finite difference derivatives
- cost of evaluating f is high $\rightarrow$no line search

Fixed point iteration

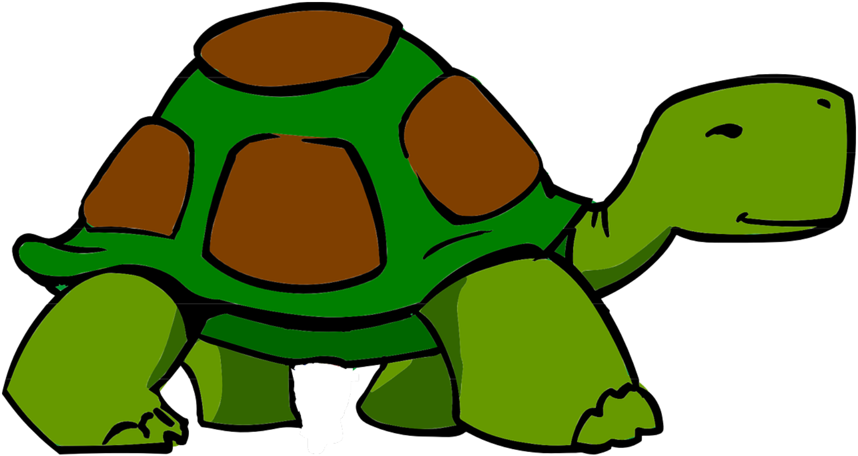
To keep things simple we try

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

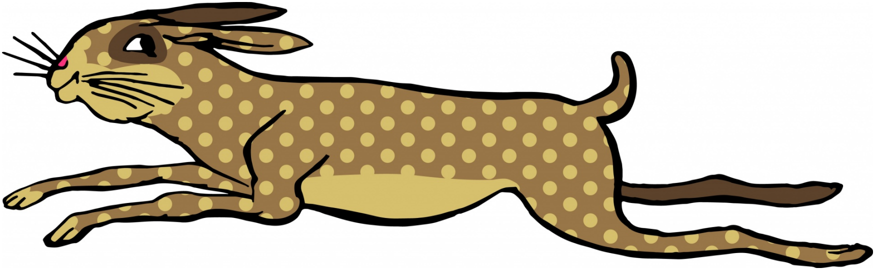
for $k = 0, 1, \ldots$ **do**
  Set $x_{k+1} = f(x_k)$.
end

Algorithm 1: Fixed point iteration (original)

This works, but ...



We want to be like...



General AA

We may as well use the information gained from previous evaluations. In the following we assume for simplicity that our memory is unlimited. If we form a weighted average we get

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

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

Algorithm 2: General AA (Anderson Acceleration)

AA-II

Since finding a fixed point of f is equivalent to finding a zero of $g = \mathrm{Id} - f$ the following seems like a good idea

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

for $k = 0, 1, \ldots$ **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 α minimises $\|\sum_i \alpha_i g_i\|_2$.

   Set $x_{k+1} = \sum_i \alpha_i f_i$.

end

Algorithm 3: AA-II

AA-II (reformulated)

One can show that this can be brought into the form of a quasi-Newton-like method

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

Set $x_1 = f(x_0)$.
for $k = 0, 1, \ldots$ **do**
    Set $g_k = g(x_k)$.
    Construct $S_k = \begin{bmatrix} x_1 - x_0 & \cdots & x_k - x_{k-1} \end{bmatrix} \in \mathbb{R}^{n\times k}$ and $Y_k = \begin{bmatrix} g_1 - g_0 & \cdots & g_k - g_{k-1} \end{bmatrix} \in \mathbb{R}^{n\times k}$.
    Set $H_k = \mathrm{Id} + (S_k - Y_k)\left(Y_k^\top Y_k\right)^{-1} Y_k^\top$.
    Set $x_{k+1} = x_k - H_k g_k$.
end

Algorithm 4: AA-II (reformulated)

AA-I

This is the form of a quasi-Newton-like method so one could expect H_k to be an approximate inverse of $\nabla f(x_k)$. Indeed

Proposition (Approximate inverse Jacobian)

H_k minimises $\|H_k - \mathrm{Id}\|_F$ under the multisecant condition $H_k S_k = Y_k$.

From Broydens method we know that it is a good idea to approximate the Jacobian rather than its inverse.

Definition (Approximate Jacobian)

Let B_k be minimiser of $\|B_k - \mathrm{Id}\|_F$ under the condition $B_k Y_k = S_k$.

