Extrapolation methods and their applications in image reconstruction and restoration

Anna Karapiperi
Lancaster University, University of Padova

NASCA 2018

Kalamata, 2 July 2018.

Outline

1. Scalar extrapolation
   - Shanks transformation
   - The ε-algorithm and its particular rules
2. Vector extrapolation
   - Topological Shanks transformation and the simplified topological ε-algorithm (STEA)
   - Image reconstruction using STEA

Introduction

Let (S_n) be a sequence of numbers converging slowly to S.
Sequence transformation

$$T : (S_n) \longmapsto (T_n)$$

1. (T_n) must converge
2. (T_n) must converge to the same limit as (S_n)
3. (T_n) must converge to S faster than (S_n)

$$\lim_{n\to\infty} \frac{T_n - S}{S_n - S} = 0$$

The kernel $\mathcal{K}_T$ of a transformation $T : (S_n) \longmapsto (T_n)$ is the set of sequences for which $\exists S$ such that

$$\forall n \geq N, \; T_n = S.$$

Any sequence transformation is an extrapolation method.

The kernel $\mathcal{K}_T$ of a transformation $T : (S_n) \longmapsto (T_n)$ is the set of sequences for which $\exists S$ such that

$$\forall n \geq N, \; T_n = S.$$

Any sequence transformation is an extrapolation method.

Any sequence transformation can be applied to a sequence, even if it does not belong to the kernel of the transformation.

A universal transformation able to accelerate any sequence cannot exist (J.P. Delahaye, B. Germain-Bonne, 1980).

Shanks transformation

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}} \quad k, n = 0, 1, \ldots$$

where $\Delta S_n = S_{n+1} - S_n$, $n = 0, 1, \ldots$.

Shanks transformation

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}} \quad k, n = 0, 1, \ldots$$

where $\Delta S_n = S_{n+1} - S_n$, $n = 0, 1, \ldots$.

Its kernel is the set of sequences such that there exist S, $\alpha_0, \alpha_1, \ldots, \alpha_k$, with $\alpha_0 \, \alpha_k \neq 0$ and $\alpha_0 + \cdots + \alpha_k = 1$, satisfying

$$\alpha_0(S_n - S) + \ldots + \alpha_k(S_{n+k} - S) = 0, \quad n = 0, 1, \ldots,$$

The scalar ε-algorithm

For implementing Shanks transformation, Wynn (1963) proposed the scalar ε-algorithm

$$\begin{cases} \varepsilon_{-1}^{(n)} = 0, \quad \varepsilon_0^{(n)} = S_n, & n = 0, 1, ... \\ \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + (\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)})^{-1}, & k, n = 0, 1, ... \end{cases}$$

$$e_k(S_n) = \varepsilon_{2k}^{(n)}, \quad k, n = 0, 1, \ldots$$

Theorem

For all $n \geq N$ ($N > 0$),

$$e_k(S_n) = S$$

if and only if there exist $\alpha_0, \alpha_1, \ldots, \alpha_k$, with $\alpha_0 \, \alpha_k \neq 0$ and $\alpha_0 + \cdots + \alpha_k \neq 0$, such that, for all n,
$\alpha_0(S_n - S) + \cdots + \alpha_k(S_{n+k} - S) = 0.$

The numbers $\varepsilon_k^{(n)}$ are displayed in a double entry table, called the ε-*array*.

$$
\begin{array}{ccccccc}
\varepsilon_{-1}^{(0)} = 0 & & & & & & \\
 & \varepsilon_0^{(0)} = S_0 & & & & & \\
\varepsilon_{-1}^{(1)} = 0 & & \varepsilon_1^{(0)} & & & & \\
 & \varepsilon_0^{(1)} = S_1 & & \varepsilon_2^{(0)} & & & \\
\varepsilon_{-1}^{(2)} = 0 & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & & \\
 & \varepsilon_0^{(2)} = S_2 & & \varepsilon_2^{(1)} & & \varepsilon_4^{(0)} & \\
\varepsilon_{-1}^{(3)} = 0 & & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & \vdots & \ddots \\
 & \varepsilon_0^{(3)} = S_3 & & \varepsilon_2^{(2)} & \vdots & & \\
\varepsilon_{-1}^{(4)} = 0 & & \varepsilon_1^{(3)} & \vdots & & & \\
 & \varepsilon_0^{(4)} = S_4 & \vdots & & & & \\
\varepsilon_{-1}^{(5)} = 0 & \vdots & & & & & \\
\vdots & & & & & &
\end{array}
$$

