Applied Mathematics, 2016, 7, 1260-1278
Published Online July 2016 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2016.711111

SVD-MPE: An SVD-Based Vector Extrapolation Method of Polynomial Type

Avram Sidi

Computer Science Department, Technion-Israel Institute of Technology, Haifa, Israel
Email: asidi@cs.technion.ac.il

Received 10 May 2016; accepted 18 July 2016; published 21 July 2016

Abstract

An important problem that arises in different areas of science and engineering is that of computing the limits of sequences of vectors $\{x_m\}$, where $x_m \in \mathbb{C}^N$, N being very large. Such sequences arise, for example, in the solution of systems of linear or nonlinear equations by fixed-point iterative methods, and $\lim_{m\to\infty} x_m$ are simply the required solutions. In most cases of interest, however, these sequences converge to their limits extremely slowly. One practical way to make the sequences $\{x_m\}$ converge more quickly is to apply to them vector extrapolation methods. Two types of methods exist in the literature: polynomial type methods and epsilon algorithms. In most applications, the polynomial type methods have proved to be superior convergence accelerators. Three polynomial type methods are known, and these are the minimal polynomial extrapolation (MPE), the reduced rank extrapolation (RRE), and the modified minimal polynomial extrapolation (MMPE). In this work, we develop yet another polynomial type method, which is based on the singular value decomposition, as well as the ideas that lead to MPE. We denote this new method by SVD-MPE. We also design a numerically stable algorithm for its implementation, whose computational cost and storage requirements are minimal. Finally, we illustrate the use of SVD-MPE with numerical examples.

Keywords

Vector Extrapolation, Minimal Polynomial Extrapolation, Singular Value Decomposition, Krylov Subspace Methods

1. Introduction and Background

An important problem that arises in different areas of science and engineering is that of computing limits of sequences of vectors $\{\boldsymbol{x}_m\}$[1], where $\boldsymbol{x}_m \in \mathbb{C}^N$, the dimension N being very large in many applications. Such vector sequences arise, for example, in the numerical solution of very large systems of linear or nonlinear equations by fixed-point iterative methods, and $\lim_{m\to\infty} \boldsymbol{x}_m$ are simply the required solutions to these systems. One common source of such systems is the finite-difference or finite-element discretization of continuum problems.

In most cases of interest, however, the sequences $\{\boldsymbol{x}_m\}$ converge to their limits extremely slowly. That is, to approximate $\boldsymbol{s} = \lim_{m\to\infty} \boldsymbol{x}_m$, with a reasonable prescribed level of accuracy, by $\boldsymbol{x}_m$, we need to consider very large values of m. Since the vectors $\boldsymbol{x}_m$ are normally computed in the order $m = 0,1,2,\cdots$, it is clear that we have to compute many such vectors until we reach one that has acceptable accuracy. Thus, this way of approximating $\boldsymbol{s}$ via the $\boldsymbol{x}_m$ becomes very expensive computationally.

Nevertheless, we may ask whether we can do something with those $\boldsymbol{x}_m$ that are already available, to somehow obtain new approximations to $\boldsymbol{s}$ that are better than each individual available $\boldsymbol{x}_m$. The answer to this question is in the affirmative for at least a large class of sequences that arise from fixed-point iteration of linear and nonlinear systems of equations. One practical way of achieving this is by applying to the sequence $\{\boldsymbol{x}_m\}$ a suitable *convergence acceleration method* (or *extrapolation method*).

Of course, in case $\lim_{m\to\infty} \boldsymbol{x}_m$ does not exist, it seems that no use could be made of the $\boldsymbol{x}_m$. Now, if the sequence $\{\boldsymbol{x}_m\}$ is generated by an iterative solution of a linear or nonlinear system of equations, it can be thought of as “diverging from” the solution $\boldsymbol{s}$ of this system. We call $\boldsymbol{s}$ the *antilimit* of $\{\boldsymbol{x}_m\}$ in such a case. It turns out that vector extrapolation methods can be applied to such divergent sequences $\{\boldsymbol{x}_m\}$ to obtain good approximations to the relevant antilimits, at least in some cases.

Two different types of vector extrapolation methods exist in the literature:

1) Polynomial type methods: The *minimal polynomial extrapolation* (MPE) of Cabay and Jackson [1], the *reduced rank extrapolation* (RRE) of Kaniel and Stein [2], Eddy [3], and Mešina [4], and the *modified minimal polynomial extrapolation* (MMPE) of Brezinski [5], Pugachev [6] and Sidi, Ford, and Smith [7].

2) Epsilon algorithms: The *scalar epsilon algorithm* (SEA) of Wynn [8] (which is actually a recursive procedure for implementing the transformation of Shanks [9]), the *vector epsilon algorithm* (VEA) of Wynn [10], and the *topological epsilon algorithm* (TEA) of Brezinski [5].

The paper by Smith, Ford, and Sidi [11] gives a review of all these methods (except MMPE) that covers the developments in vector extrapolation methods until the end of the 1970s. For up-to-date reviews of MPE and RRE, see Sidi [12] and [13]. Numerically stable algorithms for implementing MPE and RRE are given in Sidi [14], these algorithms being also economical both computationally and storagewise. Jbilou and Sadok [15] have developed an analogous algorithm for MMPE along the lines suggested in Sidi, Ford, and Smith [7] and Sidi [14]. For the convergence properties and error analyses of MPE, RRE, MMPE, and TEA, as these are applied to vector sequences generated by fixed-point iterative methods from linear systems, see the works by Sidi [16]-[18], Sidi, Ford, and Smith [7], Sidi and Bridger [19], and Sidi and Shapira [20] [21]. VEA has been studied by Brezinski [22] [23], Gekeler [24], Wynn [25] [26], and Graves-Morris [27] [28].

Vector extrapolation methods are used effectively in various branches of science and engineering in accelerating the convergence of iterative methods that result from large sparse systems of equations.

All of these methods have the useful feature that their only input is the vector sequence $\{\boldsymbol{x}_m\}$ whose convergence is to be accelerated, nothing else being needed. In most applications, however, the polynomial type methods, especially MPE and RRE, have proved to be superior convergence accelerators; they require much less computation than, and half as much storage as, the epsilon algorithms for the same accuracy.

In this work, we develop yet another polynomial type method, which is based on the singular value decomposition (SVD), as well as some ideas that lead to MPE. We denote this new method by SVD-MPE. We also design a numerically stable algorithm for its implementation, whose computational cost and storage requirements are minimal. The new method is described in the next section. In Section 3, we show how the error in the approximation produced by SVD-MPE can be estimated at zero cost in terms of the quantities already used in the construction of the approximation. In Section 4, we give a very efficient algorithm for implementing SVD-MPE. In Section 5, we derive determinant representations for the approximations produced by SVD-MPE, while in Section 6, we show that this method is a Krylov subspace method when applied to vector sequences that result from the solution of linear systems via fixed-point iterative schemes. Finally, in Section 7, we illustrate its use

[1]Unless otherwise stated, $\{c_m\}$ will mean $\{c_m\}_{m=0}^{\infty}$ throughout this work.

with two numerical examples.

Before closing, we state the (*reduced* version of) the well known *singular value decomposition* (SVD) theorem. For different proofs, we refer the reader to Golub and Van Loan [29], Horn and Johnson [30], Stoer and Bulirsch [31], and Trefethen and Bau [32], for example.

Theorem 1.1 *Let* $\boldsymbol{A}\in\mathbb{C}^{r\times s}$, $r\ge s$. *Then there exist unitary matrices* $\boldsymbol{U}\in\mathbb{C}^{r\times s}$, $\boldsymbol{V}\in\mathbb{C}^{s\times s}$, *and a diagonal matrix* $\boldsymbol{\Sigma}=\operatorname{diag}(\sigma_1,\cdots,\sigma_s)\in\mathbb{R}^{s\times s}$, *with* $\sigma_1\ge\sigma_2\ge\cdots\ge\sigma_s\ge 0$, *such that*

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*.$$

Furthermore, if $\boldsymbol{U}=[\boldsymbol{u}_1\,|\cdots|\,\boldsymbol{u}_s]$ *and* $\boldsymbol{V}=[\boldsymbol{v}_1\,|\cdots|\,\boldsymbol{v}_s]$, *then*

$$\boldsymbol{A}\boldsymbol{v}_i=\sigma_i\boldsymbol{u}_i,\quad \boldsymbol{A}^*\boldsymbol{u}_i=\sigma_i\boldsymbol{v}_i,\quad i=1,\cdots,s.$$

In case $\operatorname{rank}(\boldsymbol{A})=t$, *there holds* $\sigma_i>0$, $i=1,\cdots,t$, *and the rest of the* σ_i *are zero.*

Remark: The σ_i are called the *singular values* of $\boldsymbol{A}$ and the $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ are called the corresponding *right* and *left singular vectors* of $\boldsymbol{A}$, respectively. We also have

$$\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{v}_i=\sigma_i^2\boldsymbol{v}_i,\quad \boldsymbol{A}\boldsymbol{A}^*\boldsymbol{u}_i=\sigma_i^2\boldsymbol{u}_i,\quad i=1,\cdots,s.$$

2. Development of SVD-MPE

In what follows, we use boldface lower case letters for vectors and boldface upper case letters for matrices. In addition, we will be working with general inner products $(\cdot,\cdot)$ and the l_2 norms $\|\cdot\|$ induced by them: These are defined as follows:

- In $\mathbb{C}^N$, with $\boldsymbol{M}\in\mathbb{C}^{N\times N}$ hermitian positive definite,

$$(\boldsymbol{a},\boldsymbol{b})_{\boldsymbol{M}}=\boldsymbol{a}^*\boldsymbol{M}\boldsymbol{b},\quad \|\boldsymbol{a}\|_{\boldsymbol{M}}=\sqrt{(\boldsymbol{a},\boldsymbol{a})_{\boldsymbol{M}}}. \tag{2.1}$$

- In $\mathbb{C}^{k+1}$, $k=1,2,\cdots$, with $\boldsymbol{L}_k\in\mathbb{C}^{(k+1)\times(k+1)}$ hermitian positive definite,

$$(\boldsymbol{a},\boldsymbol{b})_{\boldsymbol{L}_k}=\boldsymbol{a}^*\boldsymbol{L}_k\boldsymbol{b},\quad \|\boldsymbol{a}\|_{\boldsymbol{L}_k}=\sqrt{(\boldsymbol{a},\boldsymbol{a})_{\boldsymbol{L}_k}}. \tag{2.2}$$

Of course, the standard Euclidean inner product $\boldsymbol{a}^*\boldsymbol{b}$ and the l_2 norm $\sqrt{\boldsymbol{a}^*\boldsymbol{a}}$ induced by it are obtained by letting $\boldsymbol{M}=\boldsymbol{I}$ in (2.1) and $\boldsymbol{L}_k=\boldsymbol{I}$ in (2.2); we will denote these norms by $\|\cdot\|_2$ (we will denote by $\boldsymbol{I}$ the identity matrix in every dimension).

2.1. Summary of MPE

We begin with a brief summary of minimal polynomial extrapolation (MPE). We use the ideas that follow to develop our new method.

