Two approaches for absolute value equations by using smoothing functions

Jein-Shan Chen

Department of Mathematics
National Taiwan Normal University

August 26, 2020

The absolute value equation (AVE)

The absolute value equation (AVE) is in the form of

$$Ax + B|x| = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, $B \neq 0$, and $b \in \mathbb{R}^n$. Here $|x|$ means the componentwise absolute value of vector $x \in \mathbb{R}^n$, i.e., $|x| = (|x_1|, |x_2|, \cdots, |x_n|)$.

When $B = -I$, where I is the identity matrix, the AVE (1) reduces to the special form:

$$Ax - |x| = b. \tag{2}$$

The AVE (1) was first introduced by Rohn in 2004. Originally, an alternative theorem for $Ax + B|x| = b$ was proposed as below.

Theorem (Rohn, LMA, 2004)

Let $A, D \in \mathbb{R}^{n\times n}$, $D \geq 0$. Then, the following alternative holds:

(i) *for each $B \in \mathbb{R}^{n\times n}$ with $|B| \leq D$ and for each $b \in \mathbb{R}^n$ the equation $Ax + B|x| = b$ has a unique solution.*

(ii) *there exist $\lambda \in [0,1]$ and a ± 1-vector y such that the equation $Ax + \lambda\,\mathrm{diag}(y)\,D|x| = b$ has a nontrivial solution.*

Solving the AVE is NP-hard (Mangasarian, 2006).
The problem of checking whether the AVE has unique or multiple solutions is NP-complete (Prokopyev, 2009).

History of the AVEs - (2)

- Indeed, the name "absolute value equation" was termed by Mangasarian in 2006.
- The AVEs have been investigated by many researchers, for example, Caccetta-Qu-Zhou, Hu-Huang, Jiang-Zhang, Ketabchi-Moosaei, Mangasarian, Mangasarian-Meyer, Moosaei-Ketabchi-Noor-Iqbal-Hooshyarbakhsh, Prokopyev, Rohn, and Zhang-Wei.
- Mangasarian-Meyer (2006) and Prokopyev (2009) prove that the LCP (linear complementarity problem) can be reformulated as an AVE and hence the AVE is NP-hard.

Why study the AVE ?

- Many mathematical programming problems like linear programs, quadratic programs, bimatrix games, mixed integer programs and other problems can all be reduced to an LCP which in turn is equivalent to the AVE.
- The AVE formulation is simpler to state than an LCP.
- The AVE is an equation! It attracts more audience for solving equations.

Equivalence and solvability of the AVE

- Mangasarian and Meyer (2009) show that the AVE (1) is equivalent to the bilinear program, and the standard LCP provided 1 is not an eigenvalue of A.
- Prokopyev (2009) further improves the above equivalence which indicates that the AVE (1) can be recast as an LCP without any assumption on A and B, and also provides a relationship with mixed integer programming.
- In general, if solvable, the AVE (1) could have either unique solution or multiple (e.g., exponentially many) solutions.

Numerical methods for solving AVEs

- A parametric successive linearization algorithm for the AVE (1) that terminates at a point satisfying necessary optimality conditions is studied by Mangasarian (2007).
- The generalized Newton algorithm for the AVE (2) is also investigated by Mangasarian (2009), in which it is proved that this algorithm converges linearly from any starting point to the unique solution of the AVE (2) under the condition that $\|A^{-1}\| < \frac{1}{4}$.
- The generalized Newton algorithm with semismooth and smoothing Newton steps combined into the algorithm is considered by Zhang-Wei (2009).
- The smoothing-type algorithms for solving the AVEs (1)-(2) are studied by Caccetta-Qu-Zhou (2011), Hu-Huang (2010), Jiang-Zhang (2013).

- A branch and bound method for the absolute value programs (AVP), which contains absolute values of variables in its objective function and constraints and is an extension of the AVE, is studied by Yamanaka-Fukushima (2014).
- The absolute value equation associated with second order cone (SOCAVE) is studied by Hu-Huang-Zhang (2011), Miao-Yang-Saheya-Chen (2017), and Miao-Hsu-Nguyen-Chen (2020).
- The absolute value equation associated with circular cone (CCAVE) is studied by Miao-Yang-Hu (2015).
- For SOCAVE and CCAVE, $|x|$ is defined in a different way from AVE. More specifically, $|x| := (x \circ x)^{1/2}$ by certain Jordan product "$\circ$".

Infeasible AVEs

- If b has at least one positive element and $\|A\|_\infty < \frac{\gamma}{2}$ where $\gamma = \max_{b_i > 0} b_i / \max |b_i|$, then the the AVE (1) has no solution (Prokopyev, 2009).
- In many real models, one often encounter problems which present themselves as systems of infeasible AVEs.
- The reasons for the infeasibility of an AVE include errors in the data, the complexity of the model, optimistic objectives, and lack of communication between different decision makers.
- The correction of infeasible AVEs is studied by Ketabchi-Moosaei-Fallahi (2013) and Moosaei-Ketabchi-Pardalos (2016).

Two approaches for sloving the AVEs

- Many approaches have been proposed during the past decade and most of them focus on reformulating it as complementarity problem and then solve it accordingly.
- Another approach is to recast the AVE as a system of "nonsmooth" equations and then tackle with the nonsmooth equations.
- Our 1st approach is to rewrite it as a system of "smooth" equations and propose four new smoothing functions along with a smoothing-type algorithm to solve the system of equations.
- Our 2nd approach is neural network approach.

Main idea of 1st approach

More specifically, we define $H_i : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ as

$$H_i(\mu, x) = \begin{bmatrix} \mu \\ Ax + B\Phi_i(\mu, x) - b \end{bmatrix} \quad \text{for } \mu \in \mathbb{R} \text{ and } x \in \mathbb{R}^n \quad (3)$$

where $\Phi_i : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is given by

$$\Phi_i(\mu, x) := \begin{bmatrix} \phi_i(\mu, x_1) \\ \phi_i(\mu, x_2) \\ \vdots \\ \phi_i(\mu, x_n) \end{bmatrix} \quad \text{for } \mu \in \mathbb{R} \text{ and } x \in \mathbb{R}^n \quad (4)$$

- Main idea: $H_i(\mu, x) = 0$ if and only if x solves the AVE (1).
- Will introduce four smoothing functions $\phi_i \approx |t|$.

Four smoothing functions

$$\phi_1(\mu,t) = \mu\left[\ln(1+e^{-\frac{t}{\mu}})+\ln(1+e^{\frac{t}{\mu}})\right] \quad (5)$$

$$\phi_2(\mu,t) = \begin{cases} t & \text{if} \quad t \geq \frac{\mu}{2}, \\ \dfrac{t^2}{\mu}+\dfrac{\mu}{4} & \text{if} \quad -\frac{\mu}{2} < t < \frac{\mu}{2}, \\ -t & \text{if} \quad t \leq -\frac{\mu}{2}. \end{cases} \quad (6)$$

$$\phi_3(\mu,t) = \sqrt{4\mu^2+t^2} \quad (7)$$

$$\phi_4(\mu,t) = \begin{cases} \dfrac{t^2}{2\mu} & \text{if} \quad |t| \leq \mu, \\ |t|-\frac{\mu}{2} & \text{if} \quad |t| > \mu. \end{cases} \quad (8)$$

The function ϕ_4 is constructed in a different way.

Properties of ϕ_i functions

Theorem (Saheya-Yu-Chen, JAMC, 2017)

Let $\phi_i : \mathbb{R}^2 \to \mathbb{R}$ for $i = 1, 2, 3, 4$ be defined as in (5), (6), (7) and (8), respectively. Then, we have

(a) ϕ_i *is continuously differentiable at* $(\mu, t) \in \mathbb{R}_{++} \times \mathbb{R}$;

(b) $\lim_{\mu \downarrow 0} \phi_i(\mu, t) = |t|$ *for any* $t \in \mathbb{R}$.

Graph of ϕ_1 function

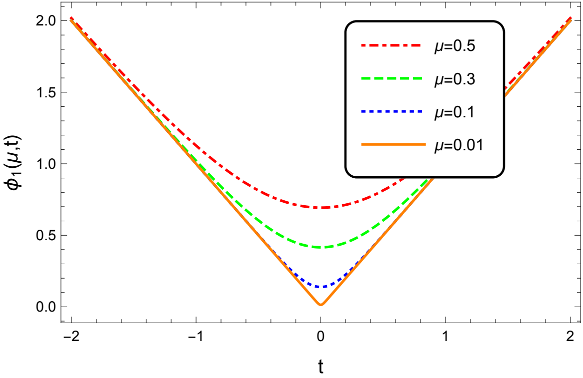


Figure: Graphs of $\phi_1(\mu, t)$ with $\mu = 0.01, 0.1, 0.3, 0.5$.