Analogously to AA-II we have

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

This yields the AA-I algorithm

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

Set $x_1 = f(x_0)$
for $k = 0, 1, \ldots$ **do**
  Set $g_k = g(x_k)$.
  Construct S_k from $x_0, \ldots, x_k$ and Y_k from $g_0, \ldots, g_k$.
  Set $B_k = \mathrm{Id} + (Y_k - S_k)\left(S_k^\top S_k\right)^{-1} S_k^\top$.
  Set $H_k = B_k^{-1}$.
  Set $x_{k+1} = x_k - H_k g_k$.
end

Algorithm 5: AA-I

But this algorithm has some problems

- computational efficiency: the approach is not matrix-free, we have to solve a linear system $\rightarrow$rank-1 update for B_k and later H_k
- well-definedness of H_k: B_k might not be well-defined or singular $\rightarrow$Powell-type regularisation, restarting iteration
- memory usage: though infinite memory is nice to have it is not very realistic $\rightarrow$restarting iteration
- convergence: the algorithm does not necessarily converge $\rightarrow$safeguarding steps

Luckily for us we can save some computations by using the rank-1 update formula

Proposition (Rank-1 update for B_k)

We have

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)\hat{s}_k^\top}{\hat{s}_k^\top s_k}$$

where $y_k = g_{k+1} - g_k$, $B_0 = \mathrm{Id}$ *and*

$$\hat{s}_k = s_k - \sum_{j=0}^{k-1} \frac{\hat{s}_k^\top s_k}{\|\hat{s}_k\|^2} \hat{s}_k$$

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

Taking everything together we obtain

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

Set $B_0 = \mathrm{Id}$ 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)

Powell-type regularisation

Note that B_k may be singular. To fix this we use powell-type regularisation.

s **Input :** $x^0 \in \mathbb{R}^n$, $f \colon \mathbb{R}^n \to \mathbb{R}^n$ and $\bar{\theta} \in (0,1)$.

Set $B_0 = \mathrm{Id}$ and $x_1 = f(x_0)$.
for $k = 0, 1, \ldots$ **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

One can obtain

Lemma (Powell-type regularisation)

If B_k is well-defined in algorithm 7 we have that B_k is invertible and

$$|\det B_k| \geq \theta^k .$$

Proof.

See [1, Lemma 2]. □

Restarting iteration

Note that

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

is ill-defined iff $\|\hat{s}_k\|^2 = \hat{s}_k^\top s_k = 0$, i.e. $\hat{s}_k = 0$. This occurs in algorithm 7 for $k > n$ as then $\hat{s}_k = 0$ by linear dependence. If we restart the algorithm with x_k as the new starting point if $k = m+1$ for some $m \in \mathbb{N}$ or $\|\hat{s}_{k-1}\| < \tau \|s_{k-1}\|$ for some $\tau \in (0,1)$ then

$$g_k \neq 0 \implies s_k = -B_k g_k \neq 0 \implies \hat{s}_k \neq 0\,.$$

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

Set $B_0 = \mathrm{Id}$, $x_1 = f(x_0)$ 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 $B_{k-1} = \mathrm{Id}$.

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

Lemma (Restarting iteration)

In algorithm 9 we have that B_k is well-defined and there exists a constant $c_1 = c_1(m, \bar{\theta}, \tau) > 0$ such that

$$\|B_k\| \leq c_1 \,.$$

Proof.

See [1, Lemma 3]. □

From the Sherman-Morrison formula one can obtain

Proposition (Rank-1 update for H_k)

We have

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)\hat{s}_k^\top H_k}{\hat{s}_k^\top H_k y_k}$$

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

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

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 $H_{k-1} = \mathrm{Id}$.

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

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

end

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

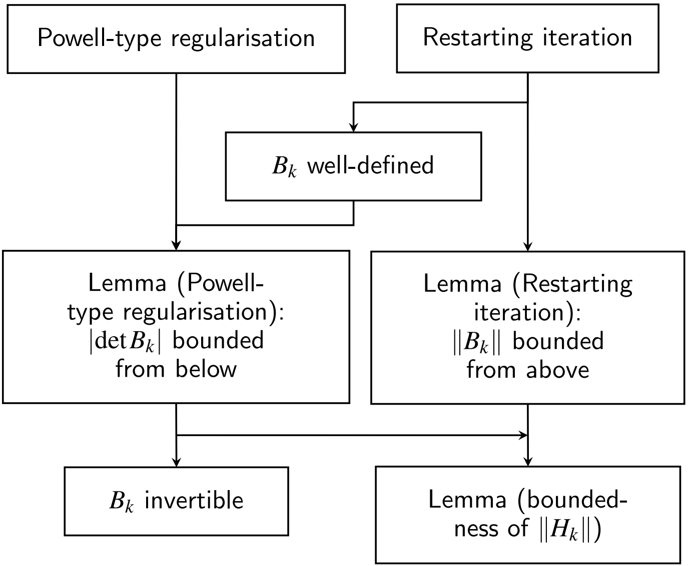
Lemma (bound on $\|H_k\|_2$)

In algorithm 9 there exists $c_2 = c_2(m, n, \bar{\theta}, \tau) > 0$ such that

$$\|H_k\|_2 \leq c_2 \,.$$

Proof.

