

# Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations

Avram Sidi

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

Received 19 June 2007; accepted 6 November 2007

Dedicated to the memory of Professor Gene H. Golub (1932–2007)

---

## Abstract

An important problem that arises in different areas of science and engineering is that of computing the 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  $\lim_{n \rightarrow \infty} x_n$  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  $\{x_n\}$  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 the derivation of these methods, describe the most accurate and stable algorithms for their implementation along with the effective modes of usage in solving systems of equations, nonlinear as well as linear, and present their convergence and stability theory. We also discuss their close connection with the method of Arnoldi and with GMRES, two well-known *Krylov subspace methods* for linear systems. We show that they can be used very effectively to obtain the dominant eigenvectors of large sparse matrices when the corresponding eigenvalues are known, and provide the relevant theory as well. One such problem is that of computing the *PageRank* of the Google matrix, which we discuss in detail. In addition, we show that a recent extrapolation method of Kamvar et al. that was proposed for computing the PageRank is very closely related to MPE. We present a generalization of the method of Kamvar et al. along with a very economical algorithm for this generalization. We also provide the missing convergence theory for it.

© 2007 Elsevier Ltd. All rights reserved.

**Keywords:** Vector extrapolation methods; Minimal polynomial extrapolation; Reduced rank extrapolation; Iterative methods; Krylov subspace methods; Large sparse systems of equations; Singular linear systems; Eigenvalue problems; Power iterations; Stochastic matrices; Google matrix; PageRank computations

---

## 1. Introduction

An important problem that arises in different areas of science and engineering is that of computing the 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  $\lim_{n \rightarrow \infty} x_n$  are simply the required

---

*E-mail address:* [asidi@cs.technion.ac.il](mailto:asidi@cs.technion.ac.il).

*URL:* <http://www.cs.technion.ac.il/~asidi/>.

solutions. One common source of such systems is the finite-difference or finite-element discretization of continuum problems.

Let us consider the (linear or nonlinear) system of equations

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

whose solution we denote  $s$ . Then, starting with a suitable vector  $x_0$ , an initial approximation to  $s$ , the sequence  $\{x_n\}$  can be generated by some fixed-point iterative method as in

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

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,  $\lim_{n \rightarrow \infty} x_n = s$ . One possible form of  $F(x)$  would be

$$F(x) = x + \omega A(x)\psi(x),$$

where  $A(x)$  is a nonsingular  $N \times N$  matrix and  $\omega$  is a relaxation parameter.

In most cases of interest, however, the sequences  $\{x_n\}$  converge to their limits extremely slowly; for example, when they arise from the finite-difference or finite-element discretizations of continuum problems, their rates of convergence deteriorate as the relevant mesh-sizes become smaller (hence  $N$  becomes larger). One practical way to accelerate their convergence is to apply to them suitable *vector extrapolation methods*.

A detailed review of vector extrapolation methods, containing the developments up to the early 1980s can be found in the work of Smith, Ford, and Sidi [1]. This work discusses (i) two polynomial-type methods, namely, the *minimal polynomial extrapolation* (MPE) of Cabay and Jackson [2] and the *reduced rank extrapolation* (RRE) of Eddy [3] and Mešina [4], and (ii) three epsilon algorithms, namely, the scalar and vector epsilon algorithms of Wynn [5,6], and the topological epsilon algorithm of Brezinski [7]. Numerical experience suggests that, when applied to very large systems of equations, polynomial-type methods are in general more economical than the epsilon algorithms as far as computation time and storage requirements are concerned.

In the first part of this work, we present an up-to-date review of MPE and RRE, the two polynomial-type methods. This review contains the developments that have taken place in the study of MPE and RRE since the publication of [1]. We discuss the derivation of MPE and RRE, 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 [8] and with GMRES of Saad and Schultz [9], two well-known *Krylov subspace methods* for solving linear systems.

When applied to the sequence of partial sums of a vector-valued power series of a vector-valued meromorphic function, MPE serves as a tool for analytic continuation: it produces vector-valued rational approximations to the function in question that approximate it even outside the circle of convergence of its power series. These rational approximations are closely related to the Arnoldi method for eigenvalue approximations. This topic, which we do not go into here, is discussed at length in Sidi [10,11].

