Journal of Computational Science 3 (2012) 92–101

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Review of two vector extrapolation methods of polynomial type with applications to large-scale problems

Avram Sidi

Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 26 July 2010
Received in revised form 5 January 2011
Accepted 17 January 2011
Available online 16 February 2011

Keywords:
Vector extrapolation methods
Acceleration of convergence
Minimal polynomial extrapolation
Reduced rank extrapolation
Arnoldi method
GMRES
Large-scale sparse systems
Eigenvector computation
PageRank computation
Vector-valued rational approximation

ABSTRACT

An important problem that arises in different areas of science and engineering is that of computing limits of sequences of vectors $\{x_n\}$, where $x_n \in \mathbb{C}^N$ with N very large. Such sequences arise, for example, in the solution of systems of linear or nonlinear equations by fixed-point iterative methods, and $\lim_{n\to\infty} x_n$ are simply the required solutions. In most cases of interest, these sequences converge to their limits extremely slowly, or even diverge. One practical way to make the sequences $\{x_n\}$ converge more quickly is to apply to them *vector extrapolation methods*. In this work, we review two polynomial-type vector extrapolation methods that have proved to be very efficient convergence accelerators; namely, the *minimal polynomial extrapolation* (MPE) and the *reduced rank extrapolation* (RRE). We discuss their derivation, describe the most accurate and stable algorithms for their implementation along with the effective modes of usage, and present their convergence and stability theory. We also discuss their close connection with the method of Arnoldi and GMRES, two well known *Krylov subspace methods* for linear systems. Finally, we discuss some of their applications to different large-scale problems, such as solution of large-scale systems of equations, eigenvalue problems, computation of the *PageRank* of the Google matrix, and summation of vector-valued power series.

1. Introduction

An important problem that arises in different areas of science and engineering is that of computing limits of sequences of vectors $\{x_n\}$, where $x_n \in \mathbb{C}^N$ with N very large. Vector sequences arise, for example, in the solution of systems of linear or nonlinear equations by fixed-point iterative methods, and $\lim_{n\to\infty} x_n$ are simply the required solutions. Such systems, in turn, arise from finite-difference or finite-element discretization of continuum problems.

For example, if

$$\psi(x) = 0; \quad \psi : \mathbb{C}^N \to \mathbb{C}^N, \tag{1.1}$$

is such a system of equations, then $\{x_n\}$ is generated by some fixed-point iterative method as in

$$x_{n+1} = F(x_n), \quad n = 0, 1, \ldots; \quad F : \mathbb{C}^N \to \mathbb{C}^N, \tag{1.2}$$

where $x - F(x) = 0$ is a possibly "preconditioned" form of (1.1) hence has the same solution s [that is, $\psi(s) = 0$ and also $s = F(s)$], and, in case of convergence, $\lim_{n\to\infty} x_n = s$. (Here, x_0 is an initial approximation to s.) One possible form of $F(x)$ would be $F(x) = x + C(x)\psi(x)$, where $C(x)$ is an $N \times N$ matrix that may involve a number of relaxation parameters, such that $C(s)$ is nonsingular.

In most cases of interest, the sequences $\{x_n\}$ converge to their limits very slowly. This happens, for example, when they arise from the finite-difference or finite-element discretizations of continuum problems; in such cases, their rates of convergence become smaller as the relevant mesh-sizes become smaller, in which case their size N becomes larger, in addition. In view of this, we ask whether something can be done to make the convergence of the sequences $\{x_n\}$ faster without having to tamper with the fixed-point methods used to generate them. The answer to this question is in the affirmative; namely, we can accelerate their convergence by applying to them *vector extrapolation methods*, which are especially suitable when the dimension N is very large. It is important to mention also that extrapolation methods can cause a divergent sequence generated as above to converge under certain circumstances.

There are several vector extrapolation methods in the literature; for a survey of these methods covering earlier developments, see the review article by Smith et al. [32]. See also Brezinski and Redivo Zaglia [5, Chapter 4]. Briefly speaking, vector extrapolation methods can be classified in two major groups: (i) polynomial methods, and (ii) epsilon algorithms. Two important polynomial methods are the *minimal polynomial extrapolation* (MPE) of Cabay and Jackson [6] and the *reduced rank extrapolation* (RRE) of Eddy [7] and Mešina [16]. Important methods in the epsilon class are the

E-mail address: asidi@cs.technion.ac.il
URL: http://www.cs.technion.ac.il/ asidi/.

1877-7503/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jocs.2011.01.005

vector epsilon algorithm (VEA) of Wynn [35], which was the very first vector extrapolation method, and the *topological epsilon algorithm* (TEA) of Brezinski [4].

Theoretical results and numerical experiments tend to suggest that MPE and RRE are, generally speaking, more efficient than the rest of the methods as convergence accelerators. Indeed, MPE and RRE have been used successfully as effective accelerators in diverse areas of science and engineering, such as computational fluid dynamics, structures, materials, semiconductor research, computerized tomography, and image processing, to name a few. In this work, we present a review of these two methods. We discuss their derivation, describe the most accurate and stable algorithms for their implementation along with the effective modes of usage, and present their convergence and stability theory. We also discuss their close connection with the method of Arnoldi [1] and GMRES [17], two well known *Krylov subspace methods* for linear systems.

Finally, we discuss several applications of vector extrapolation methods to different problems, starting with the solution of nonlinear systems of equations. We next consider the computation of an eigenvector of an arbitrary large and sparse matrix, corresponding to its largest eigenvalue when this eigenvalue is known. This problem has attracted much attention recently because it arises in the computation of the *PageRank* of the Google Web matrix. We next discuss the summation of vector-valued power series by vector-valued rational functions, a problem that arises, for example, when solving differential equations with a small parameter.

Before we end this section, we would like to mention that both MPE and RRE use the vector sequence $\{x_n\}$ as their only input, and compute approximations to $\lim_{n\to\infty} x_n$ that are of the form

$$s_{n,k} = \sum_{i=0}^{k} \gamma_i^{(n,k)} x_{n+i}, \tag{1.3}$$

where $\gamma_i^{(n,k)}$ are scalars depending nonlinearly on the x_m, also satisfying

$$\sum_{i=0}^{k} \gamma_i^{(n,k)} = 1. \tag{1.4}$$

Technically, the two methods differ in the way they compute the $\gamma_i^{(n,k)}$. Therefore, MPE and RRE (and the rest of the vector extrapolation methods as well) can be applied to any vector sequence whether generated linearly or nonlinearly. This is an important feature of these methods.

Throughout this work, we shall be using the standard Euclidean inner product $(u, v) = u^*v$ and the vector norm induced by it $\|u\| = \sqrt{(u, u)}$. Specifically, if $u = [\xi_1, \ldots, \xi_N]^{\mathrm{T}}$ and $v = [\eta_1, \ldots, \eta_N]^{\mathrm{T}}$, then $(u, v) = \sum_{i=1}^{N} \overline{\xi}_i \eta_i$. Note, however, that any inner product, and the norm induced by it, can be used.

2. Preliminaries and motivation

In order to motivate the derivation of vector extrapolation methods, we first look at the problems for which they were designed. Thus, we start by discussing the nature of the vectors x_n that arise from (1.2). Assuming that $\lim_{n\to\infty} x_n$ exists, hence that $x_n \approx s$ for all large n [recall that s is the solution to the system $\psi(x) = 0$ and hence to the system $x = F(x)$], we expand $F(x_n)$ in (1.2) about s, thus obtaining

$$x_{n+1} = F(s) + F'(s)(x_n - s) + O(\|x_n - s\|^2) \quad \text{as } n \to \infty. \tag{2.1}$$

Here, $F'(s)$ is the Jacobian matrix of the vector-valued function $F(x)$ evaluated at $x = s$. It is known that convergence will take place from any x_0 sufficiently close to s provided $\rho(F'(s)) < 1$, where $\rho(A)$ stands for the spectral radius of the (square) matrix A. See Atkinson [2], for example. The result in (2.1) can be expressed also as in

$$\begin{aligned} &x_{n+1} = Tx_n + b + O(\|x_n - s\|^2) \quad \text{as } n \to \infty; \\ &T = F'(s), \quad b = [I - F'(s)]s, \end{aligned} \tag{2.2}$$

T being a constant matrix and b being a constant vector. In other words, we have

$$x_{n+1} \approx Tx_n + b \quad \text{for all large } n.$$

We have thus shown that the system $x = F(x)$ "behaves" linearly when x is close to the solution s. This then suggests that we should look at linear systems for deriving vector extrapolation methods.