Graph of ϕ_2 function

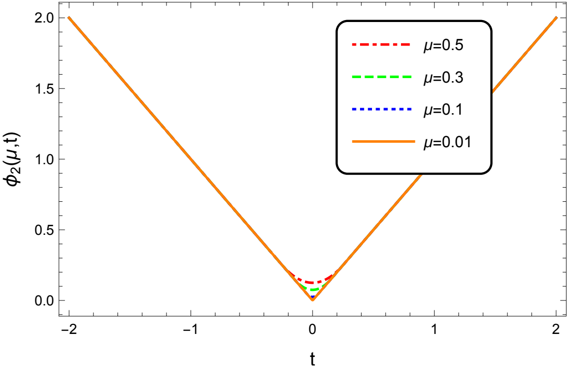


Figure: Graphs of $\phi_2(\mu, t)$ with $\mu = 0.01, 0.1, 0.3, 0.5$.

Graph of ϕ_3 function

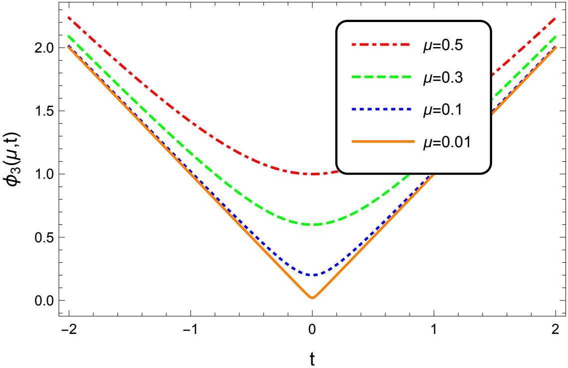


Figure: Graphs of $\phi_3(\mu, t)$ with $\mu = 0.01, 0.1, 0.3, 0.5$.

Graph of ϕ_4 function

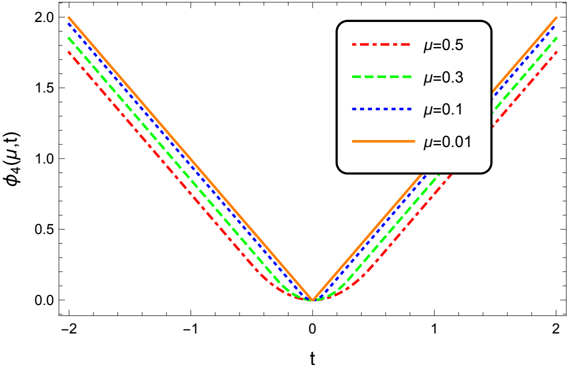


Figure: Graphs of $\phi_4(\mu, t)$ with $\mu = 0.01, 0.1, 0.3, 0.5$.

Differentiability of H_i - (1)

Theorem (Saheya-Yu-Chen, JAMC, 2017)

Let $\Phi_i(\mu, x)$ for $i = 1, 2, 3, 4$ be defined as in (4). Then, we have

(a) *$H_i(\mu, x) = 0$ if and only if x solves the AVE (1);*

(b) *H_i is continuously differentiable on $\mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$ with the Jacobian matrix given by*

$$\nabla H_i(\mu, x) := \begin{bmatrix} 1 & 0 \\ B\,\nabla_1 \Phi_i(\mu, x) & A + B\,\nabla_2 \Phi_i(\mu, x) \end{bmatrix} \tag{9}$$

where

$$\nabla_1 \Phi_i(\mu, x) := \begin{bmatrix} \frac{\partial \phi_i(\mu, x_1)}{\partial \mu} \\ \frac{\partial \phi_i(\mu, x_2)}{\partial \mu} \\ \vdots \\ \frac{\partial \phi_i(\mu, x_n)}{\partial \mu} \end{bmatrix},$$

$$\nabla_2 \Phi_i(\mu, x) := \begin{bmatrix} \frac{\partial \phi_i(\mu, x_1)}{\partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial \phi_i(\mu, x_2)}{\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\partial \phi_i(\mu, x_n)}{\partial x_n} \end{bmatrix}.$$

Geometric comparison of ϕ_i functions

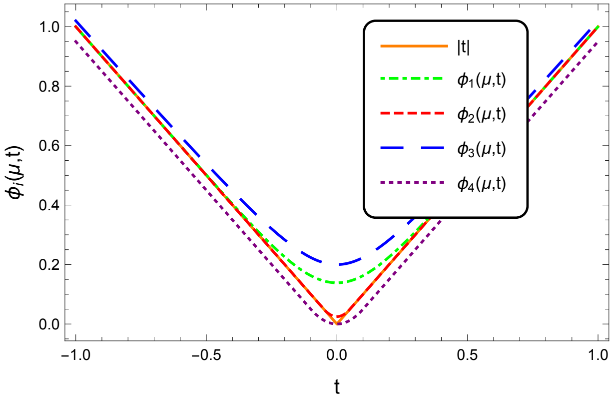


Figure: Graphs of $|t|$ and all four $\phi_i(\mu, t)$ with $\mu = 0.1$.

Question

From the figure, we see that ϕ_2 is the one which best approximates the function $|t|$ under the sense that it is closest to $|t|$ among all ϕ_i for $i = 1, 2, 3, 4$. How to verify it algebraically?

In fact, for fixed $\mu > 0$, there has the local behavior:

$$\phi_3(\mu, t) > \phi_1(\mu, t) > \phi_2(\mu, t) > |t| > \phi_4(\mu, t).$$

Algebraic comparison of ϕ_i functions - (1)

How to measure the distance of two functions?

We employ $\|\cdot\|_\infty$ to measure the distance of two functions. In other words, for given two real-valued functions f and g, the distance between them is defined as

$$\|f-g\|_\infty = \max_{t\in\mathbb{R}}\{f(t)-g(t)\}.$$

Algebraic comparison of ϕ_i functions - (2)

First, for any fixed $\mu > 0$, we verify that

$$\lim_{|t|\to\infty} \left|\phi_i(\mu, t) - |t|\right| = 0, \text{ for } i = 1, 2, 3.$$

This implies that

$$\max_{t\in\mathbb{R}} \left|\phi_i(\mu, t) - |t|\right| = |\phi_i(\mu, 0)|, \text{ for } i = 1, 2, 3.$$

Since, $\phi_1(\mu, 0) = (2\ln 2)\mu \approx 1.4\mu$, $\phi_2(\mu, 0) = \frac{\mu}{4}$, and $\phi_3(\mu, 0) = 2\mu$, we have

$$\begin{aligned} \left\|\phi_1(\mu, t) - |t|\right\|_\infty &= (2\ln 2)\mu \approx 1.4\mu \\ \left\|\phi_2(\mu, t) - |t|\right\|_\infty &= \frac{\mu}{4} \\ \left\|\phi_3(\mu, t) - |t|\right\|_\infty &= 2\mu \end{aligned}$$

On the other hand, we see that

$$\lim_{t\to\infty} \left|\phi_4(\mu, t) - |t|\right| = \frac{\mu}{2} \quad \text{and} \quad \phi_4(\mu, 0) = 0,$$

which says

$$\max_{t\in\mathbb{R}} \left|\phi_4(\mu, t) - |t|\right| = \frac{\mu}{2}.$$

Hence, we obtain

$$\left\|\phi_4(\mu, t) - |t|\right\|_\infty = \frac{\mu}{2}.$$

Algebraic comparison of ϕ_i functions - (4)

From all the above, we conclude that

$$\begin{aligned} & \left\| \phi_3(\mu, t) - |t| \right\|_\infty > \left\| \phi_1(\mu, t) - |t| \right\|_\infty \\ > & \left\| \phi_4(\mu, t) - |t| \right\|_\infty > \left\| \phi_2(\mu, t) - |t| \right\|_\infty . \end{aligned}$$

This shows that ϕ_2 is the function among ϕ_i, $i = 1, 2, 3, 4$ which best approximates the function $|t|$, and

$$\phi_3(\mu, t) > \phi_1(\mu, t) > \phi_2(\mu, t) > |t| > \phi_4(\mu, t).$$

Question

A natural question arises here, does the smoothing algorithm based on ϕ_2 perform best among all $\phi_1, \phi_2, \phi_3, \phi_4$?

Solvability of the AVE

Assumption (A1)

The minimal singular value of the matrix A is strictly greater than the maximal singular value of the matrix B.

The Assumption (A1) is used to guarantee that $\nabla H_i(\mu, x)$ is invertible at any $(\mu, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$.