Following the summary of MPE and RRE, we discuss their application to the solution of systems of equations, and we describe a strategy that is known as *cycling or restarting*. We also describe some extensions of cycling that have proved to be effective in that they reduce the time and/or storage overhead involved in the application of MPE and RRE.

In the second part of this work, we consider the computation of an eigenvector of an arbitrary large and sparse matrix, corresponding to its largest eigenvalue when this eigenvalue is known. In addition to being of importance in its own right, this problem has attracted much attention recently because it also arises in the computation of the *PageRank* of the Google Web matrix. We show that vector extrapolation methods, especially MPE and RRE, can be used effectively in the solution of this problem and hence in PageRank computations. In addition, drawing on some earlier results concerning Drazin-inverse solutions of consistent singular linear systems, we propose and justify theoretically the use of standard Krylov subspace methods, such as the method of Arnoldi and GMRES, for solving this problem. We present a polynomial preconditioner to improve the performance of these methods as well.

Turning next to the computation of the PageRank, we study an extrapolation method that was proposed recently by Kamvar et al. [12] and denoted *quadratic extrapolation* there. We generalize this method, and prove that the resulting

generalization is very closely related to MPE, and also state a convergence theorem for it. In addition, we provide an algorithm that is twice as economical as that of [12] storagewise.

The contents of this part of the paper on computing eigenvectors in general and the PageRank in particular are new and originally appeared in the author’s 2004 technical report [13].

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 and used 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.

Finally, we mention that MPE and RRE have been used as effective accelerators for solving large and sparse nonlinear systems of equations that arise in diverse areas of sciences and engineering, such as computational fluid dynamics, structures, materials, semiconductor research, computerized tomography, and computer vision, to name a few. As we do not intend to give even a partial list of these applications here, we refer the reader to the relevant literature for them.

Note that, throughout this work,  $\|z\|$  with  $z \in \mathbb{C}^s$  will stand for the standard vector  $l_2$ -norm of  $z$ , namely,  $\|z\| = \sqrt{z^*z}$ , and  $\|A\|$  with  $A \in \mathbb{C}^{s \times s}$  will stand for the matrix norm induced by this vector norm.

## 2. Theoretical preliminaries

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 the iterative method of (1.2), the function  $F(x)$  there being nonlinear in general. Assuming that  $\lim_{n \rightarrow \infty} x_n$  exists, hence  $x_n \approx s$  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

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

Here,  $F'(s)$  is the  $N \times N$  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  $A$ . The result in (2.1) can be also expressed as in

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

$T$  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 its solution  $s$ . This then suggests that we should look at linear systems for deriving vector extrapolation methods. We aim at developing methods whose only input is the vector sequence  $\{x_n\}$  generated by a black box.

## 3. Derivation of MPE and RRE

The treatment we give in this section follows in part that of Smith, Ford, and Sidi [1]. (For a completely different approach that goes through the transformation of Shanks [14], we refer the reader to the paper by Sidi, Ford, and Smith [15].)

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

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

Let us also define

$$e_n = x_n - s, \quad u_n = \Delta x_n, \quad w_n = \Delta u_n = \Delta^2 x_n, \quad n = 0, 1, \dots \quad (3.3)$$

Here,  $\Delta z_n = z_{n+1} - z_n$  and  $\Delta^2 z_n = \Delta(\Delta z_n) = z_{n+2} - 2z_{n+1} + z_n$ . It is easy to see that

$$e_n = T^n e_0, \quad u_n = T^n u_0, \quad w_n = T^n w_0, \quad n = 0, 1, \dots \quad (3.4)$$

In addition,

$$u_n = (T - I)e_n, \quad n = 0, 1, \dots \quad (3.5)$$

**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.

We then have the following known result:

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

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  $e_0 = x_0 - s$ . (Also notice that, by [Theorem 3.2](#),  $k = \deg P \leq N$ .) Thus,  $P(T)e_0 = 0$ , which, by (3.3) and (3.4), also means that

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

From this, we have

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

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 (i)  $P(\lambda)$  divides the characteristic polynomial of  $T$ , and (ii) 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  $x_0, x_1, \dots, x_k$ , 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 believe that, in order to know  $P(\lambda)$ , we must know  $s$ . Fortunately, this is not the case; as we show in the next theorem,  $P(\lambda)$  is also the minimal polynomial of  $T$  with respect to  $u_0 = x_1 - x_0$ , and  $u_0$  is known.