3. Derivation of MPE and RRE

The treatment we give in this section follows in part that of Smith et al. [32]. Consider the linear system

$$x = Tx + b, \quad b, x \in \mathbb{C}^N, \quad T \in \mathbb{C}^{N\times N}. \tag{3.1}$$

Assume that $(I - T)$ is nonsingular, which means that 1 is not an eigenvalue of T. As a result, the system in (3.1) has a unique solution that we denote by s. Let us choose an initial vector $x_0 \in \mathbb{C}^N$ and generate the sequence $\{x_n\}$ via

$$x_{n+1} = Tx_n + b, \quad n = 0, 1, \ldots. \tag{3.2}$$

Let us also define

$$\begin{aligned} &e_n = x_n - s, \quad u_n = x_{n+1} - x_n, \\ &w_n = u_{n+1} - u_n, \quad n = 0, 1, \ldots. \end{aligned} \tag{3.3}$$

It is easy to see that

$$\begin{aligned} &e_n = T^n e_0, \quad u_n = T^n u_0, \\ &w_n = T^n w_0, \quad n = 0, 1, \ldots. \end{aligned} \tag{3.4}$$

In addition,

$$u_n = (T - I)e_n, \quad n = 0, 1, \ldots. \tag{3.5}$$

3.1. Construction of solution to $x = Tx + b$ via $\{x_n\}$

The solution s to the linear system $x = Tx + b$ can be constructed using a finite number of the vectors x_n as we show next. This will be the starting point for the development of MPE and RRE.

Denote by $R(\lambda)$ the characteristic polynomial of T; namely, $R(\lambda) = \det(\lambda I - T)$ and is monic. Then, by the Cayley–Hamilton theorem, $R(T) = 0$. See Horn and Johnson [13, p. 86], for example. Similarly, denote by $Q(\lambda)$ the minimal polynomial of T; namely, $Q(\lambda)$ is the unique monic polynomial of smallest degree with the property $Q(T) = 0$. See [13, p. 142], for example. In addition, we know that $Q(\lambda)$ divides $R(\lambda)$. We next need the concept of the so-called *minimal polynomial of a matrix with respect to a vector*, which is less well known:

Definition 3.1. Let $y \neq 0$ be a vector in $\mathbb{C}^N$. The monic polynomial $P(\lambda)$ is said to be the *minimal polynomial of T with respect to y* if $P(T)y = 0$ and $P(\lambda)$ has smallest degree.

Concerning $P(\lambda)$, we have the following result (see Householder [14, p. 18], for example):

Theorem 3.2. (i) *The minimal polynomial of T with respect to the vector y exists and is unique.* (ii) *If $M(\lambda)$ is another monic polynomial for which $M(T)y = 0$, then $\deg M > \deg P$ and $P(\lambda)$ divides $M(\lambda)$. In particular, $P(\lambda)$ divides the minimal polynomial of T, which, in turn, divides the characteristic polynomial of T.*

Proof. Let $Q(\lambda)$ be the minimal polynomial of T. Then $Q(T) = 0$. Consequently, $Q(T)y = 0$ too, and this implies that that $P(\lambda)$ exists. Let $\hat{P}(\lambda)$ be another minimal polynomial of T with respect to

y, and $\hat{P}(\lambda) \not\equiv P(\lambda)$. Of course, $\deg P = \deg\hat{P}$ must hold. Then $S(\lambda) = P(\lambda) - \hat{P}(\lambda)$ also satisfies $S(T)y = 0$ but $\deg S < \deg P$, which is impossible. Therefore, $P(\lambda)$ is unique. Let now $M(\lambda)$ be a monic polynomial with the properties that $M(T)y = 0$ and $\deg M > \deg P$. Then there exist polynomials $G(\lambda)$ and $H(\lambda)$, such that $M = GP + H$, $\deg G = \deg M - \deg P$, and $\deg H < \deg P$. With these, we have

$$0 = M(T)y = G(T)P(T)y + H(T)y = H(T)y.$$

Since $\deg H < \deg P$, $H(T)y = 0$ is possible only if $H(\lambda) \equiv 0$. Thus, $P(\lambda)$ divides $M(\lambda)$. The rest of the proof can be carried out similarly. □

Let $P(\lambda) = \sum_{j=0}^{k} c_j\lambda^j$, $c_k = 1$, be the minimal polynomial of T with respect to the vector e_0. (Of course, $k = \deg P \le N$ by Theorem 3.2.) Thus, $P(T)e_0 = 0$, which, by (3.3) and (3.4), also means that

$$\sum_{j=0}^{k} c_j T^j(x_0 - s) = \sum_{j=0}^{k} c_j(x_j - s) = 0. \tag{3.6}$$

Solving for s, we have

$$s = \frac{\sum_{j=0}^{k} c_j x_j}{\sum_{j=0}^{k} c_j}. \tag{3.7}$$

Note that division by $\sum_{j=0}^{k} c_j$ in (3.7) is allowed, because $\sum_{j=0}^{k} c_j = P(1) \neq 0$ since $P(\lambda)$ divides the characteristic polynomial of T and 1 is not an eigenvalue of T.

We have thus shown that the solution s can be constructed from the $k+1$ vectors of iteration $x_0, x_1, \ldots, x_k$, *provided* $P(\lambda)$ *is known*. Now, being the minimal polynomial of T with respect to $e_0 = x_0 - s$, $P(\lambda)$ depends on s, as well as on x_0. This may lead us to conclude that, in order to know $P(\lambda)$, we must know s. Fortunately, this is not the case, as we show in the next theorem.

Theorem 3.3. *$P(\lambda)$, the minimal polynomial of T with respect to $e_0 = x_0 - s$, is also the minimal polynomial of T with respect to $u_0 = x_1 - x_0$.*

Proof. Denote the minimal polynomial of T with respect to $u_0 = x_1 - x_0$ by $\hat{P}(\lambda)$. Multiplying $P(T)e_0 = 0$ on the left by $T - I$ and invoking (3.5), we obtain $P(T)u_0 = 0$, which, by part (ii) of Theorem 3.2, means that $\hat{P}(\lambda)$ divides $P(\lambda)$. Next, multiplying $\hat{P}(T)u_0 = 0$ on the left by $(T - I)^{-1}$ (recall that $T - I$ is nonsingular) and invoking (3.5) again, we obtain $\hat{P}(T)e_0 = 0$, which, by part (ii) of Theorem 3.2, means that $P(\lambda)$ divides $\hat{P}(\lambda)$. Therefore, $P(\lambda) \equiv \hat{P}(\lambda)$. □

As is clear from Theorem 3.3, $P(\lambda) = \sum_{j=0}^{k} c_j\lambda^j$ can be determined because the vector $u_0 = x_1 - x_0$ is available. Now, since $P(\lambda)$ is monic, we have $c_k = 1$, and hence our unknowns are $c_0, c_1, \ldots, c_{k-1}$. First, by (3.4), $P(T)u_0 = 0$ can be re-expressed as in

$$\sum_{j=0}^{k} c_j T^j u_0 = \sum_{j=0}^{k} c_j u_j = 0,$$

which, in turn, can be expressed as a system of linear equations in the form

$$U_{k-1}c' = -u_k; \quad c' = [c_0, c_1, \ldots, c_{k-1}]^{\mathrm{T}} \in \mathbb{C}^k, \tag{3.8}$$

where

$$U_j = [u_0 \,|\, u_1 \,|\, \cdots \,|\, u_j] \in \mathbb{C}^{N\times(j+1)}, \quad j = 0, 1, \ldots. \tag{3.9}$$

Note that the matrix U_{k-1} has full rank, that is, $\mathrm{rank}(U_{k-1}) = k$, because its columns, namely, the vectors $u_0, u_1, \ldots, u_{k-1}$, are linearly independent by the fact that $P(\lambda) = \sum_{j=0}^{k} c_j\lambda^j$ is the minimal polynomial of T with respect to u_0. Hence, the linear system $U_{k-1}c' = -u_k$ has a unique solution for $c_0, c_1, \ldots, c_{k-1}$. Invoking also $c_k = 1$, we complete the determination of s as in (3.7).

Let

$$\gamma_j = \frac{c_j}{\sum_{i=0}^{k} c_i}, \quad j = 0, 1, \ldots, k.$$

Therefore, we also have

$$\sum_{j=0}^{k} \gamma_j = 1.$$

Multiplying both sides of (3.8) by $\left(\sum_{j=0}^{k} c_j\right)^{-1}$, it is easy to see that the γ_j satisfy the system of linear equations

$$U_k\gamma = 0, \quad \sum_{j=0}^{k} \gamma_j = 1; \quad \gamma = [\gamma_0, \gamma_1, \ldots, \gamma_k]^{\mathrm{T}} \in \mathbb{C}^{k+1}. \tag{3.10}$$

Here too U_j is as defined in (3.9). With these, we can write (3.7) also in the form

$$s = \sum_{j=0}^{k} \gamma_j x_j; \quad \sum_{j=0}^{k} \gamma_j = 1. \tag{3.11}$$

This suggests that s is some sort of "weighted" average of the vectors $x_0, x_1, \ldots, x_k$. Of course, the γ_j do not have to be nonnegative; they do not even have to be real.

By the developments above, it is clear that s can be constructed from the knowledge of the $k+2$ vectors x_j, $j = 0, 1, \ldots, k+1$.

The developments above leading to the solution s given in (3.7) or (3.11) are made possible by the fact that k is the degree of the minimal polynomial of T with respect to u_0. Normally, k would be very close to N, and this would make the solution process prohibitively expensive in the sense that its computational cost and computer memory requirements would be extremely large. This raises the question as to what would happen if we took k to be an arbitrary integer (and much smaller than N). Obviously, the linear systems $U_{k-1}c' = -u_k$ in (3.8) and $U_k\gamma = 0$ and $\sum_{j=0}^{k} \gamma_j = 1$ in (3.10) will be inconsistent, hence will not have a solution in the ordinary sense. We address this question next.