Given the vector sequence $\{\boldsymbol{x}_m\}$ in $\mathbb{C}^N$, we define

$$\boldsymbol{u}_m=\boldsymbol{x}_{m+1}-\boldsymbol{x}_m,\quad m=0,1,\cdots, \tag{2.3}$$

and, for some fixed *n*, define the matrices $\boldsymbol{U}_k$ via

$$\boldsymbol{U}_k=[\boldsymbol{u}_n\,|\,\boldsymbol{u}_{n+1}\,|\cdots|\,\boldsymbol{u}_{n+k}]\in\mathbb{C}^{N\times(k+1)}. \tag{2.4}$$

Clearly, there is an integer $k_0\le N$, such that the matrices $\boldsymbol{U}_k$, $k=0,1,\cdots,k_0-1$, are of full rank, but $\boldsymbol{U}_{k_0}$ is not; that is,

$$\operatorname{rank}(\boldsymbol{U}_k)=k+1,\quad k=0,1,\cdots,k_0-1;\quad \operatorname{rank}(\boldsymbol{U}_{k_0})=k_0. \tag{2.5}$$

(Of course, this is the same as saying that $\{\boldsymbol{u}_n,\boldsymbol{u}_{n+1},\cdots,\boldsymbol{u}_{n+k_0-1}\}$ is a linearly independent set, but $\{\boldsymbol{u}_n,\boldsymbol{u}_{n+1},\cdots,\boldsymbol{u}_{n+k_0}\}$ is not.) Following this, we pick a positive integer $k<k_0$ and let the vector $\boldsymbol{c}'=[c_0,c_1,\cdots,c_{k-1}]^{\mathrm{T}}\in\mathbb{C}^k$ be the solution to

$$\min_{c_0,c_1,\cdots,c_{k-1}} \left\| \sum_{i=0}^{k-1} c_i \boldsymbol{u}_{n+i} + \boldsymbol{u}_{n+k} \right\|_{\boldsymbol{M}}. \tag{2.6}$$

This minimization problem can also be expressed as in

$$\min_{\boldsymbol{c}'} \left\| \boldsymbol{U}_{k-1}\boldsymbol{c}' + \boldsymbol{u}_{n+k} \right\|_{\boldsymbol{M}}, \tag{2.7}$$

and, as is easily seen, $\boldsymbol{c}'$ is the standard least-squares solution to the linear system $\boldsymbol{U}_{k-1}\boldsymbol{c}' = -\boldsymbol{u}_{n+k}$, which, when $k < N$, is overdetermined, and generally inconsistent. With $c_0, c_1, \cdots, c_{k-1}$ determined, set $c_k = 1$, and compute the scalars $\gamma_0, \gamma_1, \cdots, \gamma_k$ via

$$\gamma_i = \frac{c_i}{\sum_{j=0}^{k} c_j}, \quad i = 0,1,\cdots,k, \tag{2.8}$$

provided $\sum_{j=0}^{k} c_j \neq 0$. Note that

$$\sum_{i=0}^{k} \gamma_i = 1. \tag{2.9}$$

Finally, set

$$\boldsymbol{s}_{n,k} = \sum_{i=0}^{k} \gamma_i \boldsymbol{x}_{n+i} \tag{2.10}$$

as the approximation to $\boldsymbol{s}$, whether $\boldsymbol{s}$ is the limit or antilimit of $\{\boldsymbol{x}_m\}$.

What we have so far is only the definition (or the theoretical development) of MPE as a method. It should not be taken as an efficient computational procedure (algorithm), however. For this topic, see [14], where numerically stable and computationally and storagewise economical algorithms for MPE and RRE are designed for the case in which $\boldsymbol{M} = \boldsymbol{I}$. A well documented FORTRAN 77 code for implementing MPE and RRE in a unified manner is also provided in [14], Appendix B.

2.2. Development of SVD-MPE

We start by observing that the unconstrained minimization problem for MPE given in (2.7) can also be expressed as a superficially "constrained" minimization problem as in

$$\min_{\boldsymbol{c}} \left\| \boldsymbol{U}_k \boldsymbol{c} \right\|_{\boldsymbol{M}}, \quad \text{subject to} \quad c_k = 1; \quad \boldsymbol{c} = [c_0, c_1, \cdots, c_k]^{\mathrm{T}}. \tag{2.11}$$

For the SVD-MPE method, we replace this "constrained" minimization problem by the following *actual* constrained minimization problem:

$$\min_{\boldsymbol{c}} \left\| \boldsymbol{U}_k \boldsymbol{c} \right\|_{\boldsymbol{M}}, \quad \text{subject to} \quad \left\| \boldsymbol{c} \right\|_{\boldsymbol{L}_k} = 1; \quad \boldsymbol{c} = [c_0, c_1, \cdots, c_k]^{\mathrm{T}}. \tag{2.12}$$

With $c_0, c_1, \cdots, c_k$ determined, we again compute $\gamma_0, \gamma_1, \cdots, \gamma_k$ via

$$\gamma_i = \frac{c_i}{\sum_{j=0}^{k} c_j}, \quad i = 0,1,\cdots,k, \tag{2.13}$$

provided $\sum_{j=0}^{k} c_j \neq 0$, noting again that

$$\sum_{i=0}^{k} \gamma_i = 1. \tag{2.14}$$

Finally, we set

$$\boldsymbol{s}_{n,k} = \sum_{i=0}^{k} \gamma_i \boldsymbol{x}_{n+i} \tag{2.15}$$

as the SVD-MPE approximation to $\boldsymbol{s}$, whether $\boldsymbol{s}$ is the limit or antilimit of $\{\boldsymbol{x}_m\}$.

Of course, the minimization problem in (2.12) has a solution for $\boldsymbol{c}=[c_0,c_1,\cdots,c_k]^{\mathrm{T}}$. Let $\sigma_{\min}=\|\boldsymbol{U}_k\boldsymbol{c}\|_{\boldsymbol{M}}$ for this (optimal) $\boldsymbol{c}$. Lemma 2.1 that follows next gives a complete characterization of $\sigma_{\min}$ and the (optimal) $\boldsymbol{c}$.

Lemma 2.1 *Let* $\sigma_{k0},\sigma_{k1},\cdots,\sigma_{kk}$ *be the singular values of the* $N\times(k+1)$ *matrix*

$$\tilde{\boldsymbol{U}}_k=\boldsymbol{M}^{1/2}\boldsymbol{U}_k\boldsymbol{L}_k^{-1/2}, \tag{2.16}$$

ordered as in

$$\sigma_{k0}\geq\sigma_{k1}\geq\cdots\geq\sigma_{kk}, \tag{2.17}$$

and let $\boldsymbol{h}_{ki}$ *be the corresponding right singular vectors of* $\tilde{\boldsymbol{U}}_k$, *that is,*

$$\tilde{\boldsymbol{U}}_k^*\tilde{\boldsymbol{U}}_k\boldsymbol{h}_{ki}=\sigma_{ki}^2\boldsymbol{h}_{ki},\quad \|\boldsymbol{h}_{ki}\|_2=1,\quad i=0,1,\cdots,k. \tag{2.18}$$

Assuming that σ_{kk}, *the smallest singular value of* $\tilde{\boldsymbol{U}}_k$, *is simple, the* (*optimal*) *solution* $\boldsymbol{c}$ *to the minimization problem in* (2.12) *is unique* (*up to a multiplicative constant* τ, $|\tau|=1$), *and is given as in*

$$\boldsymbol{c}=\boldsymbol{L}_k^{-1/2}\boldsymbol{h}_{kk};\quad \sigma_{\min}=\|\boldsymbol{U}_k\boldsymbol{c}\|_{\boldsymbol{M}}=\sigma_{kk}. \tag{2.19}$$

Proof. The proof is achieved by observing that, with $\tilde{\boldsymbol{c}}=\boldsymbol{L}_k^{1/2}\boldsymbol{c}$,

$$\|\boldsymbol{U}_k\boldsymbol{c}\|_{\boldsymbol{M}}=\|\tilde{\boldsymbol{U}}_k\tilde{\boldsymbol{c}}\|_2\quad\text{and}\quad\|\boldsymbol{c}\|_{\boldsymbol{L}_k}=\|\tilde{\boldsymbol{c}}\|_2, \tag{2.20}$$

so that the problem in (2.12) becomes

$$\min_{\tilde{\boldsymbol{c}}}\|\tilde{\boldsymbol{U}}_k\tilde{\boldsymbol{c}}\|_2,\quad\text{subject to}\quad\|\tilde{\boldsymbol{c}}\|_2=1. \tag{2.21}$$

The (optimal) solution to this problem is $\tilde{\boldsymbol{c}}=\boldsymbol{h}_{kk}$ and $\min_{\tilde{\boldsymbol{c}}}\|\tilde{\boldsymbol{U}}_k\tilde{\boldsymbol{c}}\|_2=\sigma_{kk}$. We leave the details to the reader. □

In view of the nature of the solution for the (optimal) $\boldsymbol{c}$ involving singular values and vectors, as described in Lemma 2.1, we call this new method *SVD-MPE*.

Remarks:

1) Recall that there exists a positive integer $k_0\leq N$, such that $\operatorname{rank}(\boldsymbol{U}_k)=k+1$, for $k<k_0$, but $\operatorname{rank}(\boldsymbol{U}_{k_0})=k_0$. Therefore, we have $\sigma_{kk}>0$ for all $k\leq k_0$.

2) Of course, $\boldsymbol{s}_{n,k}$ exists if and only if the (optimal) $\boldsymbol{c}=[c_0,c_1,\cdots,c_k]^{\mathrm{T}}$ satisfies $\sum_{j=0}^{k}c_j\neq 0$. In addition, by (2.13), the γ_i are unique when σ_{kk} is simple.

Before we go on to the development of our algorithm in the next section, we state the following result concerning the finite termination property of SVD-MPE, whose proof is very similar to that pertaining to MPE and RRE given in [13]:

Theorem 2.2 *Let* $\boldsymbol{s}$ *be the solution to the nonsingular linear system* $\boldsymbol{x}=\boldsymbol{T}\boldsymbol{x}+\boldsymbol{d}$, *and let* $\{\boldsymbol{x}_m\}$ *be the sequence obtained via the fixed-point iterative scheme* $\boldsymbol{x}_{m+1}=\boldsymbol{T}\boldsymbol{x}_m+\boldsymbol{d}$, $m=0,1,\cdots$, *with* $\boldsymbol{x}_0$ *chosen arbitrarily. If k is the degree of the minimal polynomial of* $\boldsymbol{T}$ *with respect to* $\boldsymbol{\epsilon}_n=\boldsymbol{x}_n-\boldsymbol{s}$ (*equivalently, with respect to* $\boldsymbol{u}_n$)[2], *then* $\boldsymbol{s}_{n,k}$ *produced by SVD-MPE satisfies* $\boldsymbol{s}_{n,k}=\boldsymbol{s}$.

3. Error Estimation

We now turn to the problem of estimating at zero cost the error $\boldsymbol{s}_{n,k}-\boldsymbol{s}$, whether $\boldsymbol{s}$ is the limit or antilimit of $\{\boldsymbol{x}_m\}$. Here we assume that $\boldsymbol{s}$ is the solution to the system of equations

[2]Given a matrix $\boldsymbol{B}\in\mathbb{C}^{r\times r}$ and a nonzero vector $\boldsymbol{a}\in\mathbb{C}^r$, the monic polynomial $P(\lambda)$ is said to be a *minimal polynomial of* $\boldsymbol{B}$ *with respect to* $\boldsymbol{a}$ if $P(\boldsymbol{B})\boldsymbol{a}=0$ and if $P(\lambda)$ has smallest degree. It is easy to show that the minimal polynomial $P(\lambda)$ of $\boldsymbol{B}$ with respect to $\boldsymbol{a}$ exists, is unique, and divides the minimal polynomial of $\boldsymbol{B}$, which in turn divides the characteristic polynomial of $\boldsymbol{B}$. [Thus, the degree of $P(\lambda)$ is at most *r*, and its zeros are some or all of the eigenvalues of $\boldsymbol{B}$.] Moreover, if $Q(\boldsymbol{B})\boldsymbol{a}=0$ for some polynomial $Q(\lambda)$ with $\deg Q>\deg P$, then $P(\lambda)$ divides $Q(\lambda)$. Concerning this subject, see Householder [33], for example.

$$\boldsymbol{x}=\boldsymbol{f}(\boldsymbol{x}); \quad \boldsymbol{f}:\mathbb{C}^N \to \mathbb{C}^N, \quad \boldsymbol{x}\in\mathbb{C}^N,$$

and that the vector sequence $\{\boldsymbol{x}_m\}$ is obtained via the fixed-point iterative scheme

$$\boldsymbol{x}_{m+1}=\boldsymbol{f}(\boldsymbol{x}_m), \quad m=0,1,\cdots,$$

$\boldsymbol{x}_0$ being the initial approximation to the solution $\boldsymbol{s}$.