Theorem (Jiang-Zhang, JIMO, 2013)

The AVE (1) is uniquely solvable for any $b \in \mathbb{R}^n$ if Assumption (A1) is satisfied.

A Smoothing-Type Algorithm - (1)

A Smoothing-Type Algorithm

Step 0. Choose $\delta, \sigma \in (0,1), \mu_0 > 0, x^0 \in \mathbb{R}^n$. Set $z^0 := (\mu, x^0)$. Denote $e^0 := (1,0) \in \mathbb{R} \times \mathbb{R}^n$. Choose $\beta > 1$ such that $\left(\min\left\{1, \|H_i(z^0)\|\right\}\right)^2 \leq \beta\mu_0$. Set $k := 0$.

Step 1. If $\|H_i(z^k)\| = 0$, stop.

Step 2. Set $\tau_k := \min\left\{1, \|H_i(z^k)\|\right\}$, and compute $\triangle z^k := (\triangle\mu_k, \triangle x^k) \in \mathbb{R} \times \mathbb{R}^n$ by using

$$\nabla H_i(z^k)\triangle z^k = -H_i(z^k) + (1/\beta)\tau_k^2 e^0,$$

where $\nabla H_i(\cdot)$ is defined as in (9).

A Smoothing-Type Algorithm - (2)

A Smoothing-Type Algorithm

Step 3. Let α_k be the maximum of the values $1, \delta, \delta^2, \cdots$ such that

$$\left\| H_i(z^k + \alpha_k \triangle z^k) \right\| \leq [1 - \sigma(1 - 1/\beta)\alpha_k] \left\| H_i(z^k) \right\|$$

Step 4. Set $z^{k+1} := z^k + \alpha_k \triangle z^k$ and $k := k + 1$. Back to Step 1.

The algorithm is well-defined

Theorem (Saheya-Yu-Chen, JAMC, 2017)

(a) *Suppose that Assumption (A1) holds. Then, the above Algorithm is well-defined.*

(b) *Let the sequence* $\{z^k\}$ *be generated by the Algorithm. Then,*

   (i) *both* $\{\|H_i(z^k)\|\}$ *and* $\{\tau_k\}$ *are monotonically decreasing;*

   (ii) $\tau_k^2 \leq \beta\mu_k$ *holds for all* k*;*

   (iii) *the sequence* $\{\mu_k\}$ *is monotonically decreasing, and* $\mu_k > 0$ *for all* k*.*

Convergence of the algorithm

Theorem (Saheya-Yu-Chen, JAMC, 2017)

Let H_i and ∇H_i be given as in (3) and (9), respectively. Suppose that Assumption (A1) holds. Then, $\nabla H_i(\mu, x)$ is invertible at any $(\mu, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$.

Theorem (Saheya-Yu-Chen, JAMC, 2017)

Suppose that Assumption (A1) holds and that the sequence $\{z^k\}$ is generated by the Algorithm. Then,

(a) *$\{z^k\}$ is bounded;*

(b) *any accumulation point of $\{z^k\}$ is a solution of the AVE (1);*

(c) *the whole sequence $\{z^k\}$ convergence to z^* with $\|z^{k+1} - z^k\| = o\left(\|z^k - z^*\|\right)$ and $\mu_{k+1} = \mu_k^2$.*

Numerical implementation

- All numerical experiments are carried out in Mathematica 10.0 running on a PC with Intel i5 of 3.00GHz CPU processor, 4.00GB Memory and 32-bit Windows 7 operating system.
- In our numerical experiments, the stoping criteria for the Algorithm is $\|H_i(z^k)\| \leq 1.0e-6$.
- We also stop programs when the total iteration is more than 100.
- Throughout the computational experiments, the following parameters are used:

$$\delta = 0.5, \;\; \sigma = 0.0001, \;\; \mu_0 = 0.1, \;\; \beta = \max\left\{1, 1.01 * \tau_0^2/\mu\right\}.$$

Notations

Dim	the size of problem,
N_ϕ_i	the average number of iterations,
T_ϕ_i	the average value of the CPU time in seconds,
Ar_ϕ_i	the average value of $\|H(z^k)\|$ when Algorithm stops.

Test Problem 1

Consider the ODE (Haghani, [Example 4.2] in JOTA, 2015):

$$\frac{d^2x}{dt^2} - |x| = (1 - t^2), \ x(0) = -1, \ x(1) = 0, \ t \in [0, 1]. \qquad (10)$$

After descretization, the above ODE can be recast an AVE (2):

$$Ax - |x| = b,$$

where the matrix A is given by

$$a_{ij} = \begin{cases} -242, & i = j, \\ 121, & |i - j| = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Numerical results of Test Problem 1

Table: The numerical results of ordinary differential equation (10)

Dim	N_ϕ_1	T_ϕ_1	Ar_ϕ_1	N_ϕ_2	T_ϕ_2	Ar_ϕ_2	N_ϕ_3	T_ϕ_3	Ar_
2	5.1	0.0967	3.30E-07	3.9	0.0015	6.92E-08	5.1	0.0016	5.93E
5	5.9	0.3697	2.23E-07	4.1	0.0031	7.47E-08	5.6	0.0062	2.21E
10	6.4	0.4851	2.98E-07	4.3	0.0094	2.10E-07	5.9	0.0031	1.05E
20	5.2	0.4290	2.41E-07	4.9	0.0078	1.10E-08	6.3	0.0078	2.13E
40	8.8	4.4117	4.66E-07	6.1	0.5210	5.28E-08	7.3	0.0172	6.59E
60	9.1	2.4289	2.31E-07	6.8	0.0281	4.49E-08	9	0.0312	1.20E
80	9.8	2.0514	3.61E-07	7.4	0.0374	3.21E-10	9.3	0.0452	3.21E
100	9.8	8.2306	4.44E-07	7.8	0.0577	8.78E-08	10	0.0671	2.26E

Performance profile of iteration for Problem 1

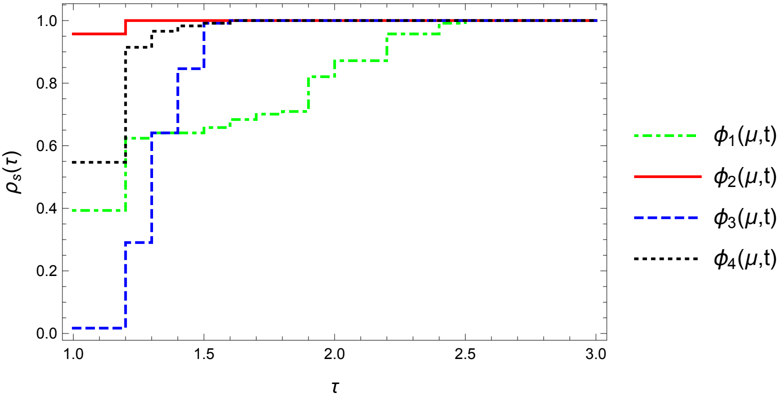


Figure: Performance profile of iteration numbers of for the ODE (10).

Performance profile of computing time for Problem 1

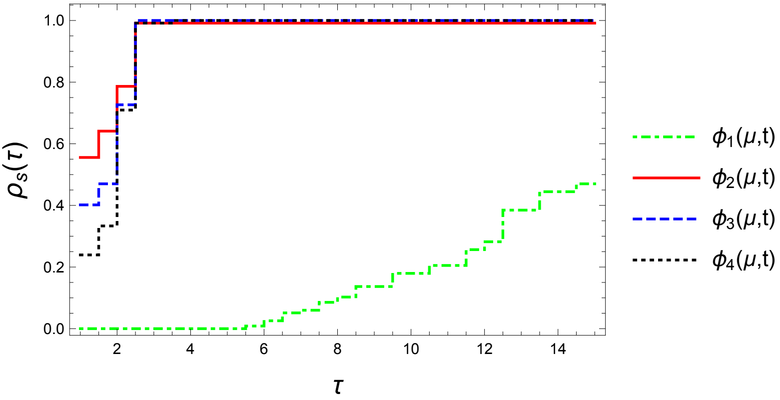


Figure: Performance profile of computing time for the ODE (10).

Performance summary for Problem 1

In view of "iteration numbers", there has

$$\phi_2(\mu, t) > \phi_4(\mu, t) > \phi_3(\mu, t) > \phi_1(\mu, t)$$

where ">" means "better performance".

In view of "computing time", there has

$$\phi_2(\mu, t) > \phi_3(\mu, t) > \phi_4(\mu, t) > \phi_1(\mu, t)$$

where ">" means "better performance".

Test Problem 2

Consider the general AVE (1): $Ax + B|x| = b$ where A (or B) is equal to a normal distribution random matrix minus another one so that we can randomly generate the testing problems.