3.2. Derivation of MPE

Let us *define* c' as the least-squares solution of the system $U_{k-1}c' = -u_k$, leaving everything else unchanged. This results in MPE, whose complete definition is as follows:

Definition 3.4. Given the vector sequence $\{x_n\}$ in $\mathbb{C}^N$, choose $k \le N$ arbitrarily. Let the vectors u_n be as in (3.3), and define the matrix U_{k-1} via

$$U_{k-1} = [u_0 \,|\, u_1 \,|\, \cdots \,|\, u_{k-1}] \in \mathbb{C}^{N\times k}.$$

Let $c' = [c_0, c_1, \ldots, c_{k-1}]^{\mathrm{T}}$ be the least-squares solution to the linear system $U_{k-1}c' = -u_k$; this means that c' is the solution to the problem

$$\min_{c_0, c_1, \ldots, c_{k-1}} \left\| \sum_{j=0}^{k-1} c_j u_j + u_k \right\|.$$

Set $c_k = 1$, and compute $\gamma_0, \gamma_1, \ldots, \gamma_k$ via

$$\gamma_j = \gamma_j^{\mathrm{MPE}} = \frac{c_j}{\sum_{i=0}^{k} c_i}, \quad j = 0, 1, \ldots, k, \tag{3.12}$$

provided $\sum_{i=0}^{k} c_i \neq 0$. Then, the MPE approximation to $s = \lim_{n\to\infty} x_n$, denoted s_k^{MPE}, is given by

$$s_k^{\mathrm{MPE}} = \sum_{j=0}^{k} \gamma_j^{\mathrm{MPE}} x_j. \tag{3.13}$$

3.3. Derivation of RRE

Let us *define* γ as the least-squares solution of the system $U_k\gamma = 0$, subject to the constraint $\sum_{j=0}^{k}\gamma_j = 1$, leaving everything else unchanged. This results in RRE, whose complete definition is as follows:

Definition 3.5. Given the vector sequence $\{x_n\}$ in $\mathbb{C}^N$, choose $k \leq N$ arbitrarily. Let the vectors u_n be as in (3.3), and let the matrix U_k be defined via

$$U_k = [u_0 \mid u_1 \mid \cdots \mid u_k] \in \mathbb{C}^{N\times(k+1)}.$$

Let γ be the least-squares solution to the linear system $U_k\gamma = 0$ subject to the constraint $\sum_{j=0}^{k}\gamma_j = 1$; this means that γ is the solution to the constrained minimization problem

$$\min_{\gamma_0,\gamma_1,\ldots,\gamma_k} \left\| \sum_{j=0}^{k} \gamma_j u_j \right\|, \quad \text{subject to} \sum_{j=0}^{k} \gamma_j = 1.$$

If we denote the solution to this minimization problem by γ_j^{RRE}, then the RRE approximation to $s = \lim_{n\to\infty} x_n$, denoted s_k^{RRE}, is given by

$$s_k^{\mathrm{RRE}} = \sum_{j=0}^{k} \gamma_j^{\mathrm{RRE}} x_j. \tag{3.14}$$

Note that the way we have approached the definition of RRE here is not the only way possible, and it differs from that of [32] and those of the original works [7,16]. (The approaches of [7,16] differ from each other greatly, and their equivalence is established in [32].) Our approach here is the most direct, however. The definition of RRE here is also that given in [16], and turns out to be very suitable for computational purposes. For a completely different approach, see Sidi et al. [29].

For completeness, here we reproduce the definition of [7], which writes s_k^{RRE} in the form

$$s_k^{\mathrm{RRE}} = x_0 + \sum_{i=0}^{k-1} \xi_i u_i, \tag{3.15}$$

with no constraints on the ξ_i. [Note that this is consistent with (3.14) and $\sum_{j=0}^{k}\gamma_j^{\mathrm{RRE}} = 1$.] The ξ_i are determined from the least-squares solution of the linear system

$$W_{k-1}\xi = -u_0; \quad \xi = [\xi_0, \xi_1, \ldots \xi_{k-1}]^{\mathrm{T}} \in \mathbb{C}^k, \tag{3.16}$$

where

$$W_j = [w_0 \mid w_1 \mid \cdots \mid w_j] \in \mathbb{C}^{N\times(j+1)}. \tag{3.17}$$

Here, the w_n, just as the u_n, are as defined in (3.3). Thus, s_k^{RRE} can also be expressed in the compact form

$$s_k^{\mathrm{RRE}} = x_0 - U_{k-1}W_{k-1}^{+}u_0, \tag{3.18}$$

where U_{k-1} is as before and W_{k-1}^{+} is the Moore–Penrose inverse of W_{k-1}. For the Moore–Penrose inverse and for other generalized inverses of matrices, see Ben-Israel and Greville [3], for example.

3.4. Some remarks on MPE and RRE

As can easily be seen from Definitions 3.4 and 3.5, the approximations produced by MPE and RRE are defined *exclusively* in terms of the vectors x_j, nothing else being required as input. Even though the derivation of the two methods was based on the solution of linear systems $x = Tx + b$, their definition is totally independent of the way in which these vectors are generated. Thus, these methods can be used for (hopefully) accelerating the convergence of vector sequences $\{x_n\}$, whether these sequences are generated by linear systems or not. By the way they are derived, it is reasonable to assume that MPE and RRE will be effective accelerators for vector sequences arising from fixed-point iteration techniques on linear systems. Since nonlinear systems behave linearly close to their solutions, we can hope that MPE and RRE will be effective for accelerating the convergence of vector sequences $\{x_n\}$ generated by fixed-point methods on nonlinear systems as well.

Because the least-squares problem for the γ_j always has a solution for RRE, we conclude that s_k^{RRE} exists unconditionally for all k. In case of MPE, even though there is always a solution for the c_j, we cannot guarantee that $\sum_{i=0}^{k} c_i \neq 0$ always. This means that s_k^{MPE} may not always exist. A sufficient condition for the existence of s_k^{MPE} is given in Theorem 6.1.

3.5. Connection with Krylov subspace methods

When applied to a sequence $\{x_n\}$ generated by the fixed-point iterative technique in (3.2) for solving the linear system in (3.1), MPE and RRE become equivalent to some well known Krylov subspace methods for solving (3.1), namely, the method of Arnoldi [1] and GMRES [17], respectively, when the latter are applied with x_0 as the initial vector. We recall that, for each $k = 1, 2, \ldots$, both Krylov subspace methods compute a vector z_k of the form $z_k = x_0 + y_k$, with $y_k \in \mathcal{K}_k(A; r_0)$, where $A = I - T$, $r_0 = b - Ax_0$, and $\mathcal{K}_k(A; r_0) = \mathrm{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}$. Let us define $r(x) = b - Ax$ as the residual vector associated with x. The approximation z_k^{Arnoldi} to s is then determined by requiring that $r(z_k)$ be such that $(y, r(z_k)) = 0$ for all vectors $y \in \mathcal{K}_k(A; r_0)$. The approximation z_k^{GMRES} to s is determined by requiring that $r(z_k)$ be such that $(Ay, r(z_k)) = 0$ for all vectors $y \in \mathcal{K}_k(A; r_0)$, or equivalently, by requiring that $\|r(z_k)\| = \min_{y\in\mathcal{K}_k(A;r_0)}\|r(x_0 + y)\|$. We then have the following theorem proved in Sidi [20]:

Theorem 3.6. *With s_k^{MPE}, s_k^{RRE}, z_k^{Arnoldi}, and z_k^{GMRES} as in the preceding paragraph, there hold*

$$s_k^{\mathrm{MPE}} = z_k^{\mathrm{Arnoldi}} \quad \text{and} \quad s_k^{\mathrm{RRE}} = z_k^{\mathrm{GMRES}}.$$

4. Algorithms for MPE and RRE

The definitions of MPE and RRE given above can also be used to design algorithms (computational procedures) for implementing MPE and RRE. The most immediate, and computationally inexpensive, algorithms would be those that use the normal equations (i) $U_{k-1}^{*}U_{k-1}c = -U_{k-1}^{*}u_k$ in Definition 3.4 to determine the c_i for MPE, and (ii) $W_{k-1}^{*}W_{k-1}\xi = -W_{k-1}^{*}u_0$ in (3.16) to determine the ξ_i for RRE. Since matrices U_{k-1} and W_{k-1} become very ill-conditioned with increasing k, these algorithms are not stable numerically, and they produce s_k with reduced accuracy in floating-point arithmetic. Concerning the solution of least-squares problems via normal equations, see Golub and Van Loan [11].

Numerically fast and stable and storagewise economical algorithms have been given in Sidi [21], where a fully documented FORTRAN 77 code can also be found. We now turn to a summary of these algorithms. One important feature of these algorithms is that

they both proceed through the solution of least-squares problems by QR factorization.