**Theorem 3.3.** *Let  $P(\lambda)$  and  $Q(\lambda)$  be the minimal polynomials of  $T$  with respect to  $e_0$  and  $u_0$ , respectively. Then  $P(\lambda) \equiv Q(\lambda)$ .*

**Proof.** By the fact that  $P(T)e_0 = 0$  and by (3.5), we have that

$$0 = (T - I)P(T)e_0 = P(T)(T - I)e_0 = P(T)u_0.$$

Thus, by part (ii) of [Theorem 3.2](#), we have that  $Q(\lambda)$  divides  $P(\lambda)$ . Next, by the fact that  $Q(T)u_0 = 0$  and by (3.5), we have

$$0 = Q(T)u_0 = Q(T)(T - I)e_0 = (T - I)Q(T)e_0.$$

Now invoking the fact that  $(T - I)$  is a nonsingular matrix because 1 is not an eigenvalue of  $T$ , this last relation is equivalent to  $Q(T)e_0 = 0$ , which, by part (ii) of [Theorem 3.2](#), implies that  $P(\lambda)$  divides  $Q(\lambda)$ . Since both  $P(\lambda)$  and  $Q(\lambda)$  are monic, it is now clear that  $P(\lambda) \equiv Q(\lambda)$ . ■

The fact stated in [Theorem 3.3](#) enables us to determine the coefficients of  $P(\lambda)$ , namely,  $c_0, c_1, \dots, c_{k-1}$ , again from our knowledge of the  $x_j$ . (Recall that  $P(\lambda)$  is monic, that is,  $c_k = 1$ .) First, by (3.4),  $P(T)u_0 = 0$  can be

re-expressed as in

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

What we have here is a possibly overdetermined but consistent set of linear equations for the  $c_j$  that is of the form

$$Uc = -u_k; \quad c = [c_0, c_1, \dots, c_{k-1}]^T \in \mathbb{C}^k, \quad U = [u_0 \mid u_1 \mid \dots \mid u_{k-1}] \in \mathbb{C}^{N \times k}. \tag{3.8}$$

Note that the matrix  $U$  has full rank, that is,  $\text{rank}(U) = k$ , because its columns, namely, the vectors  $u_0, u_1, \dots, u_{k-1}$ , 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  $Uc = -u_k$  has a unique solution given by

$$c = -U^+ u_k, \tag{3.9}$$

where  $U^+$  is the Moore–Penrose generalized inverse of  $U$ , which, in this case, is given by  $U^+ = (U^*U)^{-1}U^*$ . (Note that this is in complete agreement with the fact that  $P(\lambda)$  is unique.) Clearly, this solution for  $c$  is also the solution we would obtain by solving the linear system  $Uc = -u_k$  by linear least-squares. It is these observations that will lead us to the definition of MPE and, following that, to the definition of RRE.

By the developments above, it is clear that  $s$  can be constructed from the knowledge of the  $k + 2$  vectors  $x_j$ ,  $j = 0, 1, \dots, k + 1$ , nothing else being needed. We have also shown that  $s$  is some sort of “weighted average” of the vectors  $x_0, x_1, \dots, x_k$ ; that is, it is of the form

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

Here, of course,

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

### 3.1. Definition of MPE

As is clear, the developments above leading to the solution  $s$  given in (3.7) 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. 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 system  $Uc = -u_k$  in (3.8) will be inconsistent, hence will not have a solution in the ordinary sense. We can, however, *define*  $c$  as the least-squares solution of the system  $Uc = -u_k$ , leaving everything else unchanged. This results in the following definition:

**Definition 3.4.** Let  $\{x_n\}$  be a given sequence in  $\mathbb{C}^N$ , and let the vectors  $u_n$  be as in (3.3). Choose  $k \leq N$  arbitrarily and define the matrix  $U$  via

$$U = [u_0 \mid u_1 \mid \dots \mid u_{k-1}] \in \mathbb{C}^{N \times k}.$$

Let  $c$  be the least-squares solution to the linear system  $Uc = -u_k$ ; this means that  $c$  is the solution to the problem

$$\min_{c_0, c_1, \dots, c_{k-1}} \left\| \sum_{j=0}^{k-1} c_j u_j + u_k \right\|. \tag{3.10}$$

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

$$\gamma_j = \frac{c_j}{\alpha}, \quad j = 0, 1, \dots, k; \quad \alpha = \sum_{i=0}^k c_i, \tag{3.11}$$

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

$$s_k^{\text{MPE}} = \sum_{j=0}^k \gamma_j x_j. \quad (3.12)$$

### 3.2. Definition of RRE

Going back to the situation in which  $k$  is the degree of the minimal polynomial of  $T$  with respect to  $u_0$ , we first observe that, by dividing the equations  $Uc = -u_k$  in (3.8) by  $\sum_{i=0}^k c_i$ , denoting  $\gamma_j = c_j / \sum_{i=0}^k c_i$ ,  $j = 0, 1, \dots, k$ , and observing that  $\sum_{j=0}^k \gamma_j = 1$ , we can rewrite (3.8) as in

$$U'\gamma = 0 \quad \text{and} \quad \sum_{j=0}^k \gamma_j = 1; \\ \gamma = [\gamma_0, \gamma_1, \dots, \gamma_k]^T \in \mathbb{C}^{k+1}, \quad U' = [u_0 \mid u_1 \mid \dots \mid u_k] \in \mathbb{C}^{N \times (k+1)}. \quad (3.13)$$

As before, a unique solution to this system for  $\gamma$  exists in case  $k$  is the degree of the minimal polynomial of  $T$  with respect to  $u_0$ . When  $k$  is an arbitrary integer smaller than this degree, there is no solution to the system  $U'\gamma = 0$  and  $\sum_{j=0}^k \gamma_j = 1$  in the ordinary sense. We can, however, *define*  $\gamma$  to be the least-squares solution of the system  $U'\gamma = 0$  subject to the constraint  $\sum_{j=0}^k \gamma_j = 1$ . This results in the following definition:

**Definition 3.5.** Let  $\{x_n\}$  be a given sequence in  $\mathbb{C}^N$ , and let the vectors  $u_n$  be as in (3.3). Choose  $k \leq N$  arbitrarily and define the matrix  $U'$  via

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

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

$$\min_{\gamma_0, \gamma_1, \dots, \gamma_k} \left\| \sum_{j=0}^k \gamma_j u_j \right\|, \quad \text{subject to} \quad \sum_{j=0}^k \gamma_j = 1. \quad (3.14)$$

Then,  $s_k^{\text{RRE}}$ , the RRE approximation to  $\lim_{n \rightarrow \infty} x_n = s$ , is given by

$$s_k^{\text{RRE}} = \sum_{j=0}^k \gamma_j x_j. \quad (3.15)$$

Note that the way we have approached the definition of RRE here is not the only way possible, and it differs from that of [1] and those of the original works [3,4]. (The approaches of [3,4] differ from each other greatly, and their equivalence is established in [1].) Our approach here is the most direct, however. The definition of RRE here is also that given in [4], and turns out to be very suitable for computational purposes. For a completely different approach, see Sidi, Ford, and Smith [15].

For completeness, here we reproduce the definition of [3]:

$$s_k^{\text{RRE}} = x_0 + \sum_{i=0}^{k-1} \xi_i u_i, \quad (3.16)$$

where the  $\xi_i$  are determined from the least-squares solution of the linear system

$$W\xi = -u_0, \quad (3.17)$$

where

$$\xi = [\xi_0, \xi_1, \dots, \xi_{k-1}]^T \in \mathbb{C}^k, \quad W = [w_0 \mid w_1 \mid \dots \mid w_{k-1}] \in \mathbb{C}^{N \times k}. \quad (3.18)$$

Here, the  $w_i$ , just as the  $u_i$ , are as defined in (3.3). Thus,

$$s_k^{\text{RRE}} = x_0 - UW^+u_0, \tag{3.19}$$

with  $U = [u_0 \mid u_1 \mid \dots \mid u_{k-1}] \in \mathbb{C}^{N \times k}$ , as before.