Now, if $\boldsymbol{x}$ is some approximation to $\boldsymbol{s}$, then a good measure of the error $\boldsymbol{x}-\boldsymbol{s}$ in $\boldsymbol{x}$ is the *residual vector* $\boldsymbol{r}(\boldsymbol{x})$ of $\boldsymbol{x}$, namely,

$$\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{x}.$$

This is justified since $\lim_{\boldsymbol{x}\to\boldsymbol{s}}\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{r}(\boldsymbol{s})=0$. We consider two cases:

1) $\boldsymbol{f}(\boldsymbol{x})$ *is linear; that is,* $\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{T}\boldsymbol{x}+\boldsymbol{d}$, *where* $\boldsymbol{T}\in\mathbb{C}^{N\times N}$ *and* $\boldsymbol{I}-\boldsymbol{T}$ *is nonsingular.*

In this case, we have

$$\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{T}\boldsymbol{x}+\boldsymbol{d}-\boldsymbol{x}=(\boldsymbol{T}-\boldsymbol{I})(\boldsymbol{x}-\boldsymbol{s}),$$

and, therefore, by $\sum_{i=0}^{k}\gamma_i=1$, $\boldsymbol{s}_{n,k}=\sum_{i=0}^{k}\gamma_i\boldsymbol{x}_{n+i}$ satisfies

$$\boldsymbol{r}(\boldsymbol{s}_{n,k})=\sum_{i=0}^{k}\gamma_i\left[(\boldsymbol{T}\boldsymbol{x}_{n+i}+\boldsymbol{d})-\boldsymbol{x}_{n+i}\right]=\sum_{i=0}^{k}\gamma_i(\boldsymbol{x}_{n+i+1}-\boldsymbol{x}_{n+i})=\sum_{i=0}^{k}\gamma_i\boldsymbol{u}_{n+i},$$

and thus

$$\boldsymbol{r}(\boldsymbol{s}_{n,k})=\boldsymbol{U}_k\boldsymbol{\gamma}, \quad \boldsymbol{\gamma}=[\gamma_0,\gamma_1,\cdots,\gamma_k]^{\mathrm{T}}. \tag{3.1}$$

2) $\boldsymbol{f}(\boldsymbol{x})$ *is nonlinear.*

In this case, assuming that $\lim_{m\to\infty}\boldsymbol{x}_m=\boldsymbol{s}$ and expanding $\boldsymbol{f}(\boldsymbol{x}_m)$ about $\boldsymbol{s}$, we have

$$\boldsymbol{x}_{m+1}=\boldsymbol{f}(\boldsymbol{s})+\boldsymbol{F}(\boldsymbol{s})(\boldsymbol{x}_m-\boldsymbol{s})+O\left(\|\boldsymbol{x}_m-\boldsymbol{s}\|^2\right) \quad \text{as } m\to\infty,$$

where $\boldsymbol{F}(\boldsymbol{x})$ is the Jacobian matrix of the vector-valued function $\boldsymbol{f}(\boldsymbol{x})$ evaluated at $\boldsymbol{x}$. Recalling that $\boldsymbol{s}=\boldsymbol{f}(\boldsymbol{s})$, we rewrite this in the form

$$\boldsymbol{x}_{m+1}=\boldsymbol{s}+\boldsymbol{F}(\boldsymbol{s})(\boldsymbol{x}_m-\boldsymbol{s})+O\left(\|\boldsymbol{x}_m-\boldsymbol{s}\|^2\right) \quad \text{as } m\to\infty,$$

from which, we conclude that the vectors $\boldsymbol{x}_m$ and $\boldsymbol{x}_{m+1}$ satisfy the approximate equality

$$\boldsymbol{x}_{m+1}\approx\boldsymbol{s}+\boldsymbol{F}(\boldsymbol{s})(\boldsymbol{x}_m-\boldsymbol{s}) \quad \text{for all large } m.$$

That is, for all large *m*, the sequence $\{\boldsymbol{x}_m\}$ behaves as if it were being generated by an *N*-dimensional approximate linear system of the form $(\boldsymbol{I}-\boldsymbol{T})\boldsymbol{x}\approx\boldsymbol{d}$ through

$$\boldsymbol{x}_{m+1}\approx\boldsymbol{T}\boldsymbol{x}_m+\boldsymbol{d}, \quad m=0,1,\cdots,$$

where $\boldsymbol{T}=\boldsymbol{F}(\boldsymbol{s})$ and $\boldsymbol{d}=[\boldsymbol{I}-\boldsymbol{F}(\boldsymbol{s})]\boldsymbol{s}$. In view of what we already know about $\boldsymbol{r}(\boldsymbol{s}_{n,k})$ for linear systems [from (3.1)], for nonlinear systems, close to convergence, we have

$$\boldsymbol{r}(\boldsymbol{s}_{n,k})\approx\boldsymbol{U}_k\boldsymbol{\gamma}, \quad \boldsymbol{\gamma}=[\gamma_0,\gamma_1,\cdots,\gamma_k]^{\mathrm{T}}. \tag{3.2}$$

Remark: That the vector $\boldsymbol{U}_k\boldsymbol{\gamma}$ is the *exact* residual vector for $\boldsymbol{s}_{n,k}$ from linear systems and a true *approximate* residual vector for $\boldsymbol{s}_{n,k}$ from nonlinear systems was proved originally in Sidi [14]. $\boldsymbol{U}_k\boldsymbol{\gamma}$ was adopted in a subsequent paper by Jbilou and Sadok [15] and named the "generalized residual." Despite sounding interesting, this name has no meaning and is also misleading. By expanding $\boldsymbol{f}(\boldsymbol{s}_{n,k})$ about the solution $\boldsymbol{s}$ and retaining first order terms only, it can be shown that $\boldsymbol{U}_k\boldsymbol{\gamma}$ is actually a *genuine approximation* to the true residual vector $\boldsymbol{r}(\boldsymbol{s}_{n,k})$ when $\boldsymbol{f}(\boldsymbol{x})$ is nonlinear. There is nothing "generalized" about it.

Now, we can compute $\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_{\boldsymbol{M}}$ at no cost in terms of the quantities that result from our algorithm, without having to actually compute $\boldsymbol{U}_k\boldsymbol{\gamma}$ itself. Indeed, we have the following result on $\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_{\boldsymbol{M}}$, which can be incorporated in the algorithm for SVD-MPE that we discuss in the next section:

Theorem 3.1 *Let* σ_{kk} *be the smallest singular value of* $\tilde{\boldsymbol{U}}_k$ *and let* $\boldsymbol{h}_{kk}$ *be the corresponding right singular vector. Then the vector* $\boldsymbol{U}_k\boldsymbol{\gamma}$ *resulting from* $\boldsymbol{s}_{n,k}$ *satisfies*

$$\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_{\boldsymbol{M}} = \frac{\sigma_{kk}}{\left|\sum_{j=0}^{k} c_j\right|}. \tag{3.3}$$

Proof. First, the solution to (2.12) is $\boldsymbol{c} = \boldsymbol{L}^{-1/2}\boldsymbol{h}_{kk}$ by (2.19). Next, letting $\alpha = \sum_{j=0}^{k} c_j$, we have $\boldsymbol{\gamma} = \boldsymbol{c}/\alpha$ by (2.13). Consequently,

$$\boldsymbol{U}_k\boldsymbol{\gamma} = \frac{\boldsymbol{U}_k\boldsymbol{c}}{\alpha}.$$

Thus, by Lemma 2.1, we have

$$\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_{\boldsymbol{M}} = \frac{\|\boldsymbol{U}_k\boldsymbol{c}\|_{\boldsymbol{M}}}{|\alpha|} = \frac{\sigma_{kk}}{|\alpha|},$$

which is the required result. □

4. Algorithm for SVD-MPE

We now turn to the design of a good algorithm for constructing numerically the approximation $\boldsymbol{s}_{n,k}$ that results from SVD-MPE. We note that matrix computations in floating-point arithmetic must be done with care, and this is what we would like to achieve here.

In this section, we let $\boldsymbol{M} = \boldsymbol{I}$ and $\boldsymbol{L}_k = \boldsymbol{I}$ for simplicity. Thus, $\tilde{\boldsymbol{U}}_k = \boldsymbol{U}_k$. Since there is no room for confusion, we will also use σ_i, $\boldsymbol{h}_i$, and $\boldsymbol{g}_i$ to denote σ_{ki}, $\boldsymbol{h}_{ki}$, and $\boldsymbol{g}_{ki}$, respectively.

As we have seen in Section 2, to determine $\boldsymbol{s}_{n,k}$, we need $\boldsymbol{h}_k$, the right singular vector of $\boldsymbol{U}_k$ corresponding to its smallest singular value σ_k. Now, σ_k and $\boldsymbol{h}_k$ can be obtained from the singular value decomposition (SVD) of $\boldsymbol{U}_k \in \mathbb{C}^{N\times(k+1)}$. Of course, the SVD of $\boldsymbol{U}_k$ can be computed by applying directly to $\boldsymbol{U}_k$ the algorithm of Golub and Kahan [34], for example. Here we choose to apply SVD to $\boldsymbol{U}_k$ in an *indirect* way, which will result in a very efficient algorithm for SVD-MPE that is economical both computationally and storagewise in an optimal way. Here are the details of the computation of the SVD of $\boldsymbol{U}_k$, assuming that $\operatorname{rank}(\boldsymbol{U}_k) = k+1$:

1) We first compute the QR factorization of $\boldsymbol{U}_k$ in the form

$$\boldsymbol{U}_k = \boldsymbol{Q}_k\boldsymbol{R}_k;\quad \boldsymbol{Q}_k \in \mathbb{C}^{N\times(k+1)},\quad \boldsymbol{R}_k \in \mathbb{C}^{(k+1)\times(k+1)}, \tag{4.1}$$

where $\boldsymbol{Q}_k$ is unitary (that is, $\boldsymbol{Q}_k^*\boldsymbol{Q}_k = \boldsymbol{I}$) and $\boldsymbol{R}_k$ is upper triangular with positive diagonal elements, that is,

$$\boldsymbol{Q}_k = [\boldsymbol{q}_0 \,|\, \boldsymbol{q}_1 \,|\cdots|\, \boldsymbol{q}_k],\quad \boldsymbol{R}_k = \begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0k} \\ & r_{11} & \cdots & r_{1k} \\ & & \ddots & \vdots \\ & & & r_{kk} \end{bmatrix}, \tag{4.2}$$

$$\boldsymbol{q}_i^*\boldsymbol{q}_j = \delta_{ij} \quad \forall\, i,j;\quad r_{ij} = \boldsymbol{q}_i^*\boldsymbol{u}_j \quad \forall\, i \le j;\quad r_{ii} > 0 \quad \forall i. \tag{4.3}$$

Of course, we can carry out the QR factorizations in different ways. Here we do this by the *modified Gram-Schmidt process* (MGS) in a numerically stable way as follows:

```
1. Compute $r_{00} = \|\boldsymbol{u}_n\|_2$ and $\boldsymbol{q}_0 = \boldsymbol{u}_n/r_{00}$.
2. For $j = 1,\cdots,k$ do
    Set $\boldsymbol{u}_j^{(0)} = \boldsymbol{u}_{n+j}$
```

```
        For $i = 0,1,\cdots,j-1$ do
            $r_{ij} = \boldsymbol{q}_i^* \boldsymbol{u}_j^{(i)}$ and $\boldsymbol{u}_j^{(i+1)} = \boldsymbol{u}_j^{(i)} - r_{ij}\boldsymbol{q}_i$
        end do (i)
        Compute $r_{jj} = \left\|\boldsymbol{u}_j^{(j)}\right\|_2$ and $\boldsymbol{q}_j = \boldsymbol{u}_j^{(j)} / r_{jj}$ .
    end do (j)
```

Note that the matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are obtained from $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_{k-1}$, respectively, as follows:

$$\boldsymbol{Q}_k = \left[\boldsymbol{Q}_{k-1} \mid \boldsymbol{q}_k\right], \quad \boldsymbol{R}_k = \left[\begin{array}{ccc|c} & & & r_{0k} \\ & \boldsymbol{R}_{k-1} & & \vdots \\ & & & r_{k-1,k} \\ \hline 0 & \cdots & 0 & r_{kk} \end{array}\right]. \tag{4.4}$$

For MGS, see [29], for example.