For reasons to become clear later, it is more beneficial to apply MPE and RRE to the vectors $x_n, x_{n+1}, \ldots, x_{n+k+1}$ with $n \geq 0$, instead of $x_0, x_1, \ldots, x_{k+1}$. Let us denote the resulting approximations by $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$. (Then, the s_k defined in the preceding section are the $s_{0,k}$ in our new notation.) Thus, given a sequence $\{x_n\}$, MPE and RRE generate two-dimensional tables of approximations to $s = \lim_{n\to\infty} x_n$. In the remainder of this work, we will be dealing with $s_{n,k}$.

To discuss our algorithms conveniently, we introduce the notation

$$U_j = [u_n \,|\, u_{n+1} \,|\, \cdots \,|\, u_{n+j}], \quad j = 0, 1, \ldots.$$

Let us assume that U_j has full rank, namely, $\mathrm{rank}(U_j) = j+1$. Then, it has a QR factorization $U_j = Q_j R_j$, where $Q_j \in \mathbb{C}^{N\times(j+1)}$ is unitary [in the sense that its columns form an orthonormal basis for the column space of U_j, hence $Q_j^* Q_j = I_{(j+1)\times(j+1)}$], and $R_j \in \mathbb{C}^{(j+1)\times(j+1)}$ is upper triangular with positive diagonal entries,

$$Q_j = [q_0 \,|\, q_1 \,|\, \cdots \,|\, q_j] \in \mathbb{C}^{N\times(j+1)}; \quad Q_j^* Q_j = I_{(j+1)\times(j+1)},$$

$$R_j = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0j} \\ & r_{11} & r_{12} & \cdots & r_{1j} \\ & & r_{22} & \cdots & r_{2j} \\ & & & \ddots & \vdots \\ & & & & r_{jj} \end{bmatrix}; \quad r_{ii} > 0 \quad \forall i.$$

Also, Q_j is obtained from Q_{j-1} by appending one column (namely, the vector q_j) to the end of the latter. Similarly, R_j is obtained from R_{j-1} by appending one row of zeros and one column (namely, $[r_{0j}, r_{1j}, \ldots, r_{jj}]^{\mathrm{T}}$) to the end of the latter. This factorization can be carried out accurately and inexpensively by applying the *modified Gram–Schmidt process* (MGS) to the vectors $u_n, u_{n+1}, \ldots, u_{n+j}$. Here are the steps of MGS.

MGS algorithm

Step 1. Compute $r_{00} = \|u_n\|$ and set $q_0 = u_n / r_{00}$.
Step 2. For $k = 1, 2, \ldots,$ do
  Set $u_k^{(0)} = u_{n+k}$
  For $j = 0, 1, \ldots, k-1$ do
    Compute $r_{j,k} = (q_j, u_k^{(j)})$ and
    Compute $u_k^{(j+1)} = u_k^{(j)} - r_{jk} q_j$.
  end for(j)
  Compute $r_{kk} = \|u_k^{(k)}\|$ and $q_k = u_k^{(k)} / r_{kk}$.
end for(k)

(Note that if we replace the statement $u_k^{(j+1)} = u_k^{(j)} - r_{jk} q_j$ in Step 2 of the MGS algorithm by its mathematically equivalent $u_k^{(j+1)} = u_{n+k} - r_{jk} q_j$, we obtain the classical *Gram–Schmidt process* (GS) for orthogonalizing the set of vectors $\{u_n, u_{n+1}, \ldots, u_{n+k}\}$, which is very inaccurate numerically. For GS, MGS, and other orthogonalization processes, see Golub and Van Loan [11].)

The details of our algorithms for MPE and RRE are summarized in Table 1 in a unified fashion. For the mathematical details, see [21]. Note that, in these algorithms, we need to store only the vector x_n and the matrix Q_k, namely, the vectors $q_0, q_1, \ldots, q_k$. The rest

Table 1
Unified algorithm for implementing MPE and RRE.

Step 0. Input: The vectors $x_n, x_{n+1}, \ldots, x_{n+k+1}$.

Step 1. Compute $u_i = \Delta x_i = x_{i+1} - x_i$, $i = n, n+1, \ldots, n+k$.
Set $U_j = [u_n \,|\, u_{n+1} \,|\, \cdots \,|\, u_{n+j}] \in \mathbb{C}^{N\times(j+1)}$, $j = 0, 1, \ldots$.
Compute the QR factorization of U_k, namely, $U_k = Q_k R_k$;
$Q_k = [q_0 \,|\, q_1 \,|\, \cdots \,|\, q_k]$ unitary and $R_k = [r_{ij}]_{0\leq i,j\leq k}$ upper triangular.
($U_{k-1} = Q_{k-1} R_{k-1}$ is contained in $U_k = Q_k R_k$.)

Step 2. Computation of the γ_i:
For MPE:
Solve the (upper triangular) linear system
$R_{k-1} c' = -\rho_k$; $\rho_k = [r_{0k}, r_{1k}, \ldots, r_{k-1,k}]^{\mathrm{T}}$, $c' = [c_0, c_1, \ldots, c_{k-1}]^{\mathrm{T}}$.
(Note that $\rho_k = Q_{k-1}^* u_{n+k}$.)
Set $c_k = 1$ and compute $\alpha = \sum_{i=0}^{k} c_i$.
Set $\gamma_i = c_i/\alpha$, $i = 0, 1, \ldots, k$.
For RRE:
Solve the linear system
$R_k^* R_k d = e$; $d = [d_0, d_1, \ldots, d_k]^{\mathrm{T}}$, $e = [1, 1, \ldots, 1]^{\mathrm{T}} \in \mathbb{C}^{k+1}$.
[This amounts to solving two triangular (lower and upper) systems.]
Set $\lambda = \left(\sum_{i=0}^{k} d_i\right)^{-1}$. (Note that λ is real and positive.)
Set $\gamma = \lambda d$, that is, $\gamma_i = \lambda d_i$, $i = 0, 1, \ldots, k$.

Step 3. Compute $\xi = [\xi_0, \xi_1, \ldots, \xi_{k-1}]^{\mathrm{T}}$ by
$\xi_0 = 1 - \gamma_0$; $\xi_j = \xi_{j-1} - \gamma_j$, $j = 1, \ldots, k-1$.
Compute $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$ via
$s_{n,k} = x_n + Q_{k-1}(R_{k-1}\xi)$.
[For this, first compute $\eta = R_{k-1}\xi$, $\eta = [\eta_0, \eta_1, \ldots, \eta_{k-1}]^{\mathrm{T}}$.
Next, set $s_{n,k} = x_0 + \sum_{i=0}^{k-1} \eta_i q_i$.]

can be overwritten as soon as they have been used. Thus, for $i \geq 1$, x_{n+i} is overwritten by u_{n+i} once the latter is computed, and u_{n+i} is overwritten by q_i once the latter is computed.

The description of MGS given here and Table 1 should enable the reader to produce his own code easily, without having to use other software. Note also that the two algorithms differ only in their Step 2, where the γ_j are determined. This enables us to unify their presentation and programming in a simple way.

The error $s_{n,k} - s$ can be ascertained by looking at $||U_k\gamma||$, which is available at no additional cost through the algorithms in Table 1, without having to compute $s_{n,k}$. Actually, we have

$$\|U_k\gamma\| = \begin{cases} r_{kk}|\gamma_k| & \text{for MPE,} \\ \sqrt{\lambda} & \text{for RRE.} \end{cases}$$

Here r_{kk} is the $(k+1, k+1)$ element of the matrix R_k, and λ is the scalar computed in Step 2 of the RRE algorithm in Table 1. The justification of this is as follows: (i) When the sequence $\{x_n\}$ is generated linearly as in (3.2), we have $U_k\gamma = r(s_{n,k})$, where $r(x) = Tx + b - x$ is the residual vector associated with x, hence $U_k\gamma$ is indeed the residual vector $r(s_{n,k})$; consequently, $||r(s_{n,k})|| = ||U_k\gamma||$. (ii) When $\{x_n\}$ is generated nonlinearly as in (1.2), and $s_{n,k}$ is close to the solution s, we have $U_k\gamma \approx r(s_{n,k})$, where now $r(x) = F(x) - x$ is the residual vector associated with x; consequently, $||r(s_{n,k})|| \approx ||U_k\gamma||$.

5. Determinant representations for MPE and RRE

We next give determinant representations for the approximations $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$, derived originally in Sidi [19].

Theorem 5.1. *The approximations $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$ have the following determinant representations*:

$$s_{n,k} = \frac{D(x_n, x_{n+1}, \ldots, x_{n+k})}{D(1, 1, \ldots, 1)}, \tag{5.1}$$

where $D(g_0, g_1, \ldots, g_k)$ is a $(k+1)\times(k+1)$ determinant defined as in

$$D(g_0, g_1, \ldots, g_k) = \begin{vmatrix} g_0 & g_1 & \cdots & g_k \\ u_{0,0} & u_{0,1} & \cdots & u_{0,k} \\ u_{1,0} & u_{1,1} & \cdots & u_{1,k} \\ \vdots & \vdots & & \vdots \\ u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k} \end{vmatrix}, \tag{5.2}$$

with $u_{i,j}$ being scalars defined as in

$$u_{i,j} = \begin{cases} (u_{n+i}, u_{n+j}) & \text{for MPE,} \\ (w_{n+i}, u_{n+j}) & \text{for RRE.} \end{cases} \tag{5.3}$$

Here u_n and w_n are as in (3.3).