This follows from Lemma (Restarting iteration) and Lemma (Powell-type regularisation). $\square$


Powell-type regularisation
Restarting iteration
B_k well-defined
Lemma (Powell-type regularisation): $|\det B_k|$ bounded from below
Lemma (Restarting iteration): $\|B_k\|$ bounded from above
B_k invertible
Lemma (boundedness of $\|H_k\|$)

Safeguarding steps

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

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

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

Input : $x^0 \in \mathbb{R}^n$, $f\colon \mathbb{R}^n \to \mathbb{R}^n$, $m \in \mathbb{N}$, $\bar{\theta}, \tau, \alpha \in (0,1)$ and safe-guarding constants $D, \varepsilon > 0$

Set $H_0 = \mathrm{Id}$, $x_1 = \tilde{x}_1 = f(x_0)$, $m_0 = n_{AA} = 0$ and $\bar{U} = \|g_0\|_2$.
for $k = 0, 1, \ldots$ **do**
  Set $g_k = g(x_k)$, $m_k = m_{k-1} + 1$, $s_{k-1} = \tilde{x}_k - x_{k-1}$ and $y_{k-1} = g(\tilde{x}_k) - g_{k-1}$.
  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 $H_{k-1} = \mathrm{Id}$.
  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 $H_k = H_{k-1} + \frac{(s_{k-1} - H_{k-1}\tilde{y}_{k-1})}{\hat{s}_{k-1}^\top H_{k-1} \tilde{y}_{k-1}}$ 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

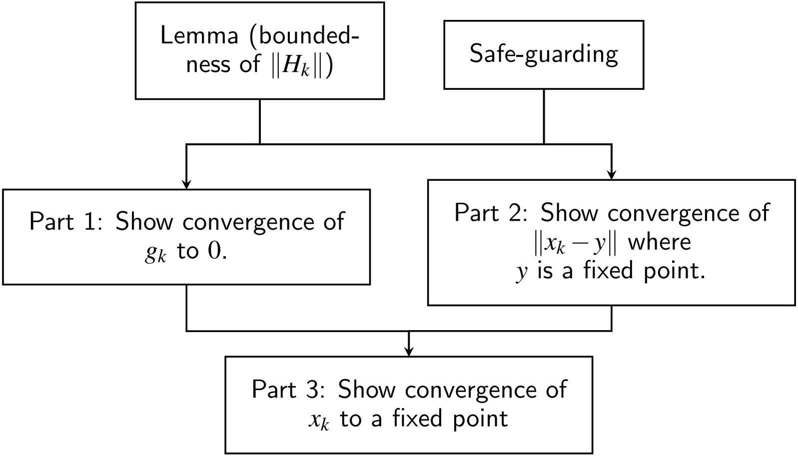
Algorithm 10: AA-I with Powell-type regularisation, Restarting and Safeguarding

Convergence result

Theorem (Convergence)

Let x_k be generated by algorithm 10 then $x_k \xrightarrow{k\to\infty} x$ and $f(x) = x$ is a fixed point.

Proof, strategy.



Proof, part 1.

The proof follows [1, Theorem 6]. We partition $\mathbb{N} = K_{AA} \sqcup K_{KM}$ where $K_{AA} = \{k_0, k_1, \ldots\}$ denote the indices k where the algorithm chose an AA-step (a) and $K_{KM} = \{l_0, l_1, \ldots\}$ 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}.

Proof, part 1 (cont.).

Let y be a fixed point. We distinguish

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

$$\begin{aligned} \|x_{k+1} - y\| &\leq \|x_k - y\| + \|H_k g_k\| \\ &\leq \|x_k - y\| + c_2 \|g_k\| \\ &\leq \|x_k - y\| + c_3 (k+1)^{-(1+\varepsilon)} \end{aligned} \tag{1}$$

case (b) $k \in K_{KM}$ then as f is nonexpansive (motivate this)

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - \alpha(1-\alpha)\|g_k\|^2 \tag{2}$$

Hence in any case

$$\|x_k - y\| \leq \|x_0 - y\| + c_3 \sum_k (k+1)^{-(1+\varepsilon)} = c_4 < \infty.$$

Proof, part 1 (cont.).