An isolated singularity

$$\begin{array}{ccccc} & & N & & \\ & NW \sim \alpha & & NE \sim \alpha & \\ W & & C \sim \infty & & E(*) \\ & SW \sim \alpha & & SE \sim \alpha & \\ & & S & & \end{array}$$

$$\begin{aligned} C &= W + 1/(SW - NW), \\ NE &= NW + 1/(C - N), \\ SE &= SW + 1/(S - C), \\ E &= C + 1/(SE - NE). \end{aligned}$$

An isolated singularity

$$\begin{array}{ccccc} & & N & & \\ & NW \sim \alpha & & NE \sim \alpha & \\ W & & C \sim \infty & & E(*) \\ & SW \sim \alpha & & SE \sim \alpha & \\ & & S & & \end{array}$$

$$\begin{aligned} C &= W + 1/(SW - NW), \\ NE &= NW + 1/(C - N), \\ SE &= SW + 1/(S - C), \\ E &= C + 1/(SE - NE). \end{aligned}$$

Wynn's particular rule: $E = r(1 + r/C)^{-1}$
where $r = S(1 - S/C)^{-1} + N(1 - N/C)^{-1} - W(1 - W/C)^{-1}$.

If $C = \infty$, then $E = S + N - W$.

A non-isolated singularity

$\varepsilon_{k-1}^{(n)}$		$\varepsilon_{k+1}^{(n-1)} = N_1$		$\cdots$		$\varepsilon_{k+2m-1}^{(n-m)} = N_m$		$\varepsilon_{k+2m+1}^{(n-m-1)}$
	$\varepsilon_{k}^{(n)} \sim \alpha$		$\cdots$		$\cdots$		$\varepsilon_{k+2m}^{(n-m)} \sim \alpha$	
W_1		$\varepsilon_{k+1}^{(n)} \sim C$		$\cdots$		$\varepsilon_{k+2m-1}^{(n-m+1)} \sim C$		E_m
	$\varepsilon_{k}^{(n+1)} \sim \alpha$		ω	$\cdots$	ω		$\varepsilon_{k+2m}^{(n-m+1)} \sim \alpha$	
$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
	$\vdots$		ω	$\cdots$	ω		$\vdots$	
W_m		$\varepsilon_{k+1}^{(n+m-1)} \sim C$		$\cdots$		$\varepsilon_{k+2m-1}^{(n)} \sim C$		E_1
	$\varepsilon_{k}^{(n+m)} \sim \alpha$		$\cdots$		$\cdots$		$\varepsilon_{k+2m}^{(n)} \sim \alpha$	
$\varepsilon_{k-1}^{(n+m+1)}$		$\varepsilon_{k+1}^{(n+m)} = S_m$		$\cdots$		$\varepsilon_{k+2m-1}^{(n+1)} = S_1$		$\varepsilon_{k+2m+1}^{(n)}$

Generalized particular rule (Cordellier, 1979)

$$E_i = r_i(1 + r_i/C)^{-1}, \quad i = 1, \ldots, m,$$

where $$r_i = S_i(1 - S_i/C)^{-1} + N_i(1 - N_i/C)^{-1} - W_i(1 - W_i/C)^{-1}.$$

Implementation

- `EPSALGO`: the classical ε-algorithm (without particular rules);
- `EPSALGOW`: ε-algorithm with Wynn's particular rules for treating isolated singularities;
- `EPSALGOC`: ε-algorithm with Cordellier's particular rules for treating non-isolated singularities caused by exactly equal values (Cordellier, 1989);
- `EPSALGOG`: the new algorithm implementing ε-algorithm with the generalized particular rules for treating non-isolated singularities caused by exactly or almost equal values.