In order to ensure that Assumption (A1) holds, we further modify the matrix A in light of the below conditions.

- If $\min\{w_{ii} : i = 1, \dots, n\} = 0$ with $\{u, w, v\} = \mathrm{SingularValueDecomposition}[A]$, then we set $A = u(w + 0.01 * \mathrm{IdentityMatrix}[n])v$.
- Set $A = \frac{\lambda_{max}(B^{\mathrm{T}}B)+0.01}{\lambda_{min}(A^{\mathrm{T}}A)} A$.

Moreover, we set $p =$2RandomVariate[NormalDistribution[ ],{n, 1}] and $b = Ap + B|p|$ so that the testing problems are solvable.

Numerical results of Test Problem 2

Table: The numerical results of experiments

Dim	N_ϕ_1	T_ϕ_1	Ar_ϕ_1	N_ϕ_2	T_ϕ_2	Ar_ϕ_2	N_ϕ_3	T_ϕ_3	Ar_ϕ_3
2	6.2	0.4596	5.00E-7	3.6	0.0031	8.56E-8	7.1	0.0016	1.79E-7
5	7.4	0.2246	6.05E-7	4.1	0.0031	8.39E-8	9.6	0.0094	4.73E-7
10	10.2	1.0733	2.23E-7	4.3	0.0062	8.26E-8	17.2	0.0187	4.79E-7
20	19.8	3.7830	5.00E-7	4.8	0.0062	9.95E-8	26.3	0.0499	1.86E-7
30	28.7	5.0575	4.46E-7	5.6	0.0140	1.00E-7	43.2	0.1295	5.22E-8
40	38.6	3.0935	6.52E-7	7.1	0.0234	5.60E-8	54.1	0.2137	1.65E-7
50	42.7	1.9016	5.37E-7	5.3	0.0218	7.73E-8	61.5	0.3120	1.93E-8
60	52.1	2.5272	5.61E-7	6.6	0.0359	5.90E-8	78.7	0.4976	1.05E-8
70	60.2	3.7050	6.10E-7	9.9	0.0624	1.12E-7	94.4	0.7332	1.80E-7
80	58.0	4.1246	4.31E-7	8.9	0.0640	6.03E-8	98.5	0.8845	3.88E-8
90	78.2	11.170	6.28E-7	10.0	0.0905	2.23E-7	114.3	1.2745	1.46E-7
100	72.2	12.211	4.77E-7	7.5	0.0709	1.62E-7	110.8	1.6477	1.31E-7

Performance profile of iteration for Problem 2

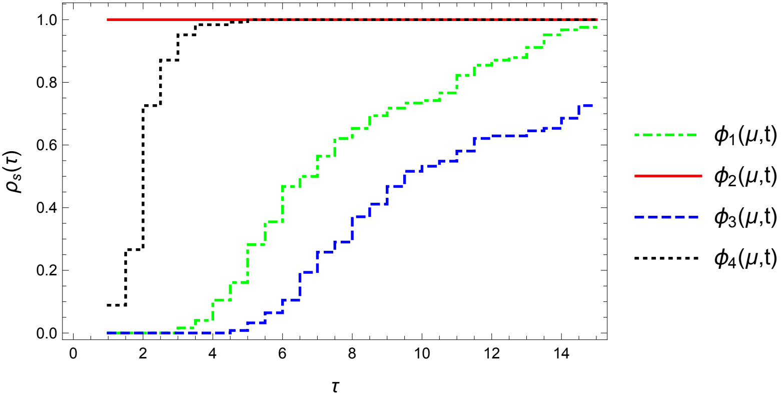


Figure: Performance profile of iteration numbers for general AVE.

Performance profile of computing time for Problem 2

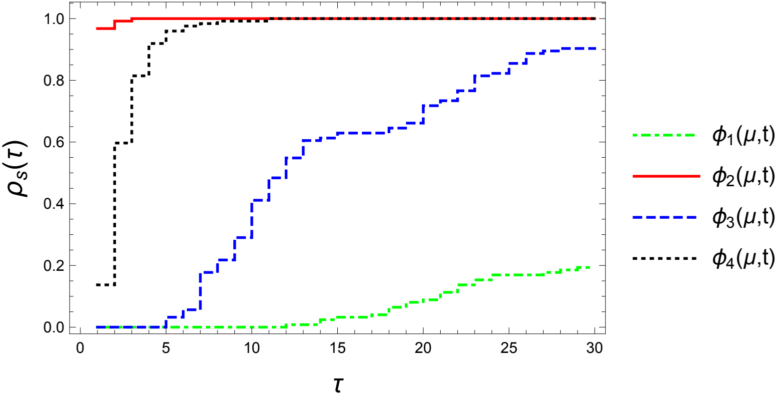


Figure: Performance profile of computing time for general AVE.

Performance summary for Problem 2

In view of "iteration numbers", there has

$$\phi_2(\mu, t) > \phi_4(\mu, t) > \phi_1(\mu, t) > \phi_3(\mu, t)$$

where "$>$" means "better performance".

In view of "computing time", there has

$$\phi_2(\mu, t) > \phi_4(\mu, t) > \phi_3(\mu, t) > \phi_1(\mu, t)$$

where "$>$" means "better performance".

Summary of numerical performance - (1)

For the "iteration" aspect, the order of numerical performance from good to bad is

$$\begin{cases} \phi_2(\mu,t) > \phi_4(\mu,t) > \phi_1(\mu,t) > \phi_3(\mu,t), & \text{for th AVE (1).} \\ \phi_2(\mu,t) > \phi_4(\mu,t) > \phi_3(\mu,t) > \phi_1(\mu,t), & \text{for th AVE (2).} \end{cases}$$

Recall that for fixed $\mu > 0$, there has

$$\phi_3(\mu,t) > \phi_1(\mu,t) > \phi_2(\mu,t) > |t| > \phi_4(\mu,t).$$

Summary of numerical performance - (2)

For the "time" aspect, the order of numerical performance from good to bad is

$$\begin{cases} \phi_2(\mu,t) > \phi_4(\mu,t) > \phi_3(\mu,t) > \phi_1(\mu,t), & \text{for th AVE (1).} \\ \phi_2(\mu,t) > \phi_3(\mu,t) > \phi_4(\mu,t) > \phi_1(\mu,t), & \text{for th AVE (2).} \end{cases}$$

Recall that for fixed $\mu > 0$, there has

$$\phi_3(\mu,t) > \phi_1(\mu,t) > \phi_2(\mu,t) > |t| > \phi_4(\mu,t).$$

Remarks

- The function $\phi_2(\mu, t)$ is the best choice of smoothing function to work with the proposed smoothing-type algorithm, meanwhile it also best approximate the function $|t|$.
- Piecewise smoothing functions seem better choices than other types of smoothing functions.
- It is interesting to check whether such phenomenon occurs in other types of algorithms.

The 2nd approach: neural network

We try another way, neural network approach, to solve AVE.

- We try to investigate all possible ways to construct smoothing functions for $|x|$.
- There are are 5 ways which are summarized as below.
- Then, we employ 8 smoothing functions in the neural network approach.

For any function $f : \mathrm{dom} f \to \mathbb{R}$, its convex conjugate $f^* : (\mathrm{dom} f)^* \to \mathbb{R}$ is defined in terms of the supremum by

$$f^*(y) := \sup_{x \in \mathrm{dom} f} \left\{ x^T y - f(x) \right\}.$$

In light of this, one can build up smooth approximation of f, denoted by f_μ, by adding strongly convex component to its dual $g := f^*$, namely,

$$f_\mu(x) = \sup_{z \in \mathrm{dom} g} \left\{ z^T x - g(x) - \mu d(z) \right\} = (g + \mu d)^*(x)$$

for some 1-strongly convex and continuous function $d(\cdot)$ (called proximity function).

Definition of 1-strongly convex function

The function $d(\cdot)$ is 1-strongly convex which means

$$d((1-t)x+ty) \leq (1-t)d(x)+td(y)-\frac{1}{2}t(1-t)\|x-y\|^2,$$

for all x, y and $t \in (0,1)$.

In general, a function $d(\cdot)$ is called μ-strongly convex with modulus $\mu > 0$, it means

$$d((1-t)x+ty) \leq (1-t)d(x)+td(y)-\frac{\mu}{2}t(1-t)\|x-y\|^2,$$

for all x, y and $t \in (0,1)$.