Note that, in case the g_j in $D(g_0, g_1, \ldots, g_k)$ are vectors, this determinant is vector-valued and is defined to be $\sum_{j=0}^{k} M_j g_j$, where M_j is the cofactor of g_j.

The determinant representations above have been used in the convergence and stability analyses of $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$. They were used, in particular, to prove the convergence acceleration results of Theorem 6.1. They have also been used by Ford and Sidi [10] to obtain recursion relations among the different $s_{n,k}$. We skip these recursions here and refer the reader to [10].

Our next result pertains to the γ_j, and has been used in the stability analysis of Theorem 6.2.

Theorem 5.2. *Denote γ_j by $\gamma_j^{(n,k)}$ to emphasize their dependence on n and k. The polynomials $Q_{n,k}(\lambda) = \sum_{j=0}^{k} \gamma_j^{(n,k)} \lambda^j$, for both MPE and RRE, have the following determinant representations*:

$$Q_{n,k}(\lambda) = \frac{D(1, \lambda, \ldots, \lambda^k)}{D(1, 1, \ldots, 1)}, \tag{5.4}$$

where $D(g_0, g_1, \ldots, g_k)$ is precisely as in Theorem 5.1.

Before we end this section, we would like to point out to a nice feature of vector extrapolation methods in general, and MPE and RRE in particular: These methods can be defined as in Definitions 3.4 and 3.5, and can be expressed as in Theorems 5.1 and 5.2, in the setting of general infinite-dimensional inner product spaces, as well as $\mathbb{C}^N$ with finite N. The algorithms and the convergence theory remain the same for all practical purposes. The treatments given in [19,23,28,29] cover vector sequences in infinite dimensional spaces.

6. Convergence acceleration and stability properties

The following theorem by Sidi [19] explains why MPE and RRE are true convergence acceleration methods.

Theorem 6.1. *Let the vector sequence $\{x_n\}$ be such that*

$$x_n = s + \sum_{i=1}^{p} v_i \lambda_i^n, \tag{6.1}$$

where $v_1, v_2, \ldots, v_p$ are linearly independent vectors, and λ_i are distinct nonzero scalars satisfying

$$\lambda_i \neq 1 \quad \text{for all } i, \tag{6.2}$$

and are ordered such that

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_p|. \tag{6.3}$$

Let us assume further that, for some integer $k < p$, there holds

$$|\lambda_k| > |\lambda_{k+1}|. \tag{6.4}$$

Then the following hold:

- *Existence: $s_{n,k}^{\mathrm{RRE}}$ exists for all n and $s_{n,k}^{\mathrm{MPE}}$ exists for all large n.*
- *Convergence acceleration:*

$$\begin{aligned} s_{n,k} - s &= [C_{n,k} + o(1)]\lambda_{k+1}^n \quad \text{as } n \to \infty, \\ &= O(\lambda_{k+1}^n) \quad \text{as } n \to \infty, \end{aligned} \tag{6.5}$$

  where $s_{n,k}$ stands for both $s_{n,k}^{\mathrm{MPE}}$ and $s_{n,k}^{\mathrm{RRE}}$, and the vector $C_{n,k}$ is uniformly bounded in n, that is, $\sup_n ||C_{n,k}|| < \infty$. In addition, $C_{n,k}^{\mathrm{MPE}} = C_{n,k}^{\mathrm{RRE}}$, so that

$$s_{n,k}^{\mathrm{MPE}} - s \sim s_{n,k}^{\mathrm{RRE}} - s \quad \text{as } n \to \infty.$$

- *Finite termination property when $k = p$:*

$$s_{n,p} = s. \tag{6.6}$$

Concerning the result (6.5) of Theorem 6.1, we can offer the following interpretation: Starting from (6.1), we first see that the contribution of the small λ_i is being diminished relative to the large ones by letting n grow. The extrapolation procedure then takes care of the contribution from the k largest λ_i. What remains is the contribution from the intermediate λ_i starting with λ_{k+1}. Note also that, even if $\{x_n\}$ is divergent, which happens when $|\lambda_1| > 1$, $\lim_{n\to\infty} s_{n,k} = s$ if $|\lambda_{k+1}| < 1$; that is, MPE and RRE can make a divergent sequence converge.

The next theorem pertains to the polynomial $Q_{n,k}(\lambda) = \sum_{j=0}^{k} \gamma_j^{(n,k)} \lambda^j$, where $\gamma_j^{(n,k)} \equiv \gamma_j$, and implies that MPE and RRE are stable methods under the conditions of Theorem 6.1.

Theorem 6.2. *Under the conditions of Theorem 6.1, the following hold*:

- *Existence of* $\lim_{n\to\infty}Q_{n,k}(\lambda)=\lim_{n\to\infty}\sum_{j=0}^{k}\gamma_j^{(n,k)}\lambda^j$*:*

$$Q_{n,k}(\lambda)=\prod_{i=1}^{k}\frac{\lambda-\lambda_i}{1-\lambda_i}+O(|\lambda_{k+1}/\lambda_k|^m)\quad as\ n\to\infty, \tag{6.7}$$

hence

$$\lim_{n\to\infty}Q_{n,k}(\lambda)=\prod_{i=1}^{k}\frac{\lambda-\lambda_i}{1-\lambda_i}. \tag{6.8}$$

Therefore, $\lim_{n\to\infty}\gamma_j^{(n,k)}$ *all exist too. Here,*

$$m=\begin{cases}2n & \text{if } (v_i,v_j)=0 \quad \text{for } i\neq j,\\ n & \text{otherwise.}\end{cases} \tag{6.9}$$

- *Stability: MPE and RRE are stable in the sense that*

$$\lim_{n\to\infty}\sum_{j=0}^{k}|\gamma_j^{(n,k)}|\le\prod_{i=1}^{k}\frac{1+|\lambda_i|}{|1-\lambda_i|}\quad\Rightarrow\quad\sup_n\sum_{j=0}^{k}|\gamma_j^{(n,k)}|<\infty. \tag{6.10}$$

- *Zeros of* $Q_{n,k}(\lambda)$*: For all large n,* $Q_{n,k}(\lambda)$ *has exactly k zeros* $\lambda_i^{(n,k)}$, $i=1,\dots,k$, *that tend to* λ_i, $i=1,\dots,k$, *respectively. In addition,*

$$\lambda_i^{(n,k)}-\lambda_i=O(|\lambda_{k+1}/\lambda_i|^m)\quad as\ n\to\infty. \tag{6.11}$$

  Her m is as defined in (6.9).
- *Finite termination property when k = p:*

$$\sum_{j=0}^{p}\gamma_j^{(n,p)}\lambda^j=\prod_{i=1}^{p}\frac{\lambda-\lambda_i}{1-\lambda_i}. \tag{6.12}$$

Note that the numerical stability of MPE and RRE depends on the size of the quantity $\Gamma^{(n,k)}=\sum_{j=0}^{k}|\gamma_j^{(n,k)}|$, which is always ≥ 1 because $\sum_{j=0}^{k}\gamma_j^{(n,k)}=1$. When the x_i are given with errors ϵ_i, the computed $s_{n,k}$, which we shall denote $\hat{s}_{n,k}$, roughly speaking, satisfy

$$\|\hat{s}_{n,k}-s_{n,k}\|\lessapprox\Gamma^{(n,k)}\left(\max_{n\le i\le n+k}\|\epsilon_i\|\right).$$

Thus, the smaller $\Gamma^{(n,k)}$, the more stable the extrapolation.

Because $x_n-s=O(\lambda_1^n)$ and $s_{n,k}-s=O(\lambda_{k+1}^n)$ as $n\to\infty$, and $|\lambda_1|\ge|\lambda_k|>|\lambda_{k+1}|$, it is clear that $\{s_{n,k}\}_{n=0}^{\infty}$ tends to s faster than does $\{x_n\}$ when $\{x_n\}$ converges. In case $\{x_n\}$ does not converge (which happens when $|\lambda_1|\ge 1$), $\{s_{n,k}\}_{n=0}^{\infty}$ will converge if $|\lambda_{k+1}|<1$, and it will diverge (but less slowly than $\{x_n\}$) if $|\lambda_{k+1}|\ge 1$. (In case of divergence, s is the antilimit of $\{x_n\}$.)

When $|\lambda_k|=|\lambda_{k+1}|$, the case not covered by Theorem 6.1, the convergence result pertaining to $s_{n,k}$, namely, $s_{n,k}-s=O(\lambda_{k+1}^n)$ as $n\to\infty$, remains the same when $\{x_n\}$ is generated linearly as in (3.2). We have

$$s_{n,k}-s=O(\lambda_{k+1}^n)\quad as\ n\to\infty,$$

(i) for RRE always and (ii) for MPE when $\alpha(I-T)$ has a positive definite hermitian part for some scalar $\alpha\neq 0$. We refer the reader to Sidi [22] for details.