2) We next compute the SVD of $\boldsymbol{R}_k$: By Theorem 1.1, there exist unitary matrices $\boldsymbol{Y}, \boldsymbol{H} \in \mathbb{C}^{(k+1)\times(k+1)}$,

$$\boldsymbol{Y} = \left[\boldsymbol{y}_0 \mid \boldsymbol{y}_1 \mid \cdots \mid \boldsymbol{y}_k\right], \quad \boldsymbol{H} = \left[\boldsymbol{h}_0 \mid \boldsymbol{h}_1 \mid \cdots \mid \boldsymbol{h}_k\right]; \quad \boldsymbol{Y}^*\boldsymbol{Y} = \boldsymbol{I}, \quad \boldsymbol{H}^*\boldsymbol{H} = \boldsymbol{I}. \tag{4.5}$$

and a square diagonal matrix $\Sigma \in \mathbb{R}^{(k+1)\times(k+1)}$,

$$\boldsymbol{\Sigma} = \operatorname{diag}\left(\sigma_0, \sigma_1, \cdots, \sigma_k\right); \quad \sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_k \geq 0, \tag{4.6}$$

such that

$$\boldsymbol{R}_k = \boldsymbol{Y}\boldsymbol{\Sigma}\boldsymbol{H}^*. \tag{4.7}$$

In addition, since $\boldsymbol{R}_k$ is nonsingular by our assumption that $\operatorname{rank}\left(\boldsymbol{U}_k\right) = k+1$, we have that $\sigma_i > 0$ for all i. Consequently, $\sigma_k > 0$.

3) Substituting (4.7) in (4.1), we obtain the following true singular value decomposition of $\boldsymbol{U}_k$:

$$\begin{aligned} \boldsymbol{U}_k = \boldsymbol{G}\boldsymbol{\Sigma}\boldsymbol{H}^*; \qquad & \boldsymbol{G} = \boldsymbol{Q}_k\boldsymbol{Y} \in \mathbb{C}^{N\times(k+1)} \quad \text{unitary}, \quad \boldsymbol{G}^*\boldsymbol{G} = \boldsymbol{I}; \\ & \boldsymbol{G} = \left[\boldsymbol{g}_0 \mid \boldsymbol{g}_1 \mid \cdots \mid \boldsymbol{g}_k\right], \quad \boldsymbol{g}_i^*\boldsymbol{g}_j = \delta_{ij}. \end{aligned} \tag{4.8}$$

Thus, σ_i, the singular values of $\boldsymbol{R}_k$, are also the singular values of $\boldsymbol{U}_k$, and $\boldsymbol{h}_i$, the corresponding right singular vectors of $\boldsymbol{R}_k$, are also the corresponding right singular vectors of $\boldsymbol{U}_k$. (Of course, the $\boldsymbol{g}_i$ are corresponding left singular vectors of $\boldsymbol{U}_k$. Note that, unlike $\boldsymbol{Y}$, $\boldsymbol{H}$, and $\boldsymbol{\Sigma}$, which we *must* compute for our algorithm, we do *not* need to actually compute $\boldsymbol{G}$ because we do not need $\boldsymbol{G}$ in our computations. The mere knowledge that the SVD of $\boldsymbol{U}_k$ is as given in (4.8) suffices to conclude that $\boldsymbol{c} = \boldsymbol{h}_k$ is the required optimal solution to (2.12); we continue with the development of our algorithm from this point.)

Remark: Computing the SVD of a matrix $\boldsymbol{A}$ by first forming its QR factorization $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, next computing the SVD of $\boldsymbol{R}$ as $\boldsymbol{R} = \boldsymbol{Y}\boldsymbol{\Sigma}\boldsymbol{H}^*$, and finally setting $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{\Sigma}\boldsymbol{H}^*$, with $\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{Y}$ was first suggested by Chan [35].

With $\boldsymbol{c} = \boldsymbol{h}_k$ already determined, we next compute the γ_i as in (2.13); that is,

$$\gamma = \frac{\boldsymbol{c}}{\sum_{j=0}^{k} c_j}, \tag{4.9}$$

provided $\sum_{j=0}^{k} c_j \neq 0$.

Next, by the fact that

$$\boldsymbol{x}_{n+i} = \boldsymbol{x}_n + \sum_{j=0}^{i-1} \boldsymbol{u}_{n+j}$$

and by (2.14), we can re-express $\boldsymbol{s}_{n,k}$ in (2.15) in the form

$$\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \sum_{j=0}^{k-1} \xi_j \boldsymbol{u}_{n+j} = \boldsymbol{x}_n + \boldsymbol{U}_{k-1}\boldsymbol{\xi}; \quad \boldsymbol{\xi} = [\xi_0, \xi_1, \cdots, \xi_{k-1}]^{\mathrm{T}}, \tag{4.10}$$

where the ξ_i are computed from the γ_j as in

$$\xi_{-1} = 1; \quad \xi_j = \sum_{i=j+1}^{k} \gamma_i = \xi_{j-1} - \gamma_j, \quad j = 0,1,\cdots,k-1. \tag{4.11}$$

Making use of the fact that $\boldsymbol{U}_{k-1} = \boldsymbol{Q}_{k-1}\boldsymbol{R}_{k-1}$, with

$$\boldsymbol{Q}_{k-1} = [\boldsymbol{q}_0 \,|\, \boldsymbol{q}_1 \,|\cdots|\, \boldsymbol{q}_{k-1}], \quad \boldsymbol{R}_{k-1} = \begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0,k-1} \\ & r_{11} & \cdots & r_{1,k-1} \\ & & \ddots & \vdots \\ & & & r_{k-1,k-1} \end{bmatrix}, \tag{4.12}$$

where the $\boldsymbol{q}_i$ and the r_{ij} are exactly those that feature in (4.2) and (4.3), we next rewrite (4.10) as in

$$\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \boldsymbol{Q}_{k-1}(\boldsymbol{R}_{k-1}\boldsymbol{\xi}). \tag{4.13}$$

Thus, the computation of $\boldsymbol{s}_{n,k}$ can be carried out economically as in

$$\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \boldsymbol{Q}_{k-1}\boldsymbol{\eta}; \quad \boldsymbol{\eta} = \boldsymbol{R}_{k-1}\boldsymbol{\xi}, \quad \boldsymbol{\eta} = [\eta_0, \eta_1, \cdots, \eta_{k-1}]^{\mathrm{T}}. \tag{4.14}$$

Of course, $\boldsymbol{Q}_{k-1}\boldsymbol{\eta}$ is best computed as a linear combination of the columns of $\boldsymbol{Q}_{k-1}$, hence (4.14) is computed as in

$$\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \sum_{i=0}^{k-1} \eta_i \boldsymbol{q}_i. \tag{4.15}$$

It is clear that, for the computation in (4.14) and (4.15), we need to save both $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$.

This completes the design of our algorithm for implementing SVD-MPE. For convenience, we provide a systematic description of this algorithm in **Table 1**, where we also include the computation of the l_2 norm of the exact or approximate residual vector $\boldsymbol{r}(\boldsymbol{s}_{n,k})$, namely, $\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_2 = \sigma_k \Big/ \left|\sum_{j=0}^{k} c_j\right|$, which is given in Theorem 3.1.

Note that the input vectors $\boldsymbol{x}_{n+i}$, $i = 1,\cdots,k+1$, need not be saved; actually, they are overwritten by $\boldsymbol{u}_{n+i}$, $i = 0,1,\cdots,k$, as the latter are being computed. As is clear from the description of MGS given above, we can overwrite the matrix $\boldsymbol{U}_k$ simultaneously with the computation of $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, the vector $\boldsymbol{q}_{n+j}$ overwriting $\boldsymbol{u}_{n+j}$ as soon as it is computed, $j = 0,1,\cdots,k$; that is, at any stage of the QR factorization, we store $k+2$ N-dimensional vectors in the memory. Since $N \gg k$ in our applications, the storage requirement of the $(k+1)\times(k+1)$ matrix $\boldsymbol{R}_k$ is negligible. So is the cost of computing the SVD of $\boldsymbol{R}_k$, and so is the cost of computing the $(k+1)$-dimensional vector $\boldsymbol{\eta}$. Thus, for all practical purposes, the computational and storage requirements of SVD-MPE are the same as those of MPE.

Remark: If we were to compute the SVD of $\boldsymbol{U}_k$, namely, $\boldsymbol{U}_k = \boldsymbol{G}\boldsymbol{\Sigma}\boldsymbol{H}^*$, directly- and not by 1) first carrying out the QR factorization of $\boldsymbol{U}_k$ as $\boldsymbol{U}_k = \boldsymbol{Q}_k\boldsymbol{R}_k$, and 2) next computing the SVD of $\boldsymbol{R}_k$ as $\boldsymbol{R}_k = \boldsymbol{Y}\boldsymbol{\Sigma}\boldsymbol{H}^*$, and 3) noting that $\boldsymbol{G} = \boldsymbol{Q}_k\boldsymbol{Y}$ without actually computing $\boldsymbol{G}$—then we would need to waste extra resources in carrying out the computation of $\boldsymbol{s}_{n,k} = \sum_{i=0}^{k} \gamma_i \boldsymbol{x}_{n+i} = \boldsymbol{x}_n + \boldsymbol{U}_{k-1}\boldsymbol{\xi}$. This direct strategy will have either of the following consequences:

1) If we have storage limitations, then we would have to overwrite $\boldsymbol{U}_k$ with $\boldsymbol{G}$. (Recall that both of these matrices are $N\times(k+1)$ and hence they are large.) As a result, we would have to compute the $\boldsymbol{x}_i$, $i = n+1,\cdots,n+k$, a second time in order to compute $\boldsymbol{s}_{n,k}$.

2) If we do not want to compute the vectors $\boldsymbol{x}_i$, $i = n+1,\cdots,n+k$, a second time, then we would have to save $\boldsymbol{U}_k$, while computing the matrix $\boldsymbol{G}$ in its singular value decomposition. Thus, we would need to save *two* $N\times(k+1)$ matrices, namely, $\boldsymbol{U}_k$ and $\boldsymbol{G}$ in the core memory simultaneously. Clearly, this limits the size of k, the order of extrapolation hence the rate of acceleration, severely.

Clearly, the indirect approach we have taken here for carrying out the singular value decomposition of $\boldsymbol{U}_k$