It then follows that

$$\begin{aligned} a_{k+1} = \|x_{k+1} - y\|^2 &\overset{(1),(2)}{\leq} \left(\|x_k - y\| + c_3(k+1)^{-(1+\varepsilon)} \right)^2 \\ &\leq \underbrace{\|x_k - y\|^2}_{=a_k} + \underbrace{c_3^2(k+1)^{-2(1+\varepsilon)} + 2c_3 \underbrace{\|x_k - y\|}_{\leq c_4}(k+1)^{-(1+\varepsilon)}}_{=b_k} \qquad (3) \\ &= a_k + b_k \end{aligned}$$

and hence

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

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 $\lim_i \|g_{k_i}\| = 0$. Thus indeed $\lim_k \|g_k\| = 0$.

Proof, part 2.

Let now n_j and $N_j \geq n_j$ be such that

$$a_{n_j} \xrightarrow{j\to\infty} \liminf_k a_k = \underline{a}$$

$$a_{N_j} \xrightarrow{j\to\infty} \limsup_k a_k = \overline{a}$$

Then it follows that

$$\overline{a} - \underline{a} \xleftarrow{n_j\to\infty} \overline{a} - a_{n_j} \xleftarrow{N_j\to\infty} a_{N_j} - a_{n_j} = \sum_{k=n_j}^{N_j-1} a_{k+1} - a_k \overset{(3)}{\leq} \sum_{k=n_j}^{\infty} b_k \xrightarrow{n_j\to\infty} 0$$

so

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

and thus $a_k = \|x_k - y\|$ converges to some a.

Proof, part 3.

Let k_j and l_j be convergent subsequences of x_k convergent against y_1 and y_2 respectively. Since by continuity of g

$$\|g(y_1)\| = \lim_j \|g(x_{k_j})\| \overset{\text{part 1}}{=} 0$$

we have that y_1 is a fixed point and y_2 too. Now by part 2

$$\|y_1\| \xleftarrow{j\to\infty} \|x_{k_j}\|^2 = \|x_{k_j} - y\|^2 + \|y\|^2 + 2y^\top x_{k_j} \xrightarrow{j\to\infty} a + \|y\|^2 + 2y^\top y_1$$

and analogously for y_2. Thus

$$\|y_i\| = a + \|y\|^2 + 2y^\top y_i$$

which implies

$$2y^\top (y_1 - y_2) = \|y_1\|^2 - \|y_2\|^2$$

Proof, part 3 (cont.).

It then follows from $y \in \{y_i\}_i$ that

$$y_1^\top (y_1 - y_2) = y_2^\top (y_1 - y_2)$$

and further

$$(y_1 - y_2)^\top (y_1 - y_2) = 0$$

and thus $y_1 = y_2$. We have shown that two convergent subsequences have the same limit and hence x_k is convergent and the limit must be a fixed point of f.

Elastic net regression

The aim is to find a fixed point of

$$f\colon \mathbb{R}^{1000} \to \mathbb{R}^{1000}, \quad x \mapsto S_{\alpha\mu/2}\Big(x - \alpha\Big(A^\top(Ax-b) + \frac{\mu}{2}x\Big)\Big)$$

with shrinkage operator

$$S_\kappa(x) = \big(\operatorname{sgn}(x_i)(|x_i| - \kappa)_+\big)_i$$

and $A \in \mathbb{R}^{500\times 1000}$, $b \in \mathbb{R}^{500}$ and some $\alpha, \mu \in \mathbb{R}$.

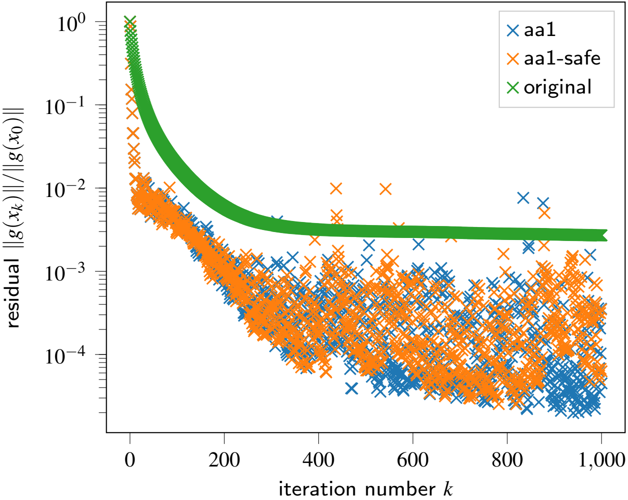


Figure: Residual norms for the elastic net regression problem.