Note that vector sequences $\{x_n\}$ satisfying the conditions of Theorem 6.1 arise from the iterative technique of (3.2) when the matrix T is diagonalizable; in this case, $\lambda_1,\dots,\lambda_p$ are some or all of the distinct nonzero eigenvalues of the matrix T, and the vectors v_p are corresponding eigenvectors, that is, $Tv_i=\lambda_i v_i$, $i=1,\dots,p$, and, of course, $p\le N$. When T is not diagonalizable, the structure of x_n is similar to that in (6.1)–(6.3), but more complicated. Consequently, the corresponding convergence and stability results are also similar to, but more complicated than, those of Theorems 6.1 and 6.2. This general case has been treated completely in Sidi and Bridger [28] and in [22].

The techniques used in proving Theorems 6.1 and 6.2 are rather universal. They have been applied by the author with success in the analysis of several other scalar and vector extrapolation methods, the Richardson extrapolation and the Shanks [18] transformation or the equivalent epsilon algorithm of Wynn [34] being some of them; see Sidi [25].

7. Efficient use of MPE and RRE

7.1. Applying extrapolation to a subsequence $\{x_{rn}\}$

As mentioned already in the preceding section, to determine $s_{n,k}$, we need to store x_n and the columns of the unitary matrix Q_k. Using the *same* storage, we can obtain more accurate approximations to s by applying MPE or RRE to the sequence $\{x_{rn}\}_{n=0}^{\infty}$ with some integer $r>1$, instead of $\{x_n\}$. To see that this is possible, let us analyze the structure of the vectors x_{rn}. By (6.1) of Theorem 6.1, we have

$$x_{rn}=s+\sum_{i=1}^{p}v_i\sigma_i^n,\quad n=1,2,\dots;\quad \sigma_i=\lambda_i^r,\quad i=1,2,\dots.$$

Applying MPE or RRE to $\{x_{rn}\}$, instead of (6.5) for convergence and (6.8) and (6.10) for stability, we now have

$$s_{n,k}-s=O(|\lambda_{k+1}|^{rn})\quad as\ n\to\infty, \tag{7.1}$$

$$\lim_{n\to\infty}\sum_{j=0}^{k}\gamma_j^{(n,k)}\sigma^j=\prod_{i=1}^{k}\frac{\sigma-\sigma_i}{1-\sigma_i}, \tag{7.2}$$

and

$$\lim_{n\to\infty}\Gamma^{(n,k)}=\lim_{n\to\infty}\sum_{j=0}^{k}|\gamma_j^{(n,k)}|\le\prod_{i=1}^{k}\frac{1+|\sigma_i|}{|1-\sigma_i|}. \tag{7.3}$$

These results are better since their upper bounds on $s_{n,k}-s$ and on $\lim_{n\to\infty}\Gamma^{(n,k)}$ are smaller because $|\lambda_i^r|<|\lambda_i|$ when $|\lambda_i|<1$, and $|1-\lambda_i^r|>|1-\lambda_i|$ when λ_i is close to 1 as a complex number. (Note that, in most cases of interest, the largest λ_i are indeed very close to 1.)

It is clear from (7.1) that, when storage is a problem, we can reduce the storage requirements by reducing k, and maintain a given level of accuracy at the same time, by applying MPE or RRE to the sequence $\{x_{rn}\}_{n=0}^{\infty}$ with increasing $r>1$.

7.2. Applying extrapolation in cycling mode

Now, Theorem 6.1 suggests that, as $n\to\infty$, $s_{n,k}$ becomes better as an approximation to s. Thus, there is a beneficial effect to increasing k. On the other hand, this requires an increasing amount of storage, which becomes prohibitive at some point. In case the x_n are produced by the iterative procedure $x_{n+1}=F(x_n)$ for solving $x=F(x)$, we can apply MPE and RRE in a strategy called *cycling* (or *restarting*). Here are the steps of this strategy:

C0. Choose integers n, k, and r, and an initial vector x_0.
C1. Compute the vectors $x_1, x_2, \dots, x_{r(n+k+1)}$ [via $x_{n+1}=F(x_n)$, cf. (1.2)], and save

$$y_n, y_{n+1},\dots,y_{n+k},y_{n+k+1};\quad y_i=x_{ri},\quad i=0,1,2,\dots.$$

C2. Apply MPE or RRE to the sequence $\{y_i\}$ precisely as in Table 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.

For an analysis of the error in this mode of usage—in case of linear $F(x)$, i.e., when $F(x) = Tx + b$ as in (3.1)—we refer the reader to [30,31]. Note that, in this mode, both k and n are kept *fixed* (in Theorem 6.1, we let n tend to infinity), and the analyses of [30,31] take this into account and derive upper bounds on the error of $s_{n,k}$ in terms of Jacobi polynomials for certain types of spectra of the matrix T. To give the reader an idea as to the nature of these bounds, we consider those $s_{n,k}$ produced by RRE only. We also consider the case $r = 1$ for simplicity. The next theorem is from [20]; it follows directly from Eisenstat et al. [8], where it is proved for the real case.

Theorem 7.1. *Let s be the solution to (3.1) and let $\{x_n\}$ be generated linearly via (3.2). Define the residual vector associated with arbitrary x by $r(x) = b - Ax$, where $A = I - T$. Assume that the hermitian part of $\tilde{A} = \alpha A$, $\alpha \neq 0$ scalar, namely, the matrix $\tilde{A}_H = \frac{1}{2}(\tilde{A} + \tilde{A}^*)$, is positive definite. Then $s_{n,k}$ from RRE satisfies*

$$\|r(s_{n,k})\| \le L^k \|r(x_n)\|; \quad L = \sqrt{1 - \nu^2/\mu^2} < 1, \tag{7.4}$$

where ν is the smallest eigenvalue of $\tilde{A}_H$ and μ is the largest singular value of A.

Let us consider cycling with $n = 0$. Since the factor L^k in (7.4) is independent of the x_i, this theorem says that at the end of the m th cycle the l_2 norm of the residual vector will be bounded by L^{mk} times the l_2 norm of the initial residual, which means that the cycling strategy converges as $m \to \infty$ under the conditions of Theorem 7.1.

The next theorem from [30,31] is concerned with the effect of taking $n > 0$ when cycling.

Theorem 7.2. *Let s be the solution to (3.1) and let $\{x_n\}$ be generated linearly via (3.2). Assume also that the matrix T is diagonalizable so that $T = V\Lambda V^{-1}$, where $\Lambda = diag(\mu_1, \ldots, \mu_N)$, μ_i being the eigenvalues of T. Define the residual vector associated with arbitrary x by $r(x) = b - Ax$, where $A = I - T$. Let also $\tilde{P}_k = \{p \in \pi_k : p(1) = 1\}$ and $\Omega_{n,k} = \min_{p \in \tilde{P}_k} \max_i |\mu_i^n p(\mu_i)|$. Then $s_{n,k}$ from RRE satisfies*

$$\|r(s_{n,k})\| = \min_{p \in \tilde{P}_k} \|T^n p(T) r(x_0)\| \le \kappa_2(V)\, \Omega_{n,k}\, \|r(x_0)\|.$$

Here $\kappa_2(V) = \|V\|\,\|V^{-1}\|$ is the l_2 condition number of V.

The important quantity here is $\Omega_{n,k}$, which, when specialized to certain types of spectra for T, can be bounded in terms of Jacobi polynomials, and hence in closed form. Two cases follow:

1. When the spectrum of T is real and in the interval $[-\beta, \beta]$ for some $\beta \in (0, 1)$, then

$$\Omega_{n,k} \le \frac{\beta^{n+k}}{\sum_{j=0}^{\nu} \binom{\nu}{j} \binom{n+\mu}{j} (1-\beta^2)^j}; \quad \nu = \left\lfloor \frac{k}{2} \right\rfloor,$$

$$\mu = \left\lfloor \frac{k+1}{2} \right\rfloor.$$

2. When the spectrum of T is purely imaginary and in the interval $[-i\beta, i\beta]$ for some real $\beta > 0$, then

$$\Omega_{n,k} \le \frac{\beta^{n+k}}{\sum_{j=0}^{\nu} \binom{\nu}{j} \binom{n+\mu}{j} (1+\beta^2)^j}; \quad \nu = \left\lfloor \frac{k}{2} \right\rfloor,$$

$$\mu = \left\lfloor \frac{k+1}{2} \right\rfloor.$$

Analyzing and/or computing the upper bounds on $\Omega_{n,k}$, we see that, with even moderate values of n, $\Omega_{n,k}$ decreases very fast with increasing k. This explains the success of applying cycling with even moderate $n > 0$. In view of this, by taking a moderate $n > 0$, it becomes possible to apply MPE and RRE with small k, hence small storage, and still obtain good convergence rates in the cycling mode when solving $x = F(x)$.