Table 1. Algorithm for SVD-MPE.

```
Step 0. Input: The vectors $\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+k+1}$.

Step 1. Compute $\boldsymbol{u}_i = \Delta\boldsymbol{x}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i,\ i = n, n+1, \ldots, n+k$.
        Set $\boldsymbol{U}_k = [\,\boldsymbol{u}_n \,|\, \boldsymbol{u}_{n+1} \,|\, \cdots \,|\, \boldsymbol{u}_{n+k}\,] \in \mathbb{C}^{N\times(k+1)}$.
        Compute the QR-factorization of $\boldsymbol{U}_k$, namely, $\boldsymbol{U}_k = \boldsymbol{Q}_k\boldsymbol{R}_k$,
        $\boldsymbol{Q}_k \in \mathbb{C}^{N\times(k+1)},\ \boldsymbol{R}_k \in \mathbb{C}^{(k+1)\times(k+1)}$,
        $\boldsymbol{Q}_k$ unitary, $\boldsymbol{R}_k$ upper triangular with positive diagonal:
        $\boldsymbol{Q}_k = [\,\boldsymbol{q}_0 \,|\, \boldsymbol{q}_1 \,|\, \cdots \,|\, \boldsymbol{q}_k\,],\quad \boldsymbol{R}_k = \begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0,k} \\ & r_{11} & \cdots & r_{1,k} \\ & & \ddots & \vdots \\ & & & r_{k,k} \end{bmatrix}$,
        $\boldsymbol{q}_i^*\boldsymbol{q}_j = \delta_{ij}\ \ \forall\, i,j;\quad r_{ij} = \boldsymbol{q}_i^*\boldsymbol{u}_j\ \ \forall\, i,j,\quad r_{ii} > 0\ \ \forall\, i$.

Step 2. Compute the SVD of $\boldsymbol{R}_k$, namely, $\boldsymbol{R}_k = \boldsymbol{Y}\boldsymbol{\Sigma}\boldsymbol{H}^*$,
        $\boldsymbol{Y}, \boldsymbol{H} \in \mathbb{C}^{(k+1)\times(k+1)}$ unitary, $\boldsymbol{\Sigma} \in \mathbb{R}^{(k+1)\times(k+1)}$ diagonal:
        $\boldsymbol{Y} = [\,\boldsymbol{y}_0 \,|\, \boldsymbol{y}_1 \,|\, \cdots \,|\, \boldsymbol{y}_k\,],\ \boldsymbol{H} = [\,\boldsymbol{h}_0 \,|\, \boldsymbol{h}_1 \,|\, \cdots \,|\, \boldsymbol{h}_k\,],\ \boldsymbol{Y}^*\boldsymbol{Y} = \boldsymbol{H}^*\boldsymbol{H} = \boldsymbol{I}$,
        $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_0, \sigma_1, \ldots, \sigma_k),\quad \sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_k \geq 0$.

Step 3. Set $\boldsymbol{c} = [c_0, c_1, \ldots, c_k]^{\mathrm{T}} = \boldsymbol{h}_k$, and compute $\alpha = \sum_{j=0}^{k} c_j$.
        Set $\gamma_i = c_i/\alpha,\quad i = 0, 1, \ldots, k$, provided $\alpha \neq 0$.
        Compute $\boldsymbol{\xi} = [\xi_0, \xi_1, \ldots, \xi_{k-1}]^{\mathrm{T}}$ via
        $\xi_0 = 1 - \gamma_0;\quad \xi_j = \xi_{j-1} - \gamma_j,\quad j = 1, \ldots, k-1$.

Step 4. Compute $\boldsymbol{s}_{n,k}$ via $\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \boldsymbol{Q}_{k-1}\boldsymbol{R}_{k-1}\boldsymbol{\xi}$ as follows:
        Compute $\boldsymbol{\eta} = \boldsymbol{R}_{k-1}\boldsymbol{\xi};\ \ \boldsymbol{\eta} = [\eta_0, \eta_1, \ldots, \eta_{k-1}]^{\mathrm{T}}$.
        Compute $\boldsymbol{s}_{n,k} = \boldsymbol{x}_n + \boldsymbol{Q}_{k-1}\boldsymbol{\eta} = \boldsymbol{x}_n + \sum_{i=0}^{k-1} \eta_i\boldsymbol{q}_i$.
        Compute $\|\boldsymbol{U}_k\boldsymbol{\gamma}\|_2 = \sigma_k/|\alpha|$, the exact or approximate $\|\boldsymbol{r}(\boldsymbol{s}_{n,k})\|_2$.
```

enables us to save extra computing and storage very conveniently when N is very large.

5. Determinant Representations for SVD-MPE

In [7] and [16], determinant representations were derived for the vectors $\boldsymbol{s}_{n,k}$ that are produced by the vector extrapolation methods MPE, RRE, MMPE, and TEA. These representations have turned out to be very useful in the analysis of the algebraic and analytic properties of these methods. In particular, they were used for obtaining interesting recursion relations among the $\boldsymbol{s}_{n,k}$ and in proving sharp convergence and stability theorems for them. We now derive two analogous determinant representations for $\boldsymbol{s}_{n,k}$ produced by SVD-MPE.

The following lemma, whose proof can be found in [7], Section 3, will be used in this derivation in Theorem 5.2.

Lemma 5.1 *Let $u_{i,j}$ and γ_j be scalars and let the γ_j satisfy the linear system*

$$\begin{aligned} &\sum_{j=0}^{k} u_{i,j}\gamma_j = 0, \quad i = 0, 1, \cdots, k-1, \\ &\sum_{j=0}^{k} \gamma_j = 1. \end{aligned} \tag{5.1}$$

Then, whether v_j are scalars or vectors, there holds

$$\sum_{j=0}^{k} \gamma_j v_j = \frac{D(v_0, v_1, \cdots, v_k)}{D(1, 1, \cdots, 1)}, \tag{5.2}$$

where

$$D(v_0, v_1, \cdots, v_k) = \begin{vmatrix} v_0 & v_1 & \cdots & v_k \\ u_{0,0} & u_{0,1} & \cdots & u_{0,k} \\ u_{1,0} & u_{1,1} & \cdots & u_{1,k} \\ \vdots & \vdots & & \vdots \\ u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k} \end{vmatrix}, \tag{5.3}$$

provided $D(1,1,\cdots,1)\neq 0$. *In case the* v_i *are vectors, the determinant* $D(v_0,v_1,\cdots,v_k)$ *is defined via its expansion with respect to its first row.*

For convenience of notation, we will write

$$\boldsymbol{L}_k=\begin{bmatrix} l_{00} & l_{01} & \cdots & l_{0k} \\ l_{10} & l_{11} & \cdots & l_{1k} \\ \vdots & \vdots & & \vdots \\ l_{k0} & l_{k1} & \cdots & l_{kk}\end{bmatrix},\quad (\boldsymbol{L}_k)_{ij}=l_{ij},\quad i,j=0,1,\cdots.$$

Then $\boldsymbol{L}_{k-1}$ is the principal submatrix of $\boldsymbol{L}_k$ obtained by deleting the last row and the last column of $\boldsymbol{L}_k$. In addition, $\boldsymbol{L}_{k-1}$ is hermitian positive definite just like $\boldsymbol{L}_k$.

Theorem 5.2 that follows gives our first determinant representation for $\boldsymbol{s}_{n,k}$ resulting from SVD-MPE and is based only on the smallest singular value σ_{kk} of $\tilde{\boldsymbol{U}}_k$ and the corresponding right singular vector $\boldsymbol{h}_{kk}$.

Theorem 5.2 *Define*

$$u_{i,j}=(\boldsymbol{u}_{n+i},\boldsymbol{u}_{n+j})_{\boldsymbol{M}}-\sigma_{kk}^2(\boldsymbol{L}_k)_{ij},\quad i,j=0,1,\cdots,k, \tag{5.4}$$

and assume that

$$\sigma_{kk}<\sigma_{k-1,k-1}.^{3} \tag{5.5}$$

Then, provided $D(1,1,\cdots,1)\neq 0$, $\boldsymbol{s}_{n,k}$ *exists and has the determinant representation*

$$\boldsymbol{s}_{n,k}=\frac{D(\boldsymbol{x}_n,\boldsymbol{x}_{n+1},\cdots,\boldsymbol{x}_{n+k})}{D(1,1,\cdots,1)}, \tag{5.6}$$

where $D(v_0,v_1,\cdots,v_k)$ *is the* $(k+1)\times(k+1)$ *determinant defined as in* (5.3) *in Lemma* 5.1 *with the* $u_{i,j}$ *as in* (5.4).

Proof. With $\tilde{\boldsymbol{U}}_k$ as in (2.16), we start by rewriting (2.18) in the form

$$(\tilde{\boldsymbol{U}}_k^*\tilde{\boldsymbol{U}}_k-\sigma_{kk}^2\boldsymbol{I})\boldsymbol{h}_{kk}=0. \tag{5.7}$$

Invoking here $\boldsymbol{h}_{kk}=\boldsymbol{L}_k^{1/2}\boldsymbol{c}$, which follows from (2.19), and multiplying the resulting equality on the left by $\boldsymbol{L}_k^{1/2}$, we obtain

$$(\boldsymbol{U}_k^*\boldsymbol{M}\boldsymbol{U}_k-\sigma_{kk}^2\boldsymbol{L}_k)\boldsymbol{c}=0. \tag{5.8}$$

Dividing both sides of this equality by $\sum_{j=0}^{k}c_j$, and invoking (2.13), we have

$$(\boldsymbol{U}_k^*\boldsymbol{M}\boldsymbol{U}_k-\sigma_{kk}^2\boldsymbol{L}_k)\boldsymbol{\gamma}=0, \tag{5.9}$$

which, by the fact that

$$(\boldsymbol{U}_k^*\boldsymbol{M}\boldsymbol{U}_k)_{ij}=\boldsymbol{u}_{n+i}^*\boldsymbol{M}\boldsymbol{u}_{n+j}=(\boldsymbol{u}_{n+i},\boldsymbol{u}_{n+j})_{\boldsymbol{M}},$$

is the same as

$$\sum_{j=0}^{k}\left[(\boldsymbol{u}_{n+i},\boldsymbol{u}_{n+j})_{\boldsymbol{M}}-\sigma_{kk}^2(\boldsymbol{L}_k)_{ij}\right]\gamma_j=0\Rightarrow\sum_{j=0}^{k}u_{i,j}\gamma_j=0,\quad i=0,1,\cdots,k, \tag{5.10}$$

where we have invoked (5.4). We will be able to apply Lemma 5.1 to prove the validity of (5.6) if we show that, in (5.10), the equations with $i=0,1,\cdots,k-1$, are linearly independent, or, equivalently, the first k rows of the matrix

$$\boldsymbol{B}_k=\boldsymbol{U}_k^*\boldsymbol{M}\boldsymbol{U}_k-\sigma_{kk}^2\boldsymbol{L}_k$$

[3]From the Cauchy interlace theorem, we already know that $\sigma_{kk}\le\sigma_{k-1,k-1}$. See [29], for example.

are linearly independent. By the fact that

$$\boldsymbol{U}_k = \left[\boldsymbol{U}_{k-1} \,|\, \boldsymbol{u}_{n+k}\right] \quad \text{and} \quad \boldsymbol{L}_k = \left[\begin{array}{c|c} \boldsymbol{L}_{k-1} & \boldsymbol{l}_k \\ \hline \boldsymbol{l}_k^* & l_{kk} \end{array}\right], \quad \boldsymbol{l}_k = \left[l_{0k}, l_{1k}, \cdots, l_{k-1,k}\right]^{\mathrm{T}},$$

we have

$$\boldsymbol{B}_k = \left[\begin{array}{c|c} \boldsymbol{B}'_{k-1} & \boldsymbol{p} \\ \hline \boldsymbol{p}^* & \beta \end{array}\right],$$

where

$$\boldsymbol{B}'_{k-1} = \boldsymbol{U}_{k-1}^* \boldsymbol{M} \boldsymbol{U}_{k-1} - \sigma_{kk}^2 \boldsymbol{L}_{k-1},$$

and

$$\boldsymbol{p} = \boldsymbol{U}_{k-1}^* \boldsymbol{M} \boldsymbol{u}_{n+k} - \sigma_{kk}^2 \boldsymbol{l}_k, \quad \beta = \boldsymbol{u}_{n+k}^* \boldsymbol{M} \boldsymbol{u}_{n+k} - \sigma_{kk}^2 l_{kk}.$$

Invoking $\boldsymbol{U}_{k-1} = \boldsymbol{M}^{-1/2} \tilde{\boldsymbol{U}}_{k-1} \boldsymbol{L}_{k-1}^{1/2}$, we obtain

$$\boldsymbol{B}'_{k-1} = \boldsymbol{L}_{k-1}^{1/2} \left(\tilde{\boldsymbol{U}}_{k-1}^* \tilde{\boldsymbol{U}}_{k-1} - \sigma_{kk}^2 \boldsymbol{I}\right) \boldsymbol{L}_{k-1}^{1/2}.$$

Since $\sigma_{k-1,k-1}^2$ is the smallest eigenvalue of $\tilde{\boldsymbol{U}}_{k-1}^* \tilde{\boldsymbol{U}}_{k-1}$ and since $\sigma_{k-1,k-1} > \sigma_{kk}$, it turns out that $\boldsymbol{B}'_{k-1}$ is positive definite, which guarantees that the first k rows of $\boldsymbol{B}_k$ are linearly independent. This completes the proof. □

Remark: We note that the condition that $D(1,1,\cdots,1) \neq 0$ in Theorem 5.2 is equivalent to the condition that $\sum_{j=0}^{k} c_j \neq 0$, which we have already met in Section 2.

The determinant representation given in Theorem 5.3 that follows is based on the complete singular value decomposition of $\tilde{\boldsymbol{U}}_k$, hence is different from that given in Theorem 5.2. Since there is no room for confusion, we will denote the singular values σ_{ki} and right and left singular vectors $\boldsymbol{h}_{ki}$ and $\boldsymbol{g}_{ki}$ of $\tilde{\boldsymbol{U}}_k$ by σ_i, $\boldsymbol{h}_i$ and $\boldsymbol{g}_i$, respectively.