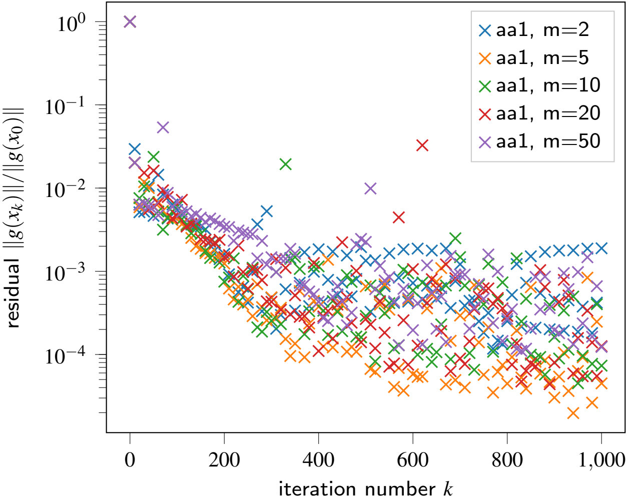


Figure: Residual norms for the elastic net regression problem.

Markov decision process

Our aim is to find a fixed point of the Bellman operator

$$f\colon \mathbb{R}^{1000} \to \mathbb{R}^{1000}, \quad x \mapsto \left(\max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s') x_{s'} \right) \right)_s$$

with some $R \in \mathbb{R}^{300\times 200}$, $P \in \mathbb{R}^{300\times 200\times 300}$, $\gamma \in \mathbb{R}$.

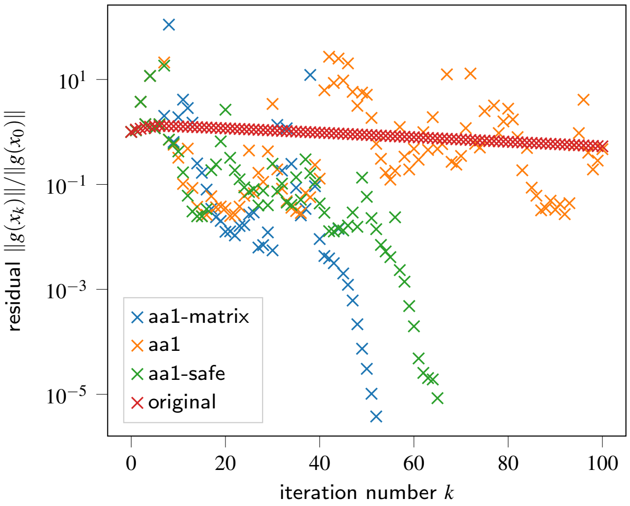


Figure: Residual norms for the Markov decision process problem.

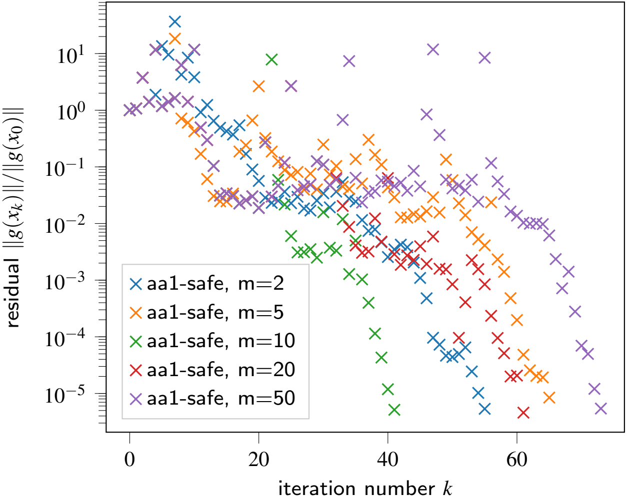


Figure: Residual norms for the Markov decision process problem.

Summary

- aim is to find a fixed point of f where
  - the dimension is large
  - f is expensive to evaluate, noisy and the gradient is a mystery
- 3 modifications to the AA-I algorithm yield well-definedness and convergence for non-expansive problems
  - Powell-type regularisation
  - Restarting iteration
  - Safeguarding steps
-

Sources I

[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.

[2] I. Guyon. (2004), Madelon data set, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Madelon.

[3] 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.

Sources II

[4] numerics-seminar-VT23, *Github repository to the project.* Online, 2023. [Online]. Available: https://github.com/TheoKoppenhoefer/numerics-seminar-VT23.

Thank you for your attention.