Another advantage of applying MPE and RRE in the cycling mode with $n > 0$ is that, in some cases, it prevents stagnation that results when applying them with $n = 0$. Stagnation takes place also when applying GMRES in the restarting (cycling) mode in some cases; here recall Theorem 3.6 on the equivalence of GMRES and RRE. (See the numerical examples in [30,31].)

7.3. Applying cycling with frozen γ_j

When storage is a problem and the unitary matrix Q_{k-1} in Step 2 of the MPE and RRE algorithms given in Table 1 needs to be saved in secondary storage, the cost of cycling as described here may increase timewise on account of input-output when working with the secondary storage. We can reduce this cost of cycling substantially as follows: Save the γ_i that are computed in Step 2 of the MPE and RRE algorithms in Table 1 after the first few cycles (sometimes one cycle may be enough). In subsequent cycles, use these last (frozen) γ_i and the y_i saved in Step 1, and set $s_{n,k} = \sum_{i=0}^{k} \gamma_i y_{n+i}$ in Step 2, instead of computing $s_{n,k}$ by MPE and RRE as described in Table 1. (Of course, these $s_{n,k}$ are not the same as those computed by MPE and RRE, but are likely to be good approximations to s, nevertheless.)

This strategy has two favorable features: (i) It enables the introduction of the vectors y_j one by one without having to save them. This also means that, because the y_j need not be stored in secondary storage, we save the time needed for performing input-output operations that would be needed otherwise. (ii) It also avoids the computational (timewise) overhead of MPE and RRE on account of the QR-factorization of the matrix U_k. These savings may be significant when computation of the vectors x_n in Step C1 is inexpensive.

The idea behind this strategy is that, from one application of Steps C1 and C2 to the next, the γ_i do not change very much. This is also suggested by (7.2), which says that the $\gamma_i^{(n,k)} \equiv \gamma_i$ associated with $s_{n,k}$ obtained by MPE and RRE are such that $\lim_{n\to\infty} \gamma_i^{(n,k)}$ all exist.

8. Computation of dominant eigenvectors of matrices

One problem that can be treated efficiently by MPE and RRE is that of computing an eigenvector corresponding to the largest eigenvalue of an arbitrary large sparse matrix $A \in \mathbb{C}^{N\times N}$ when this eigenvalue is known. This problem has become of interest recently in connection with the computation of the PageRank of the Google Web matrix. It can be solved by applying vector extrapolation methods to a sequence of vectors obtained as power iterations essentially with the matrix A. See Sidi [26,27] for a detailed treatment.

The idea is as follows: Assume for simplicity that A is diagonalizable, and let μ be its largest eigenvalue, which we do

not assume to be simple necessarily. Assume also that μ is known, and that we are interested in determining a corresponding eigenvector.

Choosing an arbitrary initial vector $x_0 \in \mathbb{C}^N$, we compute the vectors $x_1, x_2, \ldots$, as power iterations with the matrix $T = \mu^{-1}A$ via

$$x_{n+1} = Tx_n, \quad n = 0, 1, \ldots. \tag{8.1}$$

Of course, $x_n = T^n x_0$, $n = 0, 1, \ldots$. Since T has 1 as its only largest eigenvalue, in general, x_n has the spectral decomposition

$$x_n = s + \sum_{i=1}^{p} v_i \lambda_i^n, \quad n = 1, 2, \ldots. \tag{8.2}$$

Here, (i) s is an eigenvector corresponding to the eigenvalue 1, that is, $Ts = s$, (ii) λ_i are some or all of the *distinct nonzero* eigenvalues of T that are different from 1 (in fact, $|\lambda_i| < 1$), (iii) for each i, v_i is an eigenvector corresponding to λ_i, whether λ_i is simple or multiple; that is, $Ts = s$, and $Tv_i = \lambda_i v_i$, $i = 1, \ldots, p$, and, of course, (iv) $p \le N - 1$. (This is the same reasoning as that which is given in the last paragraph of Section 6.) Thus, the v_i in the summation in (8.2) are linearly independent vectors, and each of the vectors $v_i\lambda_i^n$ in (8.2) is nonzero. A nonzero eigenvalue of T (in particular, the eigenvalue 1) appears in the spectral decomposition of x_n in (8.2) if it is present in the spectral decomposition of the initial vector x_0. Let us order the λ_i in (8.2) as in

$$1 > |\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_p|. \tag{8.3}$$

Thus, the vectors x_n are exactly of the form described in Theorem 6.1. This immediately suggests that MPE and RRE can be applied to the sequence $\{x_n\}$ to accelerate its convergence. Clearly, the performance of MPE and RRE on the present problem can be improved by using the cycling strategy described in Section 7.

Remarks.

1. As explained following Theorem 6.1, the convergence result pertaining to $s_{n,k}$ remains virtually the same even when the matrix A is not diagonalizable, provided the largest eigenvalue μ has only associated eigenvectors but no principal vectors. The precise convergence result for this general case that has been given in [28] is more involved, however.
2. Note that, in the problems we are treating here, $\lim_{n\to\infty} x_n$ exists and is equal to s since $|\lambda_1| < 1$, and we have stated Theorem 6.1 to suit these problems. However, Theorem 6.1 remains valid also when $\lim_{n\to\infty} x_n$ does not exist (which happens when μ is not the largest eigenvalue, in which case $|\lambda_1| \ge 1$). In this case, $\lim_{n\to\infty} s_{n,k}$ exists and equals s, provided $|\lambda_k| > |\lambda_{k+1}|$ and $|\lambda_{k+1}| < 1$.

8.1. Application to PageRank computation

It is known that the Google Web matrix has 1 as its (unique) largest eigenvalue and that the corresponding eigenvector has positive components. The PageRank, which serves as a measure of the relative importance of Web pages, is this eigenvector, normalized such that the sum of its components is 1. In this case, power iterations with the Google matrix converge to the PageRank. Here are some of the details:

We start by recalling that a matrix A is *column-stochastic* if it is nonnegative and the sum of the elements in each of its columns is 1. Such a matrix has 1 as its largest eigenvalue.

The matrix A used in the Google PageRank computations is of the form $A = cP + (1-c)E$, where P and E are very large column-stochastic matrices and $0 < c < 1$; therefore, A is column-stochastic too. In addition, E is of the form $E = ue^{\mathrm{T}}$, where $e = [1, 1, \ldots, 1]^{\mathrm{T}}$ as before, and u is a nonnegative vector such that $e^{\mathrm{T}}u = 1$. Interestingly, whether u is positive or nonnegative, the eigenvalue 1 is always simple, and the corresponding eigenvector is positive. The rest of the eigenvalues are all less than c in modulus. (See Haveliwala [12] and Eldén [9] for more information on this point.)

Vector extrapolation methods in this case turn out to be very practical as the computation of the vectors x_n is extremely cheap, due to the fact that computing the matrix-vector products Py and Ey costs only $O(N)$ operations. The reason for this is that (i) P is a very sparse matrix in that each row of the matrix P has $O(1)$ nonzero entries and (ii) $Ey = (e^{\mathrm{T}}y)u$ despite the fact that E is a dense matrix. The numerical computation of the PageRank by MPE and RRE is illustrated in [27].

The first work applying a vector extrapolation method to the sequence of power iterations is that of Kamwar et al. [15]. The method developed in [15], called *quadratic extrapolation*, is very closely related to MPE with $k = 2$. This method was subsequently generalized to arbitrary k and analyzed in Sidi [26,27], where a very economical algorithm for it is also given.

9. Application to summation of vector-valued power series: vector-valued rational approximations

Given a vector-valued power series $\sum_{i=0}^{\infty} u_i z^i$, where z is a complex variable and u_i are constant vectors in $\mathbb{C}^N$, representing a vector-valued function $u(z)$ about $z = 0$, we can use vector extrapolation methods to approximate $u(z)$ via a vector-valued rational approximation obtained from the power series coefficients u_i.

For this, we apply the extrapolation methods to the sequence $\{x_m(z)\}$, where $x_m(z) = \sum_{i=0}^{m} u_i z^i$, $m = 0, 1, \ldots$. The approximations obtained this way are rational functions, whose numerators are vector-valued polynomials and whose denominators are scalar-valued polynomials. This topic is dealt with in detail in the paper Sidi [23], where three vector-valued rational approximations are developed and discussed. Here we give a brief and informal description of the subject, through the SMPE approximations, which is one of the three approximations proposed in [23]. When MPE is used for the purpose of accelerating the convergence of the sequence of the partial sums $x_m(z)$ of the vector-valued power series $\sum_{i=0}^{\infty} u_i z^i$, we obtain the SMPE approximations $s_{n,k}(z)$ that are given as in

$$s_{n,k}(z) = \frac{D(z^k x_n(z), z^{k-1} x_{n+1}(z), \ldots, z^0 x_{n+k}(z))}{D(z^k, z^{k-1}, \ldots, z^0)}, \tag{9.1}$$

where $D(g_0, g_1, \ldots, g_k)$ is a $(k+1)\times(k+1)$ determinant defined exactly as in (5.2) with $u_{i,j} = (u_{n+i}, u_{n+j})$ there. Thus, $s_{n,k}(z)$ is also of the form

$$s_{n,k}(z) = \frac{\sum_{i=0}^{k} c_i z^{k-i} x_{n+i}(z)}{\sum_{i=0}^{k} c_i z^{k-i}}, \tag{9.2}$$

with appropriate scalar constants c_i. Note that the numerator polynomial has degree at most $n + k$, while the denominator polynomial has degree k when the cofactor of g_0 in $D(g_0, g_1, \ldots, g_k)$ is nonzero. It is easy to see that

$$s_{n,k}(z) - u(z) = O(z^{n+k+1}) \quad \text{as } z \to 0.$$

The sequence $\{s_{n,k}(z)\}_{n=0}^{\infty}$ (with fixed k) has very nice convergence properties, which we discuss briefly. Suppose that $u(z)$ is analytic in an open disc $D_r = \{z \in \mathbb{C} : |z| < r\}$ and meromorphic in a larger open disc $D_R = \{z \in \mathbb{C} : |z| < R\}$. This implies that the series $\sum_{i=0}^{\infty} u_i z^i$ converges to $u(z)$ only for $|z| < r$, and diverges for $|z| \ge r$. Under some additional condition that has to do with the Laurent