Theorem 5.3 *Let* $\tilde{\boldsymbol{U}}_k$ *be as in* (2.16), *and let*

$$\tilde{\boldsymbol{U}}_k = \boldsymbol{G \Sigma H}^*, \quad \boldsymbol{G} \in \mathbb{C}^{N \times (k+1)}, \quad \boldsymbol{H} \in \mathbb{C}^{(k+1)\times(k+1)}, \quad \Sigma \in \mathbb{C}^{(k+1)\times(k+1)}$$

be the singular value decomposition of $\tilde{\boldsymbol{U}}_k$; *that is,*

$$\boldsymbol{G} = \left[\boldsymbol{g}_0 \,|\, \boldsymbol{g}_1 \,|\cdots|\, \boldsymbol{g}_k\right], \quad \boldsymbol{g}_i^* \boldsymbol{g}_j = \delta_{ij}; \quad \boldsymbol{H} = \left[\boldsymbol{h}_0 \,|\, \boldsymbol{h}_1 \,|\cdots|\, \boldsymbol{h}_k\right], \quad \boldsymbol{h}_i^* \boldsymbol{h}_j = \delta_{ij},$$

and

$$\Sigma = \operatorname{diag}(\sigma_0, \sigma_1, \cdots, \sigma_k), \quad \sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_k.$$

Define

$$u_{i,j} = \left(\boldsymbol{M}^{1/2} \boldsymbol{g}_i\right)^* \boldsymbol{u}_{n+j} = \boldsymbol{g}_i^* \boldsymbol{M}^{1/2} \boldsymbol{u}_{n+j}, \quad i = 0,1,\cdots,k-1, \quad j = 0,1,\cdots,k, \tag{5.11}$$

Then, $\boldsymbol{s}_{n,k}$ *has the determinant representation*

$$\boldsymbol{s}_{n,k} = \frac{D(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_{n+k})}{D(1,1,\cdots,1)}, \tag{5.12}$$

where $D(v_0, v_1, \cdots, v_k)$ *is the* $(k+1)\times(k+1)$ *determinant defined as in* (5.3) *in Lemma* 5.1 *with the* $u_{i,j}$ *as in* (5.11).

Proof. By Theorem 1.1,

$$\tilde{\boldsymbol{U}}_k \boldsymbol{h}_k = \sigma_k \boldsymbol{g}_k \quad \text{and} \quad \boldsymbol{g}_i^* \boldsymbol{g}_k = 0, \quad i = 0,1,\cdots,k-1. \tag{5.13}$$

Therefore,

$$\boldsymbol{g}_i^* \tilde{\boldsymbol{U}}_k \boldsymbol{h}_k = 0, \quad i = 0,1,\cdots,k-1. \tag{5.14}$$

By (2.16) and by the fact that $\boldsymbol{c} = \boldsymbol{L}_k^{-1/2}\boldsymbol{h}_k$, which follows from (2.19), and by the fact that $\boldsymbol{\gamma} = \boldsymbol{c}/\alpha$, $\alpha = \sum_{j=0}^{k} c_j$, which follows from (2.13), and by (5.14), we then have

$$\boldsymbol{g}_i^* \boldsymbol{M}^{1/2} \boldsymbol{U}_k \boldsymbol{\gamma} = \alpha^{-1}\left(\boldsymbol{g}_i^* \boldsymbol{M}^{1/2} \boldsymbol{U}_k \boldsymbol{c}\right) = \alpha^{-1}\left(\boldsymbol{g}_i^* \tilde{\boldsymbol{U}}_k \boldsymbol{h}_k\right) = 0, \quad i = 0,1,\cdots,k-1. \tag{5.15}$$

But, by (5.11), (5.15) is the same as

$$\sum_{j=0}^{k} u_{i,j}\gamma_j = 0, \quad i = 0,1,\cdots,k-1.$$

Therefore, Lemma 5.1 applies with $u_{i,j}$ as in (5.11), and the result follows. □

6. SVD-MPE as a Krylov Subspace Method

In Sidi [17], we discussed the connection of the extrapolation methods MPE, RRE, and TEA with Krylov subspace methods for linear systems. We now want to extend the treatment of [17] to SVD-MPE. Here we recall that a Krylov subspace method is also a projection method and that a projection method is defined uniquely by its right and left subspaces[4]. In the next theorem, we show that SVD-MPE is a bona fide Krylov subspace method and we identify its right and left subspaces.

Since there is no room for confusion, we will use the notation of Theorem 5.3.

Theorem 6.1 *Let* $\boldsymbol{s}$ *be the unique solution to the linear system* $\boldsymbol{Cx} = \boldsymbol{d}$, *which we express in the form*

$$(\boldsymbol{I} - \boldsymbol{T})\boldsymbol{x} = \boldsymbol{d} \Rightarrow \boldsymbol{x} = \boldsymbol{Tx} + \boldsymbol{d}; \quad \boldsymbol{T} = \boldsymbol{I} - \boldsymbol{C},$$

and let the vector sequence $\{\boldsymbol{x}_m\}$ *be produced by the fixed-point iterative scheme*

$$\boldsymbol{x}_{m+1} = \boldsymbol{T}\boldsymbol{x}_m + \boldsymbol{d}, \quad m = 0,1,\cdots.$$

Define the residual vector of $\boldsymbol{x}$ *via* $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{d} - \boldsymbol{Cx}$. *Let also* $\boldsymbol{s}_k \equiv \boldsymbol{s}_{0,k}$ *be the approximation to* $\boldsymbol{s}$ *produced by SVD-MPE. Then the following are true*:

1) $\boldsymbol{s}_k$ *is of the form*

$$\boldsymbol{s}_k = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \delta_i \left(\boldsymbol{C}^i \boldsymbol{r}_0\right) \quad \text{for some } \delta_i; \quad \boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{x}_0) = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}_0. \tag{6.1}$$

2) *The residual vector of* $\boldsymbol{s}_k$, *namely*, $\boldsymbol{r}(\boldsymbol{s}_k)$, *is orthogonal to the subspace*

$$L_k = \text{span}\left\{\boldsymbol{M}^{1/2}\boldsymbol{g}_0, \boldsymbol{M}^{1/2}\boldsymbol{g}_1, \cdots, \boldsymbol{M}^{1/2}\boldsymbol{g}_{k-1}\right\}.$$

Thus,

$$\left(\boldsymbol{M}^{1/2}\boldsymbol{g}_i\right)^* \boldsymbol{r}(\boldsymbol{s}_k) = 0, \quad i = 0,1,\cdots,k-1. \tag{6.2}$$

Consequently, SVD-MPE is a Krylov subspace method for the linear system $\boldsymbol{Cx} = \boldsymbol{d}$, *with the Krylov subspace* $K_k(\boldsymbol{C}; \boldsymbol{r}_0) = \text{span}\left\{\boldsymbol{r}_0, \boldsymbol{C}\boldsymbol{r}_0, \cdots, \boldsymbol{C}^{k-1}\boldsymbol{r}_0\right\}$ *as its right subspace and* $L_k = \text{span}\left\{\boldsymbol{M}^{1/2}\boldsymbol{g}_0, \boldsymbol{M}^{1/2}\boldsymbol{g}_1, \cdots, \boldsymbol{M}^{1/2}\boldsymbol{g}_{k-1}\right\}$ *as its left subspace.*

Proof. With the $\boldsymbol{x}_m$ generated as above, we have

$$\boldsymbol{u}_{m+1} = \boldsymbol{T}\boldsymbol{u}_m \Rightarrow \boldsymbol{u}_m = \boldsymbol{T}^m \boldsymbol{u}_0, \quad m = 0,1,\cdots.$$

Therefore,

$$\boldsymbol{u}_0 = \boldsymbol{x}_1 - \boldsymbol{x}_0 = \boldsymbol{T}\boldsymbol{x}_0 + \boldsymbol{d} - \boldsymbol{x}_0 = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}_0 = \boldsymbol{r}(\boldsymbol{x}_0)$$

[4]A projection method for the solution of the linear system $\boldsymbol{Cx} = \boldsymbol{d}$, where $\boldsymbol{C} \in \mathbb{C}^{N\times N}$, is defined as follows: Let Y and Z be k-dimensional subspaces of $\mathbb{C}^N$ and let $\boldsymbol{x}_0$ be a given vector in $\mathbb{C}^N$. Then the projection method produces an approximation $\boldsymbol{s}_k$ to the solution of $\boldsymbol{Cx} = \boldsymbol{d}$ as follows: 1) $\boldsymbol{s}_k = \boldsymbol{x}_0 + \boldsymbol{y}$, $\boldsymbol{y} \in Y$, 2) $\boldsymbol{h}^*\boldsymbol{r}(\boldsymbol{s}_k) = 0$ for every $\boldsymbol{h} \in Z$. Y and Z are called, respectively, the right and left subspaces of the method. If Y is the Krylov subspace $K_k(\boldsymbol{C}; \boldsymbol{r}_0) = \text{span}\left\{\boldsymbol{r}_0, \boldsymbol{C}\boldsymbol{r}_0, \cdots, \boldsymbol{C}^{k-1}\boldsymbol{r}_0\right\}$, where $\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}_0$, then the projection method is called a *Krylov subspace method.*

and

$$s_k = x_0 + \sum_{i=0}^{k-1} \xi_i u_i = x_0 + \sum_{i=0}^{k-1} \xi_i T^i u_0 = x_0 + \sum_{i=0}^{k-1} \xi_i T^i r_0.$$

Upon substituting $T = I - C$ in this equality, we obtain (6.1).

To prove (6.2), we first recall that $U_k \gamma = r(s_k)$ by (3.1). By this and by (5.15), the result in (6.2) follows. □

Remark: We recall that (see [17]), when applied to linearly generated sequences $\{x_m\}$ as in Theorem 6.1, MPE, RRE, and TEA are mathematically equivalent to, respectively, the full orthogonalization method (FOM) of Arnoldi [36], the generalized minimum residual method (GMR), the best implementation of it being GMRES by Saad and Schultz [37], and the method of Lanczos [38]. In all these methods the right subspace is the k-dimensional Krylov subspace $K_k(C; r_0)$. The left subspaces are also Krylov subspaces, with 1) $K_k(C; r_0)$ for FOM, 2) $K_k(C; Cr_0)$ for GMR, and 3) $K_k(C^*; r_0)$ for the method of Lanczos. One important point about the left subspaces of these three methods is that they expand as k increases, that is, the left subspace of dimension $k+1$ contains the left subspace of dimension k. As for SVD-MPE, its right subspace is also the k-dimensional Krylov subspace $K_k(C; r_0)$, which makes SVD-MPE a bona fide Krylov subspace method, and the left subspace is $L_k = \text{span}\{M^{1/2} g_{k0}, M^{1/2} g_{k1}, \cdots, M^{1/2} g_{k,k-1}\}$. Thus, the k-dimensional left subspace is not a Krylov subspace, since it is not contained in the left subspace of dimension $k+1$, as the left singular vectors g_{ki} of $\tilde{U}_k$ are different from the left singular vectors $g_{k+1,i}$ of $\tilde{U}_{k+1}$.

7. Numerical Examples

We now provide two examples that show the performance of SVD-MPE and compare SVD-MPE with MPE. In both examples, SVD-MPE and MPE are implemented with the standard Euclidean inner product and the norm induced by it. Thus, $M = I$ and $L_k = I$ throughout.

As we have already mentioned, a major application area of vector extrapolation methods is that of numerical solution of large systems of linear or nonlinear equations $\psi(x) = 0$ by fixed-point iterations $x_{m+1} = f(x_m)$. [Here $x = f(x)$ is a possibly preconditioned form of $\psi(x) = 0$.] For SVD-MPE, as well as all other polynomial methods discussed in the literature, the computation of the approximation $s_{n,k}$ to s, the solution of $\psi(x) = 0$, requires $k+1$ of the vectors x_m to be stored in the computer memory. For systems of very large dimension N, this means that we should keep k at a moderate size. In view of this limitation, a practical strategy for systems of equations is *cycling*, for which n and k are fixed. Here are the steps of cycling:

C0) Choose integers $n \geq 0$, and $k \geq 1$, and an initial vector x_0.