Example. (Non-isolated singularities, Almost equal values)
We consider the sequence given by

$$S_0 = 0.999999999999,\ S_1 = 1,\ S_2 = 1.000000000001,$$
$$S_3 = 1.5,\ S_n = 3S_{n-4} \text{ for n} \geq 4.$$

According to the theory, for all n, $e_4(S_n) = 0$, thus $\varepsilon_8^{(n)} = 0$.

n	EPSALGO	EPSALGOG
0	1.9587e+00	1.0658e-14
1	Inf	-8.8818e-16
2	4.3333e+00	-1.1102e-14
3	-4.0343e+00	1.0658e-14
4	5.8761e+00	-3.1974e-14
5	7.9092e+00	1.7764e-15

Part of the ε-array (columns ε_4 to ε_8) obtained by `EPSALGO` algorithm.

1.0000e+00				
	1.2000e+01			
1.0714e+00		9.7500e-01		
	1.6296e+00		2.3382e+00	
3.0000e+00		2.3864e+00		1.9587e+00
	0		0	
3.0000e+00		Inf		Inf
	0		0	
3.0000e+00		3.2500e+00		4.3333e+00
	4.0000e+00		9.2308e-01	
3.2143e+00		2.9250e+00		-4.0343e+00
	5.4321e-01		7.7939e-01	
9.0000e+00		7.1591e+00		5.8761e+00
	-1.5259e-05		1.5257e-05	
9.0000e+00		3.2777e+04		7.9092e+00
	1.5259e-05		-1.5259e-05	
9.0000e+00		9.7500e+00		
	1.3333e+00			
9.6429e+00				

Part of the ε-array (columns ε_4 to ε_8) obtained by `EPSALGOG` algorithm.

1.0000e+00				
	1.2000e+01			
1.0714e+00		9.7500e-01		
	1.6296e+00		1.7436e+00	
3.0000e+00		9.7500e+00		1.0658e-14
	1.7778e+00		1.6410e+00	
3.0000e+00		2.4375e+00		-8.8818e-16
	-2.6683e-12		1.2308e+00	
3.0000e+00		3.2500e+00		-1.1102e-14
	4.0000e+00		9.2308e-01	
3.2143e+00		2.9250e+00		1.0658e-14
	5.4321e-01		5.8120e-01	
9.0000e+00		2.9250e+01		-3.1974e-14
	5.9259e-01		5.4701e-01	
9.0000e+00		7.3125e+00		1.7764e-15
	-8.8954e-13		4.1026e-01	
9.0000e+00		9.7500e+00		
	1.3333e+00			
9.6429e+00				

Vector extrapolation

Topological Shanks transformation

(Brezinski, 1975)
The kernel of the topological Shanks transformation is

$$\alpha_0(\mathbf{x}_n - \mathbf{x}) + \cdots + \alpha_k(\mathbf{x}_{n+k} - \mathbf{x}) = \mathbf{0} \quad n = 0, 1, \ldots$$

$\mathbf{x}_i, \mathbf{x} \in E$ (topological vector space) , α_i's scalars satisfying $\alpha_0\,\alpha_k \neq 0$, $\alpha_0 + \cdots + \alpha_k \neq 0$.

Let $\mathbf{y}$ be an arbitrary vector in E^* (algebraic dual space of E).

The simplified topological ε-algorithm

(Brezinski,Redivo-Zaglia, 2014)
The main idea is to apply the scalar ε-algorithm of Wynn to the sequence

$$(S_n) = (\langle \mathbf{y}, \mathbf{x}_n \rangle).$$

Notation: The bilinear form $< \mathbf{y}, \mathbf{u} >$ *is defined as* $< \mathbf{y}, \mathbf{u} >= \sum_{i=1}^{N} y_i u_i$ *where the* y_i*'s and the* u_i*'s are the components of* $\mathbf{y}$ *and* $\mathbf{u}$ *respectively.*

STEA

$$\tilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \tilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+2)} - \varepsilon_{2k}^{(n+1)}} (\tilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)} - \tilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}), \quad k, n = 0, 1, \ldots,$$

with $\tilde{\boldsymbol{\varepsilon}}_0^{(n)} = \mathbf{x}_n \in E$, $n = 0, 1, \ldots$

Applications in imaging

(S. Gazzola, A. K., 2016)

We want to solve a linear inverse problem that arise in

- tomography (reconstruct an image of an inspected object given a set of projections of it)
- image deblurring (restore a distorted image given the so-called point spread function (PSF), which describes the distortion of each pixel of the image)

Upon discretization, a linear system of the form

$$A\mathbf{x} + \mathbf{e} = \mathbf{b}$$

is recovered, where $A \in \mathbb{R}^{M \times N}$ and $\mathbf{b} \in \mathbb{R}^M$ are known, and the vector $\mathbf{e} \in \mathbb{R}^M$ represents unknown errors or noise in the measured data $\mathbf{b}$.