Note that $|x| = \sup_{|z| \le 1} zx$. If we take $d(z) := z^2/2$, then the constructed smoothing function via conjugation leads to

$$\phi_1(x, \mu) = \sup_{|z| \le 1} \left\{ zx - \frac{\mu}{2} z^2 \right\} = \begin{cases} \frac{x^2}{2\mu}, & \text{if } |x| \le \mu, \\ |x| - \frac{\mu}{2}, & \text{otherwise.} \end{cases}$$

which is the traditional Huber function.

The smoothing function $\phi_2(x, \mu)$

It is also possible to consider another expression:

$$|x| = \sup_{\substack{z_1+z_2=1 \\ z_1, z_2 \succeq 0}} (z_1 - z_2)x$$

Under this case, if we take $d(z) := z_1 \log z_1 + z_2 \log z_2 + \log 2$, the constructed smoothing function by conjugation becomes

$$\phi_2(x, \mu) = \mu \log \left(\cosh \left(\frac{x}{\mu} \right) \right)$$

where $\cosh(x) := \dfrac{e^x + e^{-x}}{2}$.

The smoothing function $\phi_3(x, \mu)$

Alternatively, choosing $d(y) := 1 - \sqrt{1 - y^2}$ gives another smoothing function:

$$\phi_3(x, \mu) = \sup_{-1 \le y \le 1} \left(xy + \mu\sqrt{1 - y^2} - \mu \right) = \sqrt{x^2 + \mu^2} - \mu.$$

Suppose that $\mathbb{E}$ is a finite vector space and $f : \mathbb{E} \to (-\infty, \infty]$ is a closed and proper convex function. The Moreau proximal approximation yields a family of approximations $\{f_\mu^{\rm px}\}_{\mu>0}$ as below:

$$f_\mu^{\rm px}(x) = \inf_{u \in \mathbb{E}} \left\{ f(u) + \frac{1}{2\mu} \|u - x\|^2 \right\}. \tag{11}$$

Remarks about Moreau proximal smoothing

Remarks

- It is known that the Moreau proximal approximation $f_\mu^{\mathrm{px}}(x)$ is convex continuous, finite-valued, and differentiable with gradient $\nabla f_\mu^{\mathrm{px}}$ which is Lipschitz continuous with constant $\frac{1}{\mu}$.
- When applying the Moreau proximal smoothing way to construct the smoothing function for the absolute value function $|x|$, it also yields the Huber smoothing function $\phi_1(x,\mu)$ by using the Moreau envelop.

3. Nesterov's smoothing

Consider the class of nonsmooth convex functions

$$q(x) = \max\{\langle u, Ax\rangle - \phi(u) \,|\, u \in Q\}, \quad x \in \mathbb{E},$$

where $\mathbb{E}$, V are finite dimensional vector spaces, $Q \subseteq V^*$ is compact and convex, ϕ is a continuous convex function on Q, and $A : \mathbb{E} \to V$ is a linear map.

The smooth approximation of q is described by the convex function

$$q_\mu(x) = \max\{\langle u, Ax\rangle - \phi(u) - \mu d(u) \,|\, u \in Q\}, \quad x \in \mathbb{E}, \qquad (12)$$

where $d(\cdot)$ is a prox-function for Q. It was proved that the convex function $q_\mu(x)$ is $C^{1,1}(\mathbb{E})$. More specifically, its gradient mapping is Lipschitz continuous with constant $L_\mu = \dfrac{\|A\|^2}{\sigma\mu}$ and the gradient is described by $\nabla q_\mu(x) = Au_\mu(x)$, where $u_\mu(x)$ is the unique minimizer of (12).

Example by Nesterov's smoothing

For the absolute value function $\phi(x) = |x|$ with $x \in \mathbb{R}^1$, Let $A = 1$, $b = 0$, $\mathbb{E} = \mathbb{R}^1$, $Q = \{u \in \mathbb{R}^1 \,|\, |u| \leq 1\}$ and taking $d(u) := \frac{1}{2}u^2$. Then, we have

$$\begin{aligned} \phi_\mu(x,\mu) &= \max_u \{\langle Ax - b, u\rangle - \mu d(u) \,|\, u \in Q\} \\ &= \max_u \left\{ xu - \frac{\mu}{2}u^2 \right\} \\ &= \begin{cases} \frac{x^2}{2\mu}, & \text{if } |x| \leq \mu, \\ |x| - \frac{\mu}{2}, & \text{otherwise.} \end{cases} \end{aligned}$$

As we see, it also yields the Huber smoothing function $\phi_1(x,\mu)$.

4. The infimal-convolution smoothing technique - (1)

Suppose that $\mathbb{E}$ is a finite vector space and $f, g : \mathbb{E} \to (-\infty, \infty]$. The infimal convolution of f and g, $f \,\square\, g : \mathbb{E} \to [-\infty, +\infty]$ is defined by

$$(f \,\square\, g)(x) = \inf_{y \in \mathbb{E}} \{f(y) + g(x - y)\}.$$

In light of the concept of infimal convolution, one can also construct smoothing approximation functions. More specifically, we consider $f : \mathbb{E} \to (-\infty, \infty]$ which is a closed proper convex function and let $\omega : \mathbb{E} \to \mathbb{R}$ be a $C^{1,1}$ convex function with Lipschitz gradient constant $1/\sigma$ ($\sigma > 0$). Suppose that for any $\mu > 0$ and any $x \in \mathbb{E}$, the following infimal convolution is finite:

$$f_\mu^{\rm ic}(x) = \inf_{u \in \mathbb{E}} \left\{ f(u) + \mu\omega(\frac{x-u}{\mu}) \right\} = (f \,\square\, \omega_\mu)(x), \qquad (13)$$

where $\omega_\mu(\cdot) = \mu\omega(\frac{\cdot}{\mu})$. Then, $f_\mu^{\rm ic}$ is called the infimal-convolution μ-smooth approximation of f.

Example by infimal-convolution

In particular, when $\mu \in \mathbb{R}_{++}$ and $p \in (1, +\infty)$, the infimal convolution of a convex function and a power of the norm function is obtained as below:

$$f \,\square \left(\frac{1}{\mu p} \| \cdot \|^p \right) = \inf_{u \in \mathbb{E}} \left\{ f(u) + \left(\frac{1}{\mu p} \|x - u\|^p \right) \right\}. \qquad (14)$$

For the absolute value function, it can be verified that $f_\mu(x) = (| \cdot |) \,\square \left(\frac{1}{\mu * p} | \cdot |^p \right)$ is the Huber function of order p, i.e.,

$$f_\mu(x) = \begin{cases} |x| - \frac{p-1}{p} \mu^{\frac{1}{p-1}}, & \text{if} \quad |x| > \mu^{\frac{1}{p-1}}, \\ \frac{|x|^p}{\mu p}, & \text{if} \quad |x| \le \mu^{\frac{1}{p-1}}. \end{cases} \qquad (15)$$

Remarks about Huber function of order p

Remarks

- Note that when $p = 2$ in the above expression (15), the Huber function of order p reduces to the Huber function $\phi_1(x, \mu)$.
- To the contrast, plugging $p = 2$ into infimal convolution formula yields the Moreau approximation (11).

The graphs of Huber function of order p

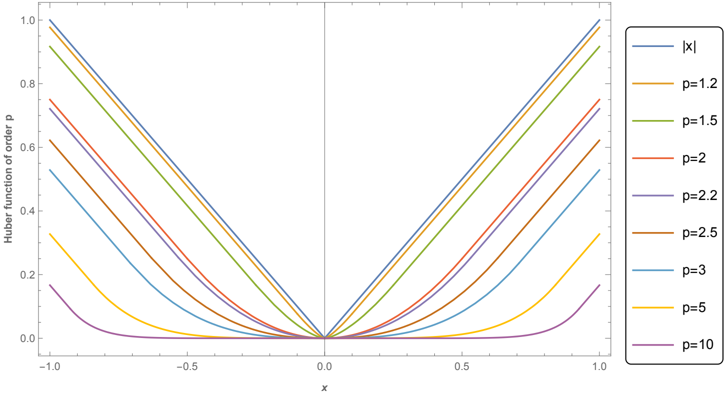


Figure: $|x|$ and Huber function of order p ($\mu = 0.3$).

A unified framework for these four ways

The infimal-convolution way covers the conjugate smoothing way, the Moreau smoothing way, and the Nesterov's smoothing way.

- Conjugate smoothing: taking $\omega_\mu(u) = \mu d^*(u)$ with d being a prox-function yields

$$f_\mu^{ic}(x) = \max_{u \in \mathrm{dom}(f)} \{\langle u, x\rangle - f^*(u) - \mu d(u)\}.$$

- Moreau smoothing: taking $\omega_\mu(u) = \frac{1}{2\mu}\|u\|^2$ yields

$$f_\mu^{ic}(x) = \inf_{u \in \mathbb{E}} \{f(u) + \frac{1}{2\mu}\|u - x\|^2\}.$$

- Nesterov's smoothing: taking $\omega_\mu = \mu d^*(u)$ with d being a prox-function yields

$$f_\mu^{ic}(x) = \max_{u \in Q} \{\langle u, Ax\rangle - f^*(u) - \mu d(u)\}, \ x \in \mathbb{E}_1.$$

The convolution smoothing idea follows by three steps.