C1) Compute the vectors $x_1, x_2, \cdots, x_{n+k+1}$ [via $x_{m+1} = f(x_m)$].

C2) Apply SVD-MPE (or MPE) to the vectors $x_n, x_{n+1}, \cdots, x_{n+k+1}$, with end result $s_{n,k}$.

C3) If $s_{n,k}$ satisfies accuracy test, stop.

Otherwise, set $x_0 = s_{n,k}$, and go to Step C1.

We will call each application of steps C1 - C3 a *cycle*, and denote by $s_{n,k}^{(r)}$ the $s_{n,k}$ that is computed in the rth cycle. We will also denote the initial vector x_0 in step C0 by $s_{n,k}^{(0)}$. Numerical examples suggest that the sequence $\{s_{n,k}^{(r)}\}_{r=0}^{\infty}$ has very good convergence properties. (A detailed study of errors and convergence properties for MPE and RRE in the cycling mode is given in [20] and [21].)

Note that the remark at the end of Section 4 is relevant to the implementation of SVD-MPE in the cycling mode when N, the dimension of the x_m, is very large and storage is limited and, therefore, the size of k is limited as well.

Example 7.1 Consider the vector sequence $\{x_m\}$ obtained from $x_{m+1} = Tx_m + d$, $m = 0, 1, \cdots$, where

$$T = 0.06 \times \begin{bmatrix} 5 & 2 & 1 & 1 & & & & \\ 2 & 6 & 3 & 1 & 1 & & & \\ 1 & 3 & 6 & 3 & 1 & 1 & & \\ 1 & 1 & 3 & 6 & 3 & 1 & 1 & \\ & 1 & 1 & 3 & 6 & 3 & 1 & 1 \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

and is symmetric with respect to both main diagonals, and $T \in \mathbb{R}^{N\times N}$ and is hermitian. Therefore, T is diagonalizable with real eigenvalues. The vector d is such that the exact solution to $x = Tx + d$ is $s = [1,1,\cdots,1]^{\mathrm{T}}$. We have $\rho(T) < 1$, so that $\{x_m\}$ converges to s.

Figure 1 shows the l_2 norms of the errors in $s_{n,k}$, $n = 0,1,\cdots$, with $k = 5$ fixed. Here $N = 100$. Note that all of the approximations $s_{n,5}$ make use of the same (infinite) vector sequence $\{x_m\}$, and, practically speaking, we are looking at how the methods behave as $n \to \infty$. It is interesting to see that SVD-MPE and MPE behave almost the same. Although we have a rigorous asymptotic theory confirming the behavior of MPE in this mode as observed in Figure 1 (see [16] [18] and [19]), we do not have any such theory for SVD-MPE at the present[5].

Figure 2 shows the l_2 norm of the error in $s_{n,k}$ in the cycling mode with $n = 0$ and $k = 20$. Now $N = 1000$, a relatively large dimension.

Example 7.2 We now apply SVD-MPE and MPE to the nonlinear system that arises from finite-difference approximation of the two-dimensional convection-diffusion equation considered in Kelley ([39], pp. 108-109), namely,

$$-\nabla^2 u + Cu\left(u_x + u_y\right) = f, \quad (x,y) \in \Omega = (0,1)\times(0,1),$$

where $u(x,y)$ satisfies homogeneous boundary conditions. $f(x,y)$ is constructed by setting $C = 20$ in the differential equation and by taking

$$u(x,y) = 10xy(1-x)(1-y)\exp\left(x^{4.5}\right)$$

as the exact solution.

The equation is discretized on a square grid by approximating u_{xx}, u_{yy}, u_x, and u_y by centered differences with truncation errors $O\left(h^2\right)$. Thus, letting $h = 1/\nu$, and

$$\left(x_i, y_j\right) = (ih, jh), \quad i,j = 0,1,\cdots,\nu,$$

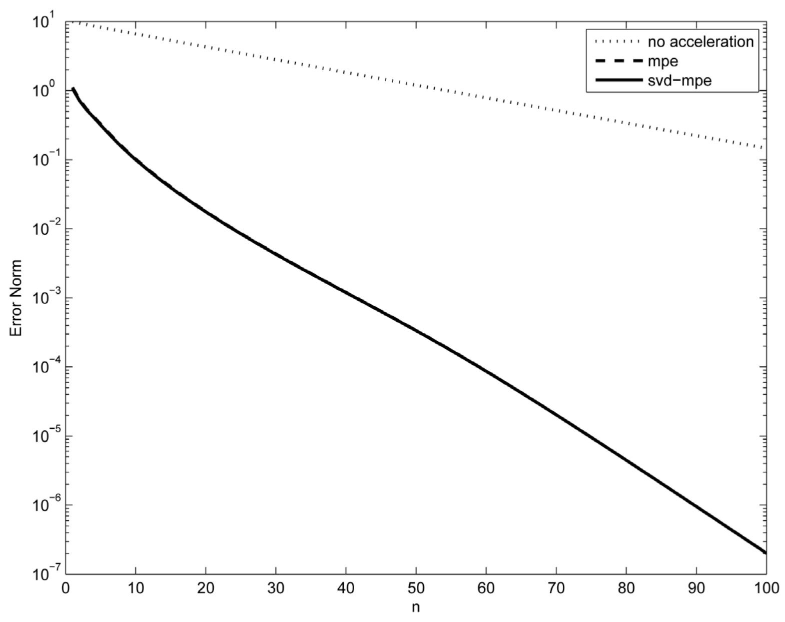

Figure 1. l_2 norm of error in $s_{n,k}$, $n = 0,1,\cdots$, with $k = 5$, from MPE and SVD-MPE, for Example 7.1 with $N = 100$.

[5]As proved in [7] and [16], if we order the distinct eigenvalues λ_i of T as $|\lambda_1| \ge |\lambda_2| \ge \cdots$, then with fixed k, we have $s_{n,k} - s = O\left(\lambda_{k+1}^n\right)$ as $n \to \infty$ for MPE, RRE, MMPE, and TEA, if T is diagonalizable and $|\lambda_k| > |\lambda_{k+1}|$. (This result explains the straight line behavior of MPE in **Figure 1**.) Generalizations and extensions of this result are given in the papers [18]-[21], for the cases in which T is not necessarily diagonalizable and/or $|\lambda_k| = |\lambda_{k+1}|$.

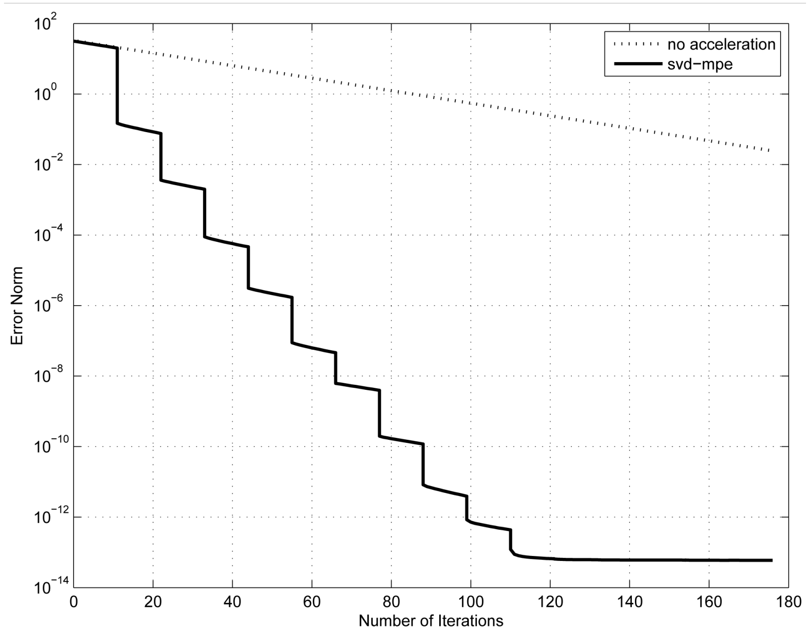

Figure 2. l_2 norm of error in $\boldsymbol{s}_{0,20}$ in the cycling mode, from SVD-MPE, for Example 7.1 with $N = 1000$.

and

$$u_x(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2h}, \quad u_y(x_i, y_j) \approx \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2h},$$

and

$$-\nabla^2 u(x_i, y_j) \approx \frac{4u(x_i, y_j) - u(x_{i+1}, y_j) - u(x_{i-1}, y_j) - u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{h^2},$$

we replace the differential equation by the finite difference equations

$$\begin{aligned} &\frac{4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}}{h^2} \\ &+ C u_{ij} \frac{u_{i+1,j} - u_{i-1,j} + u_{i,j+1} - u_{i,j-1}}{2h} = f(x_i, y_j), \quad 1 \le i, j \le \nu - 1, \end{aligned}$$

with

$$u_{0,j} = u_{i,0} = u_{\nu,j} = u_{i,\nu} = 0 \quad \forall\, i, j.$$

Here u_{ij} is the approximation to $u(x_i, y_j)$, as usual.

We first write the finite difference equations in a way that is analogous to the PDE written in the form

$$-\nabla^2 u = f - Cu(u_x + u_y),$$

and split the matrix representing $-\nabla^2$ to enable the use of the Jacobi and Gauss-Seidel methods as the iterative procedures to generate the sequences $\{\boldsymbol{x}_m\}$.

Figure 3 and **Figure 4** show the l_2 norms of the errors in $\boldsymbol{s}_{n,k}$ from SVD-MPE and MPE in the cycling mode with $n = 0$ and $k = 20$, the iterative procedures being, respectively, that of Jacobi and that of Gauss-Seidel for the linear part $-\nabla^2 u$ of the PDE. Here $\nu = 100$, so that the number of unknowns (the dimension) is $N = 99^2 = 9801$. Note that the convergence of cycling is much faster with Gauss-Seidel iteration than with

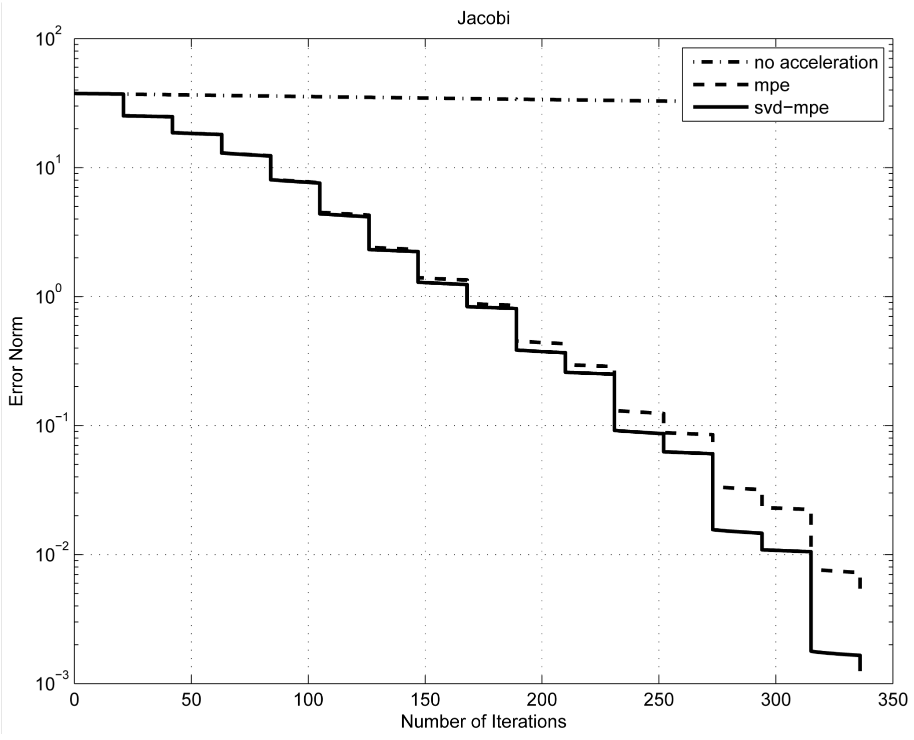

Figure 3. l_2 norm of error in $\boldsymbol{s}_{0,20}$ in the cycling mode, from MPE and SVD-MPE, for Example 7.2 with $\nu = 100$ hence $N = 99^2$. The underlying iteration method is that of Jacobi.