Iterative regularization methods for solving an inverse problem:

- Algebraic Reconstruction Techniques (Kaczmarz, Symmetric Kaczmarz, Randomized Kaczmarz)
- Simultaneous Iterative Reconstruction Techniques (Landweber, Cimmino, CAV, DROP, SART)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda_n T A^T D(\mathbf{b} - A\mathbf{x}_n)\,, \quad n = 0, 1, \ldots .$$

- Krylov subspace methods (SD, GMRES, CGLS)

Accelerated STEA

Input $A \in \mathbb{R}^{M\times N}$, $\mathbf{b} \in \mathbb{R}^{M}$, $\mathbf{x}_0 \in \mathbb{R}^{N}$
Compute $\mathbf{x}_1, \mathbf{x}_2$
Compute $\mathbf{z}_0 = \tilde{\boldsymbol{\varepsilon}}_2^{(0)}$
for $n = 2, 3, \ldots$ until a stopping rule is satisfied **do**
    Compute $\mathbf{x}_{n+1}$
    Compute $\mathbf{z}_{n-1} = \begin{cases} \tilde{\boldsymbol{\varepsilon}}_{n+1}^{(0)}, & \text{if } n \text{ is odd;} \\ \tilde{\boldsymbol{\varepsilon}}_{n}^{(1)}, & \text{if } n \text{ is even.} \end{cases}$
end for n

Restarted STEA

Input $A \in \mathbb{R}^{M\times N}$, $\mathbf{b} \in \mathbb{R}^{M}$, $\mathbf{x}_0^{(0)} \in \mathbb{R}^{N}$.
for $j = 0, 1, \ldots$ until a stopping rule is satisfied **do**
  for $n = 0, ..., \ell$ **do**
    Compute $\mathbf{x}_{n+1}^{(j)}$
  end for n
  Compute $\mathbf{z}_j = \begin{cases} \tilde{\boldsymbol{\varepsilon}}_{\ell}^{(0)}, & \text{if } \ell \text{ is even;} \\ \tilde{\boldsymbol{\varepsilon}}_{\ell-1}^{(1)}, & \text{if } \ell \text{ is odd.} \end{cases}$
  Take $\mathbf{x}_0^{(j+1)} = \mathbf{z}_j$.
end for j

Stopping criteria

The following stopping criteria may apply

- ✓ Discrepancy principle (if we assume that a good estimate of the noise level is available, then we can stop as soon as the residual lies below the noise level).
  (Hansen 1998)
- ✓ The ratios $\|\mathbf{z}_{n+1} - \mathbf{z}_n\| / \|\mathbf{x}_{n+1} - \mathbf{x}_n\|$ increase when a good precision is attained.
  (Brezinski, Redivo-Zaglia 2013)

Numerical experiments

Test problem: `seismictomo`, 20000×10000.
We add Gaussian white noise of level $\delta = 5 \cdot 10^{-2}$.

	without acceleration			Accelerated STEA		
	it_{opt}	$\text{error}_{\text{opt}}$	T	it_{opt}	$\text{error}_{\text{opt}}$	T
Landweber (with λ_{AIR})	10	0.2996	2.82	9	0.2077	2.95
Cimmino (with λ_{AIR})	10	0.2764	1.67	10	0.2001	1.97
CAV	10	0.2863	2.41	10	0.1826	2.63
DROP	10	0.3613	1.60	10	0.2352	1.81
SART	10	0.3310	0.57	10	0.2246	0.80

Minimum relative error and computational time averaged over 100 different runs of SIRT methods[1] and their extrapolated versions.