Step 1. First, one constructs a smoothing approximation for the plus function $(x)_+ = \max\{0, x\}$. To this end, we consider the piecewise continuous function $d(x)$ with finite number of pieces which is a density (kernel) function, that is, it satisfies

$$d(x) \geq 0 \quad \text{and} \quad \int_{-\infty}^{+\infty} d(x)dx = 1.$$

Step 2. Next, define $\hat{s}(x,\mu) := \frac{1}{\mu} d\left(\frac{x}{\mu}\right)$, where μ is a positive parameter. Suppose that $\int_{-\infty}^{+\infty} |x|\, d(x)dx < +\infty$, then a smoothing approximation (denoted by $\hat{p}(x,\mu)$) for $(x)_+$ is obtained as below:

$$\hat{p}(x,\mu) = \int_{-\infty}^{+\infty} (x-s)_+ \hat{s}(s,\mu)ds = \int_{-\infty}^{x} (x-s)\hat{s}(s,\mu)ds.$$

In other words,

$$\hat{p}(x,\mu) \approx (x)_+.$$

There are four well-known smoothing functions for the plus function:

$$
\begin{aligned}
\hat{p}_1(x,\mu) &= x + \mu \ln\left(1 + e^{-\frac{x}{\mu}}\right) \\
\hat{p}_2(x,\mu) &= \begin{cases} x & \text{if} \quad x \geq \frac{\mu}{2}, \\ \frac{1}{2\mu}\left(x + \frac{\mu}{2}\right)^2 & \text{if} \quad -\frac{\mu}{2} < x < \frac{\mu}{2}, \\ 0 & \text{if} \quad x \leq -\frac{\mu}{2}. \end{cases} \\
\hat{p}_3(x,\mu) &= \frac{\sqrt{4\mu^2 + x^2} + x}{2} \\
\hat{p}_4(x,\mu) &= \begin{cases} x - \frac{\mu}{2} & \text{if} \quad x > \mu, \\ \frac{x^2}{2\mu} & \text{if} \quad 0 \leq x \leq \mu, \\ 0 & \text{if} \quad x < 0. \end{cases}
\end{aligned}
$$

The corresponding kernel functions

Their corresponding kernel functions are

$$
\begin{aligned}
d_1(x) &= \frac{e^{-x}}{(1+e^{-x})^2}, \\
d_2(x) &= \begin{cases} 1 & \text{if } -\frac{1}{2} \le \text{x} \le \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases} \\
d_3(x) &= \frac{2}{(x^2+4)^{\frac{3}{2}}}, \\
d_4(x) &= \begin{cases} 1 & \text{if } 0 \le \text{x} \le 1, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

Step 3. Using the fact that $|x| = (x)_+ + (-x)_-$. Then, the smoothing function of $|x|$ via convolution can be written as

$$\hat{p}(|x|, \mu) = \hat{p}(x, \mu) + \hat{p}(-x, \mu) = \int_{-\infty}^{+\infty} |x - s|\, \hat{s}(s, \mu) ds.$$

In other words,

$$\hat{p}(|x|, \mu) \approx |x|.$$

The smoothing functions ϕ_4, ϕ_5, ϕ_6

Applying the aforementioned four kernel functions, we obtain the following smoothing functions for $|x|$:

$$\begin{aligned}
\phi_4(x,\mu) &= \mu\left[\ln\left(1+e^{-\frac{x}{\mu}}\right)+\ln\left(1+e^{\frac{x}{\mu}}\right)\right], \\
\phi_5(x,\mu) &= \begin{cases} x & \text{if } x \geq \frac{\mu}{2}, \\ \frac{x^2}{\mu}+\frac{\mu}{4} & \text{if } -\frac{\mu}{2} < x < \frac{\mu}{2}, \\ -x & \text{if } x \leq -\frac{\mu}{2}, \end{cases} \\
\phi_6(x,\mu) &= \sqrt{4\mu^2+x^2},
\end{aligned}$$

as well as the Huber function $\phi_1(x,\mu)$.

The smoothing function $\phi_7(x, \mu)$

If we take a Epanechnikov kernel function

$$K(x) = \begin{cases} \frac{3}{4}(1 - x^2) & \text{if } |x| \leq 1, \\ 0 & \text{otherwise}, \end{cases}$$

we achieve the smoothing function for $|x|$:

$$\phi_7(x, \mu) = \begin{cases} x & \text{if } \quad x > \mu, \\ -\frac{x^4}{8\mu^3} + \frac{3x^2}{4\mu} + \frac{3\mu}{8} & \text{if } \quad -\mu \leq x \leq \mu, \\ -x & \text{if } \quad x < \mu. \end{cases}$$

The smoothing function $\phi_8(x,\mu)$

If we take a Gaussian kernel function $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ for all $x \in \mathbb{R}$. Then, it yields

$$\hat{s}(x,\mu) := \frac{1}{\mu} K\left(\frac{x}{\mu}\right) = \frac{1}{\sqrt{2\pi\mu^2}} e^{-\frac{x^2}{2\mu^2}}.$$

Hence, we obtain the below smoothing function for $|x|$:

$$\phi_8(x,\mu) = x \,\mathrm{erf}\left(\frac{x}{\sqrt{2}\mu}\right) + \sqrt{\frac{2}{\pi}} \mu e^{-\frac{x^2}{2\mu^2}}.$$

where the error function is defined by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du, \quad \forall x \in \mathbb{R}.$$

Eight smoothing functions (1)

$$
\begin{aligned}
\phi_1(x,\mu) &= \sup_{|z|\le 1}\left\{zx - \frac{\mu}{2}z^2\right\} = \begin{cases} \frac{x^2}{2\mu}, & \text{if } \ |x|\le\mu, \\ |x| - \frac{\mu}{2}, & \text{otherwise.} \end{cases} \\
\phi_2(x,\mu) &= \mu \log\left(\cosh\left(\frac{x}{\mu}\right)\right). \\
\phi_3(x,\mu) &= \sup_{-1\le y\le 1}\left(xy + \mu\sqrt{1-y^2} - \mu\right) = \sqrt{x^2+\mu^2} - \mu. \\
\phi_4(x,\mu) &= \mu\left[\ln\left(1+e^{-\frac{x}{\mu}}\right) + \ln\left(1+e^{\frac{x}{\mu}}\right)\right].
\end{aligned}
$$

Eight smoothing functions (2)

$$\phi_5(x,\mu) = \begin{cases} x & \text{if} \quad x \geq \frac{\mu}{2}, \\ \frac{x^2}{\mu} + \frac{\mu}{4} & \text{if} \quad -\frac{\mu}{2} < x < \frac{\mu}{2}, \\ -x & \text{if} \quad x \leq -\frac{\mu}{2}, \end{cases}$$

$$\phi_6(x,\mu) = \sqrt{4\mu^2 + x^2}.$$

$$\phi_7(x,\mu) = \begin{cases} x & \text{if} \quad x > \mu, \\ -\frac{x^4}{8\mu^3} + \frac{3x^2}{4\mu} + \frac{3\mu}{8} & \text{if} \quad -\mu \leq x \leq \mu, \\ -x & \text{if} \quad x < \mu. \end{cases}$$

$$\phi_8(x,\mu) = x \operatorname{erf}\left(\frac{x}{\sqrt{2}\mu}\right) + \sqrt{\frac{2}{\pi}}\mu e^{-\frac{x^2}{2\mu^2}}.$$

The graphs of all eight smoothing functions

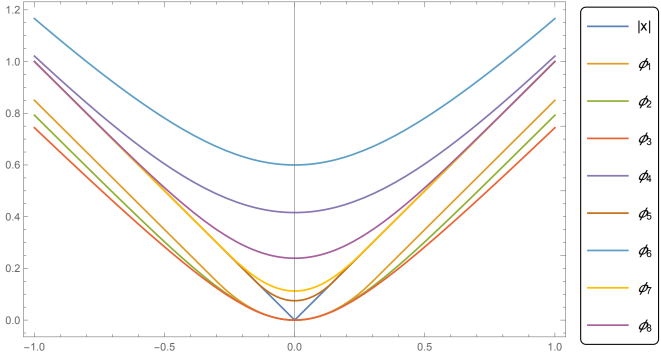


Figure: The graphs of $|x|$ and the smoothing functions $\phi_i, i = 1, \cdots, 8$ ($\mu = 0.3$).