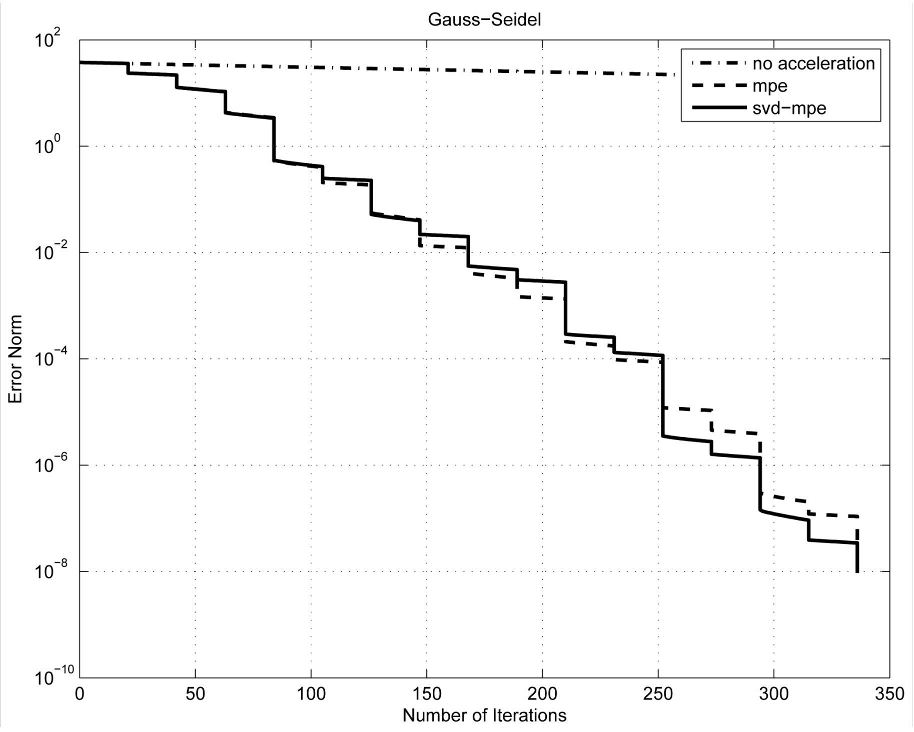

Figure 4. l_2 norm of error in $\boldsymbol{s}_{0,20}$ in the cycling mode, from MPE and SVD-MPE, for Example 7.2 with $\nu = 100$ hence $N = 99^2$. The underlying iteration method is that of Gauss-Seidel.

Jacobi iteration[6]. For both cycling computations, SVD-MPE and MPE seem to perform very similarly in this example.

Note that the Jacobi and Gauss-Seidel iterations converge extremely slowly. In view of this slow convergence, the acceleration produced by SVD-MPE and MPE in the cycling mode is remarkable.

Acknowledgements

The author would like to thank Boaz Ophir for carrying out the computations reported in Section 7 of this work.

References

[1] Cabay, S. and Jackson, L.W. (1976) A Polynomial Extrapolation Method for Finding Limits and Antilimits of Vector Sequences. *SIAM Journal on Numerical Analysis*, **13**, 734-752. http://dx.doi.org/10.1137/0713060

[2] Kaniel, S. and Stein, J. (1974) Least-Square Acceleration of Iterative Methods for Linear Equations. *Journal of Optimization Theory and Applications*, **14**, 431-437. http://dx.doi.org/10.1007/BF00933309

[3] Eddy, R.P. (1979) Extrapolating to the Limit of a Vector Sequence. In: Wang, P.C.C., Ed., *Information Linkage between Applied Mathematics and Industry*, Academic Press, New York, 387-396

[4] Mešina, M. (1977) Convergence Acceleration for the Iterative Solution of the Equations $X = AX + f$. *Computer Methods in Applied Mechanics and Engineering*, **10**, 165-173. http://dx.doi.org/10.1016/0045-7825(77)90004-4

[5] Brezinski, C. (1975) Généralisations de la transformation de Shanks, de la table de Padé, et de l'ε-algorithme. *Calcolo*, **12**, 317-360. http://dx.doi.org/10.1007/BF02575753

[6] Pugachev, B.P. (1978) Acceleration of the Convergence of Iterative Processes and a Method of Solving Systems of Nonlinear Equations. *USSR Computational Mathematics and Mathematical Physics*, **17**, 199-207. http://dx.doi.org/10.1016/0041-5553(77)90023-4

[7] Sidi, A., Ford, W.F., and Smith, D.A. (1986) Acceleration of Convergence of Vector Sequences. *SIAM Journal on Numerical Analysis*, **23**, 178-196. http://dx.doi.org/10.1137/0723013

[8] Wynn, P. (1956) On a Device for Computing the $e_m(S_n)$ Transformation. *Mathematical Tables and Other Aids to Computation*, **10**, 91-96. http://dx.doi.org/10.2307/2002183

[9] Shanks, D. (1955) Nonlinear Transformations of Divergent and Slowly Convergent Sequences. *Journal of Mathematics and Physics*, **34**, 1-42. http://dx.doi.org/10.1002/sapm19553411

[10] Wynn, P. (1962) Acceleration Techniques for Iterated Vector and Matrix Problems. *Mathematics of Computation*, **16**, 301-322. http://dx.doi.org/10.1090/S0025-5718-1962-0145647-X

[11] Smith, D.A., Ford, W.F., and Sidi, A. (1987) Extrapolation Methods for Vector Sequences. *SIAM Review*, **29**, 199-233. http://dx.doi.org/10.1137/1029042

[12] Sidi, A. (2008) Vector Extrapolation Methods with Applications to Solution of Large Systems of Equations and to PageRank Computations. *Computers & Mathematics with Applications*, **56**, 1-24. http://dx.doi.org/10.1016/j.camwa.2007.11.027

[13] Sidi, A. (2012) Review of Two Vector Extrapolation Methods of Polynomial Type with Applications to Large-Scale Problems. *Journal of Computational Science*, **3**, 92-101. http://dx.doi.org/10.1016/j.jocs.2011.01.005

[14] Sidi, A. (1991) Efficient Implementation of Minimal Polynomial and Reduced Rank Extrapolation Methods. *Journal of Computational and Applied Mathematics*, **36**, 305-337. http://dx.doi.org/10.1016/0377-0427(91)90013-A

[15] Jbilou, K. and Sadok, H. (1999) LU-Implementation of the Modified Minimal Polynomial Extrapolation Method. *IMA Journal of Numerical Analysis*, **19**, 549-561. http://dx.doi.org/10.1093/imanum/19.4.549

[16] Sidi, A. (1983) Convergence and Stability Properties of Minimal Polynomial and Reduced Rank Extrapolation Algorithms. *SIAM Journal on Numerical Analysis*, **23**, 197-209. http://dx.doi.org/10.1137/0723014

[17] Sidi, A. (1988) Extrapolation vs. Projection Methods for Linear Systems of Equations. *Journal of Computational and Applied Mathematics*, **22**, 71-88. http://dx.doi.org/10.1016/0377-0427(88)90289-0

[18] Sidi, A. (1994) Convergence of Intermediate Rows of Minimal Polynomial and Reduced Rank Extrapolation Tables. *Numerical Algorithms*, **6**, 229-244. http://dx.doi.org/10.1007/BF02142673

[19] Sidi, A. and Bridger, J. (1988) Convergence and Stability Analyses for Some Vector Extrapolation Methods in the

[6]Recall that the Gauss-Seidel method converges twice as fast as the Jacobi method when the two methods are applied to linear systems with consistently ordered matrices, and that the matrix obtained by discretizing $-\nabla^2 u$ by central differences is consistently ordered; see [31], for example. We believe this could explain the behavior of the solution with Gauss-Seidel iteration relative to that with Jacobi iteration, despite the fact that the system of equations we are solving in Example 2 is nonlinear.

Presence of Defective Iteration Matrices. *Journal of Computational and Applied Mathematics*, **22**, 35-61. http://dx.doi.org/10.1016/0377-0427(88)90287-7

[20] Sidi, A. and Shapira, Y. (1992) Upper Bounds for Convergence Rates of Vector Extrapolation Methods on Linear Systems with Initial Iterations. Technical Report 701, Computer Science Department, Technion-Israel Institute of Technology.

[21] Sidi, A. and Shapira, Y. (1998) Upper Bounds for Convergence Rates of Acceleration Methods with Initial Iterations. *Numerical Algorithms*, **18**, 113-132. http://dx.doi.org/10.1023/A:1019113314010

[22] Brezinski, C. (1970) Application de l'ε-algorithme à la résolution des systèmes non linéaires. *Comptes Rendus de l'Académie des Sciences*, **271A**, 1174-1177.

[23] Brezinski, C. (1971) Sur un algorithme de résolution des systèmes non linéaires. *Comptes Rendus de l'Académie des Sciences*, **272A**, 145-148.

[24] Gekeler, E. (1972) On the Solution of Systems of Equations by the Epsilon Algorithm of Wynn. *Mathematics of Computation*, **26**, 427-436. http://dx.doi.org/10.1090/S0025-5718-1972-0314226-X

[25] Wynn, P. (1963) Continued Fractions Whose Coefficients Obey a Noncommutative Law of Multiplication. *Archive for Rational Mechanics and Analysis*, **12**, 273-312. http://dx.doi.org/10.1007/BF00281229

[26] Wynn, P. (1964) General Purpose Vector Epsilon Algorithm Procedures. *Numerische Mathematik*, **6**, 22-36. http://dx.doi.org/10.1007/BF01386050

[27] Graves-Morris, P.R. (1983) Vector Valued Rational Interpolants I. *Numerische Mathematik*, **42**, 331-348. http://dx.doi.org/10.1007/BF01389578

[28] Graves-Morris, P.R. (1992) Extrapolation Methods for Vector Sequences. *Numerische Mathematik*, **61**, 475-487. http://dx.doi.org/10.1007/BF01385521

[29] Golub, G.H. and Van Loan, C.F. (2013) Matrix Computations. 4th Edition, Johns Hopkins University Press, Baltimore.

[30] Horn, R.A. and Johnson, C.R. (1985) Matrix Analysis. Cambridge University Press, Cambridge. http://dx.doi.org/10.1017/CBO9780511810817

[31] Stoer, J. and Bulirsch, R. (2002) Introduction to Numerical Analysis. 3rd Edition, Springer-Verlag, New York. http://dx.doi.org/10.1007/978-0-387-21738-3

[32] Trefethen, L.N. and Bau, D. (1997) Numerical Linear Algebra. SIAM, Philadelphia.

[33] Householder, A.S. (1964) The Theory of Matrices in Numerical Analysis. Blaisedell, New York.

[34] Golub, G.H. and Kahan, W. (1965) Calculating the Singular Values and Pseudo-Inverse of a Matrix. *SIAM Journal on Numerical Analysis*, **2**, 205-224. http://dx.doi.org/10.1137/0702016

[35] Chan, T.F. (1982) An Improved Algorithm for Computing the Singular value Decomposition. *ACM Transactions on Mathematical Software*, **8**, 72-83. http://dx.doi.org/10.1145/355984.355990

[36] Arnoldi, W.E. (1951) The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem. *Quarterly of Applied Mathematics*, **9**, 17-29.

[37] Saad, Y. and Schultz, M.H. (1986) GMRES: A Generalized Minimal Residual Method for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, **7**, 856-869. http://dx.doi.org/10.1137/0907058

[38] Lanczos, C. (1952) Solution of Systems of Linear Equations by Minimized Iterations. *Journal of Research of the National Bureau of Standards*, **49**, 33-53. http://dx.doi.org/10.6028/jres.049.006

[39] Kelley, C.T. (1995) Iterative Methods for Linear and Nonlinear Equations. SIAM, Philadelphia. http://dx.doi.org/10.1137/1.9781611970944