[1] AIR Tools package (Hansen,Saxild-Hansen, 2012)

Cimmino's method (1938): $\lambda = 2$, $T = I$, $D = \frac{1}{\mu}\mathrm{diag}\left(\frac{m_i}{\|\mathbf{a}_i\|_2^2}\right)_{i=1}^{M}$,

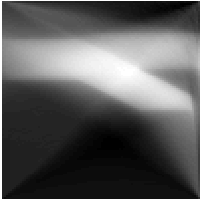
with $\mu = \sum_{i=1}^{M} m_i$ (the m_i's are arbitrary positive quantities).

	Relative errors after 100 iterations
Cimmino method	0.5144
Acc.STEA ($\mathbf{y} = \mathbf{1}$)	0.2244

	4×25	2×50	5×20	20×5	25×4
$\mathbf{y} = \mathbf{1}$	0.2368	0.2557	90.6303	0.4861	0.6438
$\mathbf{y} = \mathbf{r}_0^{(j)}$	**0.1567**	0.2244	0.1956	0.3423	0.3659
$\mathbf{y} = \mathbf{x}_0^{(j)}$	0.2243	0.1843	0.2741	0.4233	0.3169
$\mathbf{y} = -A^T\mathbf{b}$	0.1979	0.2425	0.3685	0.4395	0.6308

Table: Relative errors for Cimmino accelerated by restarted STEA, with various choices for the vector **y**.

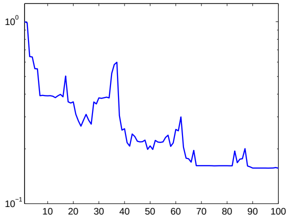
(a) Exact object



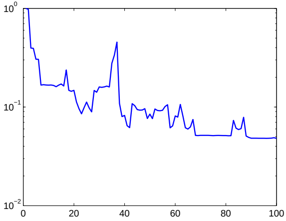
(b) Cimmino, 100 iter.

(c) Acc.STEA ($\mathbf{y} = \mathbf{1}$)

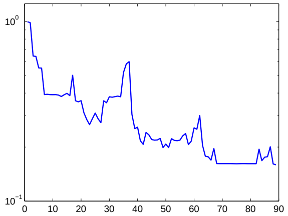
(d) Rest.STEA ($\mathbf{y} = \mathbf{r}_0^{(j)}$)



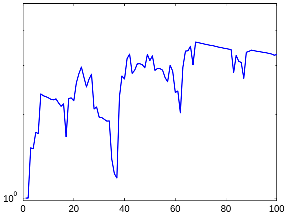
(e) Relative error history



(f) Relative residual history



(g) Relative errors when DP is used



(h) $\|\mathbf{z}_{n+1} - \mathbf{z}_n\| / \|\mathbf{x}_{n+1} - \mathbf{x}_n\|$

S. Gazzola, A. Karapiperi, *Image reconstruction and restoration using the simplified topological ε-algorithm*, Appl. Math. Comput., 274 (2016) 539-555.

Others topics covered:

- ⋆ Insight into the choice of the vector $\mathbf{y}$; the best choice for restarted STEA applied to Landweber is $\mathbf{y} = \mathbf{r}_0^{(j)}$.
- ⋆ Acceleration properties of STEA with $\mathbf{y} = -A^T\mathbf{b}$ applied to the nonnegatively projected Landweber method.
- ⋆ experiments with a deblurring test problem and an astronomical imaging problem include comparisons with other acceleration methods (ν-methods, Richardson-Lucy algorithm), some Krylov subspace methods (CGLS, GMRES, SD) and projected Landweber and SD methods.

References

- C. Brezinski, M. Redivo-Zaglia, *The simplified topological ε-algorithms: software and applications*, M. Numer Algor, 74 (2017) 1237-1260.
- F. Cordellier, *Démonstration algébrique de l'extension de l'identité de Wynn aux tables de Padé non normales*, in: L. Wuytack, ed., Padé Approximation and its Applications, volume 765 of LNM, pp. 36-60, Berlin, 1979. Springer-Verlag.
- S. Gazzola, A. Karapiperi, *Image reconstruction and restoration using the simplified topological ε-algorithm*, Appl. Math. Comput., 274 (2016) 539-555.
- A. Karapiperi, *Extrapolation methods and their applications in Numerical Analysis and Applied Mathematics*, PhD thesis, University of Padova, 2016.
- P. Wynn, *Singular rules for certain non-linear algorithms*, BIT, 3 (1963) 175-195.