Comparison of all smoothing functions

The local behavior of all eight smoothing functions can be verified (theoretically and geometrically) as

$$\phi_3 \leq \phi_2 \leq \phi_1 \leq |x| \leq \phi_5 \leq \phi_7 \leq \phi_8 \leq \phi_4 \leq \phi_6.$$

- Three smoothing function ϕ_1, ϕ_2, ϕ_3 approach to $|x|$ from below with $\phi_1 \geq \phi_2 \geq \phi_3$.
- The other five smoothing functions ϕ_4, ϕ_5, ϕ_6, ϕ_7, ϕ_8 approach to $|x|$ from above with $\phi_5 \leq \phi_7 \leq \phi_8 \leq \phi_4 \leq \phi_6$.
- Apparently, the smoothing function ϕ_1 and ϕ_5 are closest to $|x|$ among these smoothing functions.

Reformulation

Like the idea in first approach, we define $H_i : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ as

$$H_i(\mu, x) = \begin{bmatrix} \mu \\ Ax + B\Phi_i(\mu, x) - b \end{bmatrix} \quad \text{for } \mu \in \mathbb{R} \text{ and } x \in \mathbb{R}^n$$

where $\Phi_i : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is given by

$$\Phi_i(\mu, x) := \begin{bmatrix} \phi_i(\mu, x_1) \\ \phi_i(\mu, x_2) \\ \vdots \\ \phi_i(\mu, x_n) \end{bmatrix} \quad \text{for } \mu \in \mathbb{R} \text{ and } x \in \mathbb{R}^n$$

with various smoothing functions $\phi_i : \mathbb{R}^2 \to \mathbb{R}$

Then, the AVE can be transformed into an unconstrained optimization problem:

$$\min \Psi(\mu, x) = \frac{1}{2}\|H_i(\mu, x)\|^2.$$

For neural network approach, we consider the system of differential equation:

$$\begin{cases} \dfrac{du(t)}{dt} = -\rho\,\nabla\Psi(v(t), u(t)) = -\rho\nabla H_i(v(t), u(t))^{\mathrm{T}} H_i(v(t), u(t)), \\ u(t_0) = u_0. \end{cases}$$

where $u_0 = x_0 \in \mathbb{R}^n$, $v(t) = \mu_0 e^{-t}$, $\rho > 0$ is a time scaling factor.

Theorem (Saheya-Nguyen-Chen, JAMC, 2019)

Let $\Psi : \mathbb{R}^n \to \mathbb{R}_+$ *be defined as above. Then, the following results hold.*

(a) $\Psi(x) \geq 0, \forall (\mu, t) \in \mathbb{R}_{++} \times \mathbb{R}$ *and* $\Psi(\mu, x) = 0$ *if and only if* x *solve the AVE* (2).

(b) *The function* $\Psi(x)$ *is continuously differentiable on* $\mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$ *with*

$$\nabla \Psi(\mu, x) = \nabla H^{\mathrm{T}} H(\mu, x),$$

where ∇H *is the Jacobian of* $H(\mu, x)$.

(c) *The function* $\Psi(w(t))$ *is nonincreasing with respect to* t.

Equilibrium of neural network

Theorem (Saheya-Nguyen-Chen, JAMC, 2019)

Let x^* be a equilibrium of the neural network and suppose that the singular values of $A \in \mathbb{R}^{n \times n}$ exceed 1. Then, x^* solves the AVE (1).

Theorem (Saheya-Nguyen-Chen, JAMC, 2019)

(a) *For any initial point $w_0 = w(t_0)$, there exists a unique continuously maximal solution $w(t)$ with $t \in [t_0, \tau)$ for the neural network.*

(b) *If the level set $\mathcal{L}(w_0) := \left\{ w \,|\, \|H_i(w)\|^2 \le \|H(w_0)\|^2 \right\}$ is bounded, then τ can be extended to ∞.*

Stabilities of neural network

Theorem (Saheya-Nguyen-Chen, JAMC, 2019)

If the singular values of $A \in \mathbb{R}^{n\times n}$ exceed 1, then the isolated equilibrium x^* of the neural network is asymptotically stable, and hence Lyapunov stable.

Theorem (Saheya-Nguyen-Chen, JAMC, 2019)

If the singular values of $A \in \mathbb{R}^{n\times n}$ exceed 1, then the isolated equilibrium x^* of the neural network is exponentially stable.

The numerical implementation is coded by Mathematica 11.3 and the ordinary differential equation solver adopted is NDSolve[ ], which uses an Runge-Kutta (2,3) formula. The initial point of each problems are selected by randomly and the initial point is same for different smoothing functions.

ϕ_i	the smoothing functions $\phi_i, i = 1, \cdots, 8$
N	the number of iterations
t	the time when algorithm terminates
Er	the value of $\|x(t) - x^*\|$ when algorithm terminates
$H(x_t)$	the value of $\|H(x(t)) = \|Ax - \|x\| - b\|$ when terminates
CT	the CPU time in seconds

Test Example 1

Example

Consider the following absolute value equation where

$$A = \begin{pmatrix} 10 & 1 & 2 & 0 \\ 1 & 11 & 3 & 1 \\ 0 & 2 & 12 & 1 \\ 1 & 7 & 0 & 13 \end{pmatrix}, \quad b = \begin{pmatrix} 12 \\ 15 \\ 14 \\ 20 \end{pmatrix}.$$

Numerical reports for Example 1

Table: computing results of Example 1 (dt=0.2)

function	N	t	Er	$H(x_0)$	CT
ϕ_1	34	6.8	$9.3686 * 10^{-7}$	0.0000136037	1.5090863
ϕ_2	36	7.2	$8.70587 * 10^{-7}$	0.0000126414	0.7760444
ϕ_3	38	7.6	$8.41914 * 10^{-7}$	0.000012225	0.4980285
ϕ_4	2	0.4	$2.90785 * 10^{-15}$	$1.59872 * 10^{-14}$	0.0340019
ϕ_5	2	0.4	$1.11772 * 10^{-12}$	$8.41527 * 10^{-12}$	0.0740043
ϕ_6	10	2.	$7.52691 * 10^{-7}$	0.0000109295	0.1150066
ϕ_7	2	0.4	$1.29976 * 10^{-12}$	$8.61527 * 10^{-12}$	0.0730042
ϕ_8	34	6.8	$9.3686 * 10^{-7}$	0.0000136037	0.6880393

Numerical comparison for Example 1

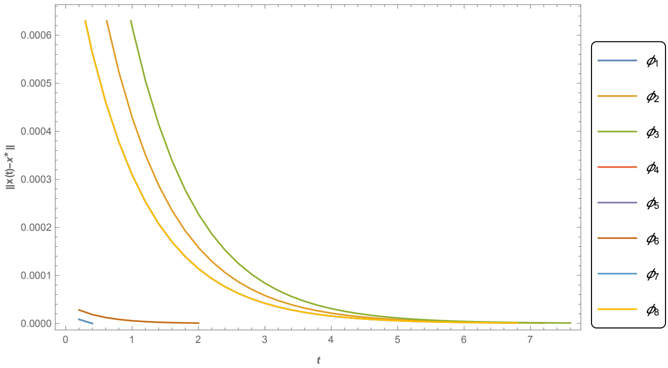


Figure: Convergence behaviour of the error $\|x(t) - x^*\|$ for $\phi_i, i = 1, \cdots, 8$ in Example 1 (dt=0.2).

The smoothing functions ϕ_4, ϕ_5, ϕ_7 perform better than others (followed by ϕ_6).