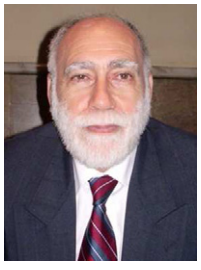
expansions of $u(z)$ about its poles, the rational approximations $s_{n,k}(z)$ obtained from the series $\sum_{i=0}^{\infty} u_i z^i$ have the property that, if k is equal to the number of the poles in D_R, then $\{s_{n,k}(z)\}_{n=0}^{\infty}$ converges to $u(z)$ *uniformly* in every compact subset of D_R excluding the poles of $u(z)$. In addition, the poles and residues of $s_{n,k}(z)$ tend to the poles and residues of $u(z)$ as $n \to \infty$.

As an example, let us consider the function $u(z) = (I - zA)^{-1} b$, where $A \in \mathbb{C}^{N\times N}$ is a constant matrix and $b \in \mathbb{C}^N$ is a constant vector. This function has the series representation $u(z) = \sum_{i=0}^{\infty} u_i z^i$, where $u_i = A^i b$ for each $i = 0, 1, \ldots$. This series converges for $|z| < 1/\rho(A)$, where, we recall, $\rho(A)$ is the spectral radius of A. In this case, $u(z)$ is a vector-valued meromorphic function (actually, a rational function) with poles equal to the reciprocals of the nonzero eigenvalues of A, the residues being related to corresponding eigenvectors and principal vectors. The poles and residues of $s_{n,k}(z)$ turn out to be the so-called Ritz values and Ritz vectors resulting from the method of Arnoldi as eigenvalue and eigenvector approximations. For details on precise convergence properties and rates of convergence, see Sidi [24].

One of the uses of these rational approximations has been to the summation of a perturbation series resulting from ODEs describing some nonlinear oscillations. See, for example, Wu and Zhong [33]. In this paper, the space we are working in is infinite dimensional, and the definitions of MPE and RRE remain unchanged, as mentioned at the end of Section 5.

References

[1] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.
[2] K.E. Atkinson, An Introduction to Numerical Analysis, Wiley, New York, 1978.
[3] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications. CMS Books in Mathematics, second edition, Springer-Verlag, New York, 2003.
[4] C. Brezinski, Généralisations de la transformation de Shanks, de la table de Padé, et de l'ϵ-algorithme, Calcolo 11 (1975) 317–360.
[5] C. Brezinski, M. Redivo Zaglia, Extrapolation Methods: Theory and Practice, North-Holland, Amsterdam, 1991.
[6] S. Cabay, L.W. Jackson, A polynomial extrapolation method for finding limits and antilimits of vector sequences, SIAM J. Numer. Anal. 13 (1976) 734–752.
[7] R.P. Eddy, Extrapolating to the limit of a vector sequence, in: P.C.C. Wang (Ed.), Information Linkage Between Applied Mathematics and Industry, Academic Press, New York, 1979, pp. 387–396.
[8] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357.
[9] L. Eldén. The eigenvalues of the Google matrix. Technical Report LiTH-MAT-R-04-01, Linköping University, 2004.
[10] W.F. Ford, A. Sidi, Recursive algorithms for vector extrapolation methods, Appl. Numer. Math. 4 (1988) 477–489.
[11] G.H. Golub, C.F. Van Loan, Matrix Computations, third edition, The Johns Hopkins University Press, London, 1996.
[12] T.H. Haveliwala, S.D. Kamvar. The Second Eigenvalue of the Google Matrix. Technical Report, Stanford University, 2003.
[13] R.A. Horn, C.R. Johnson, Matrix Analysis., Cambridge University Press, Cambridge, 1985.
[14] A.S. Householder, The Theory of Matrices in Numerical Analysis, Blaisedell, New York, 1964.
[15] S.D. Kamvar, T.H. Haveliwala, C.D. Manning, G.H. Golub, Extrapolation methods for accelerating PageRank computations, in: Proceedings of the Twelfth International World Wide Web Conference. ACM, 2003, pp. 261–270.
[16] M. Mešina, Convergence acceleration for the iterative solution of the equations $X = AX + f$, Comput. Methods Appl. Mech. Eng. 10 (1977) 165–173.
[17] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[18] D. Shanks, Nonlinear transformations of divergent and slowly convergent sequences, J. Math. Phys. 34 (1955) 1–42.
[19] A. Sidi, Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms, SIAM J. Numer. Anal. 23 (1986) 197–209 (Originally appeared as NASA TM-83443, 1983).
[20] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, J. Comp. Appl. Math. 22 (1988) 71–88.
[21] A. Sidi, Efficient implementation of minimal polynomial and reduced rank extrapolation methods, J. Comp. Appl. Math. 36 (1991) 305–337 (Originally appeared as NASA TM-103240 ICOMP-90–20, 1990).
[22] A. Sidi, Convergence of intermediate rows of minimal polynomial and reduced rank extrapolation tables, Numer. Algorithms 6 (1994) 229–244.
[23] A. Sidi, Rational approximations from power series of vector-valued meromorphic functions, J. Approx. Theory 77 (1994) 89–111.
[24] A. Sidi, Application of vector-valued rational approximation to the matrix eigenvalue problem and connections with Krylov subspace methods, SIAM J. Matrix Anal. Appl. 16 (1995) 1341–1369.
[25] A. Sidi, Practical Extrapolation Methods: Theory and Applications. Number 10 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.
[26] A. Sidi. Approximation of largest eigenpairs of matrices and applications to PageRank computation. Technical Report CS-2004-16, Computer Science Dept., Technion – Israel Institute of Technology, 2004.
[27] A. Sidi, Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations, Comp. Maths. Appl. 56 (2008) 1–24.
[28] A. Sidi, J. Bridger, Convergence and stability analyses for some vector extrapolation methods in the presence of defective iteration matrices, J. Comp. Appl. Math. 22 (1988) 35–61.
[29] A. Sidi, W.F. Ford, D.A. Smith. Acceleration of convergence of vector sequences. SIAM J. Numer. Anal. 23 (1986) 178–196 (Originally appeared as NASA TP-2193, 1983).
[30] A. Sidi, Y. Shapira, 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, 1991 (Appeared also as NASA Technical memorandum 105608, ICOMP-92-09, 1992).
[31] A. Sidi, Y. Shapira, Upper bounds for convergence rates of acceleration methods with initial iterations, Numer. Algorithms 18 (1998) 113–132.
[32] D.A. Smith, W.F. Ford, A. Sidi, Extrapolation methods for vector sequences, SIAM Rev. 29 (1987) 199–233 (Erratum: SIAM Rev. 30 (1988) 623–624).
[33] B. Wu, H. Zhong, Application of vector-valued rational approximations to a class of non-linear oscillations, Int. J. Non-Linear Mech. 38 (2003) 249–254.
[34] P. Wynn, On a device for computing the $e_m(S_n)$ transformation, Math. Tables Other Aids Comput. 10 (1956) 91–96.
[35] P. Wynn, Acceleration techniques for iterated vector and matrix problems, Math. Comput. 16 (1962) 301–322.



Avram Sidi received his Ph.D. from Tel Aviv University in 1978, and has been at the Computer Science Department of the Technion – Israel Institute of Technology, Haifa (Israel) since 1977. Presently he is a full professor and holds the Technion Administration Chair in Computer Science. He also worked for many years as a visiting scientist at NASA–Lewis Research Center, Cleveland, Ohio. One of his main areas of research is that of acceleration of convergence and extrapolation methods for scalar and vector sequences; he has been involved in the development, analysis, and algorithmic aspects, of extrapolation methods, old and new, and in their applications to computation of infinite-range integrals and the summation of infinite series of different kinds. His other areas of research are numerical integration, Pade and related scalar- and vector-valued rational approximation procedures in the complex plane, numerical linear algebra, and asymptotic methods for different problems of numerical analysis. He is the author of the 2003 book "Practical Extrapolation Methods: Theory and Applications" that covers the area of scalar extrapolation methods. He has authored and co-authored over a hundred journal papers in numerical analysis and approximation theory.