Test Example 2

Example

Consider the following linear complementary problem: find $x \in \mathbb{R}^n$ such that

$$x \geq 0, \quad Mx + q \geq 0, \quad \langle x, Mx + q \rangle = 0,$$

where

$$M = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}.$$

The LCP can be transformed into an AVE where

$$A = \begin{pmatrix} 2 & -3 & 6 & -12 \\ 0 & 2 & -3 & 6 \\ 0 & 0 & 2 & -3 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 24 \\ -12 \\ 6 \\ -3 \end{pmatrix}.$$

Numerical reports for Example 2

Table: computing results of Example 2 (dt=0.2)

function	N	T	Er	$H(x_0)$	CT
ϕ_1	55	11	$9.84216 * 10^{-7}$	$2.03995 * 10^{-7}$	3.1531804
ϕ_2	57	11.4	$9.14593 * 10^{-7}$	$1.89565 * 10^{-7}$	1.3690783
ϕ_3	59	11.8	$8.84473 * 10^{-7}$	$1.83322 * 10^{-7}$	0.6920396
ϕ_4	2	0.4	$2.67859 * 10^{-9}$	$4.86096 * 10^{-10}$	0.0370021
ϕ_5	2	0.4	$2.68658 * 10^{-9}$	$4.87548 * 10^{-10}$	0.1510086
ϕ_6	18	3.6	$9.56666 * 10^{-7}$	$2.57215 * 10^{-7}$	0.2210127
ϕ_7	2	0.4	$2.16413 * 10^{-9}$	$3.92736 * 10^{-10}$	0.1600091
ϕ_8	55	11	$9.84216 * 10^{-7}$	$2.03995 * 10^{-7}$	1.5320877

Numerical comparison for Example 2

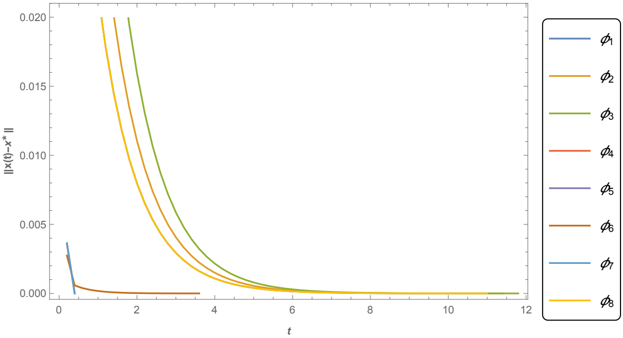


Figure: Convergence behaviour of the error $\|x(t) - x^*\|$ for $\phi_i, i = 1, \cdots, 8$ in Example 2.

The smoothing functions ϕ_4, ϕ_5, ϕ_7 perform better than others (followed by ϕ_6).

Test Example 3

Example

Consider the following linear complementarity problem: find $x \in \mathbb{R}^n$ such that

$$x \geq 0, \quad Mx + q \geq 0, \quad \langle x, Mx + q \rangle = 0,$$

where

$$M = \begin{pmatrix} 1 & -4 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} -5 \\ -5 \\ 1 \\ 1 \end{pmatrix}.$$

Reformulation of Example 3

Likewise, we can transform this linear complementarity problem into an AVE, where

$$A = \begin{pmatrix} -1 & 8 & -2 & 8 \\ 0 & -1 & 0 & -2 \\ 2 & -8 & 1 & -8 \\ 0 & 2 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} -24 \\ 8 \\ 22 \\ -10 \end{pmatrix}.$$

Numerical reports for Example 3

Table: computing results of Example 3 (dt=0.2)

function	N	T	Er	$H(x_0)$	CT
ϕ_1	48	9.6	$8.77421 * 10^{-7}$	$8.27241 * 10^{-7}$	2.6401510
ϕ_2	49	9.8	$9.95875 * 10^{-7}$	$9.3892 * 10^{-7}$	1.0670610
ϕ_3	51	10.2	$9.63078 * 10^{-7}$	$9.07998 * 10^{-7}$	0.6060347
ϕ_4	9	1.8	$1.88878 * 10^{-8}$	$8.88851 * 10^{-9}$	0.2820161
ϕ_5	9	1.8	$1.89870 * 10^{-8}$	$8.93517 * 10^{-9}$	0.3770216
ϕ_6	15	3.0	$7.23527 * 10^{-7}$	$1.06615 * 10^{-6}$	0.1870107
ϕ_7	9	1.8	$1.85496 * 10^{-8}$	$8.72934 * 10^{-9}$	0.4020229
ϕ_8	48	9.6	$8.77421 * 10^{-7}$	$8.27241 * 10^{-7}$	1.2330706

Numerical comparison for Example 3

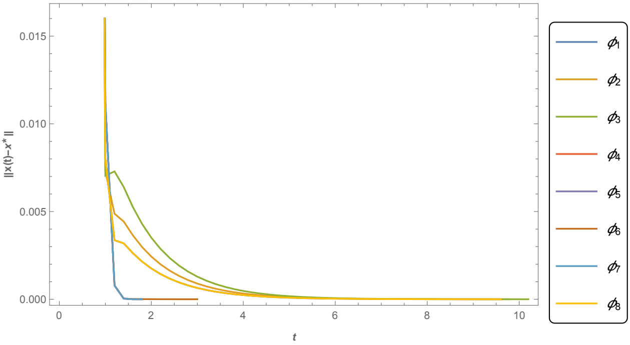


Figure: Convergence behaviour of the error $\|x(t) - x^*\|$ for $\phi_i, i = 1, \cdots, 8$ in Example 3.

The performance of the smoothing function ϕ_4, ϕ_5, ϕ_6, ϕ_7 is better than others.

Example

Consider the AVE, where the matrix A of which all the singular values are greater than 1 is generated by the following Mathematica procedure:

```
R = RandomInteger[0,50,n,n];
A = R.R + n*IdentityMatrix[n];
b = (A - IdentityMatrix[n]).Table[1,n];
```

Table: computing results of Example 4

function	N	T	Er	$H(x_0)$	CT
ϕ_1	58	5.8	$9.97651 * 10^{-7}$	0.000118298	465.6496337
ϕ_2	62	6.2	$9.27099 * 10^{-7}$	0.000109929	191.1859352
ϕ_3	65	6.5	$9.90869 * 10^{-7}$	0.000117487	162.2222786
ϕ_4	4	0.4	$1.48112 * 10^{-7}$	$5.51822 * 10^{-7}$	11.9286822
ϕ_5	4	0.4	$1.49218 * 10^{-7}$	$5.55955 * 10^{-7}$	32.2308435
ϕ_6	14	1.4	$8.85812 * 10^{-7}$	0.000105037	35.2450159
ϕ_7	4	0.4	$1.48181 * 10^{-7}$	$5.52006 * 10^{-7}$	33.0088880
ϕ_8	58	5.8	$9.97689 * 10^{-7}$	0.000118298	207.7848846

Numerical comparison for Example 4

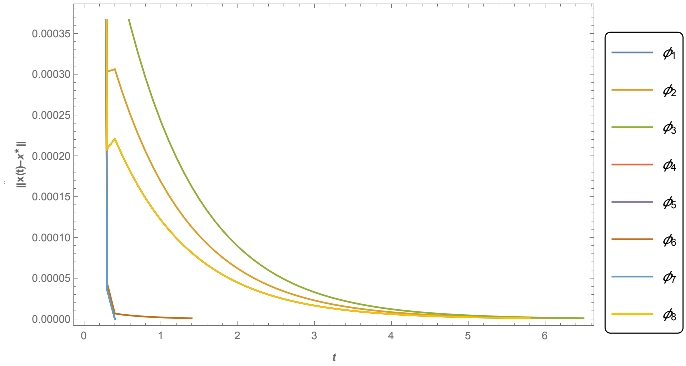


Figure: Convergence behaviour of the error $\|x(t) - x^*\|$ for $\phi_i, i = 1, \cdots, 8$ in Example 4.

For this AVE with high dimension, the smoothing functions ϕ_4, ϕ_5, ϕ_6, ϕ_7 is again the leading group regarding the efficiency.

Numerical Observations

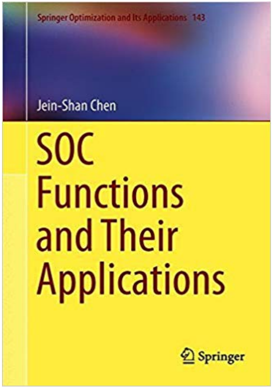
Numerical Summary

- In general, the smoothing functions ϕ_4, ϕ_5, ϕ_6, ϕ_7 are effective functions that work well along with the neural network.
- In particular, all these smoothing functions are produced from the convolution way.
- The other ways like convex conjugate way, Moreau proximal way, Nesterov's smoothing way, and infimal-convolution way, do not offer effective smoothing functions for the proposed neural network approach. This is a very interesting discovery, which deserves further investigation.

Final remarks

- If we are given two smoothing functions ψ_1 and ψ_2 for f, then $t\psi_1 + (1-t)\psi_2$ is also a smoothing function for f. This means that any convex combination of two smoothing functions for $|x|$ is again a smoothing function for $|x|$.
- In particular, we choose $\psi_1 \in \{\phi_1, \phi_2, \phi_3\}$ and pick another $\psi_2 \in \{\phi_4, \phi_5, \phi_6, \phi_7, \phi_8\}$ to make new smoothing functions for $|x|$. It makes 15 convex combinations and could try different value $t \in [0, 1]$.
- In other words, we can obtain many more smoothing functions. How do these types of smoothing functions perform? We leave it as our future study.

The End



∼ Thanks for your attention ∼