#### 4. Determinant representations

The least-squares solutions for the vectors  $c$  and  $\gamma$  in Definitions 3.4 and 3.5, respectively, give rise to normal equations. Being linear systems, these normal equations can be manipulated suitably to enable us to express the MPE and RRE approximations  $s_k^{\text{MPE}}$  and  $s_k^{\text{RRE}}$  as ratios of determinants.

Instead of applying MPE and RRE to the  $k + 2$  vectors  $x_0, x_1, \dots, x_{k+1}$ , we now propose to apply them to the  $k + 2$  vectors  $x_n, x_{n+1}, \dots, x_{n+k+1}$  with some positive integer  $n$ . This usage of MPE and RRE has some very beneficial effect, as we will discuss in the next sections. Let us denote the approximations obtained this way by  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$ . Thus,  $s_{0,k}^{\text{MPE}} = s_k^{\text{MPE}}$  and  $s_{0,k}^{\text{RRE}} = s_k^{\text{RRE}}$ .

The determinant representations for the approximations  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$  given in the next theorem were presented originally in Sidi [16].

**Theorem 4.1.** *The approximations  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$  have the following determinant representations:*

$$s_{n,k} = \frac{D(x_n, x_{n+1}, \dots, x_{n+k})}{D(1, 1, \dots, 1)}, \tag{4.1}$$

where  $D(z_0, z_1, \dots, z_k)$  is a  $(k + 1) \times (k + 1)$  determinant defined as in

$$D(z_0, z_1, \dots, z_k) = \begin{vmatrix} z_0 & z_1 & \dots & z_k \\ u_{0,0} & u_{0,1} & \dots & u_{0,k} \\ u_{1,0} & u_{1,1} & \dots & u_{1,k} \\ \vdots & \vdots & \dots & \vdots \\ u_{k-1,0} & u_{k-1,1} & \dots & u_{k-1,k} \end{vmatrix}, \tag{4.2}$$

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

$$u_{i,j} = \begin{cases} (u_{n+i}, u_{n+j}) = (\Delta x_{n+i}, \Delta x_{n+j}) & \text{for MPE} \\ (w_{n+i}, u_{n+j}) = (\Delta^2 x_{n+i}, \Delta x_{n+j}) & \text{for RRE} \end{cases}. \tag{4.3}$$

Here  $u_n$  and  $w_n$  are as in (3.3) and  $(a, b)$  is simply the Euclidean inner product in  $\mathbb{C}^N$ , that is,  $(a, b) = a^*b$ .

Note that, in case the  $z_j$  in  $D(z_0, z_1, \dots, z_k)$  are vectors, this determinant is vector-valued and is defined to be  $\sum_{j=0}^k M_j z_j$ , where  $M_j$  is the cofactor of  $z_j$ .

The determinant representations above have been used in obtaining the convergence and stability results of Theorem 5.2 in the next section. They have also been used by Ford and Sidi [17] to obtain recursion relations among the different  $s_{n,k}$ . We skip these recursions here and refer the reader to [17].

#### 5. Analytic properties of 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, 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 expect MPE and RRE to be effective in accelerating the convergence of vector sequences  $\{x_n\}$  generated by fixed-point methods on nonlinear systems as well.

Since the least-squares problem for the  $\gamma_j$  always has a solution for RRE, we conclude that  $s_k^{\text{RRE}}$  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  $s_k^{\text{MPE}}$  may not always exist. A sufficient condition for the existence of  $s_k^{\text{MPE}}$  is given in Theorem 5.2.

When applied to sequences  $\{x_n\}$  generated by fixed-point iterative techniques for linear systems, MPE and RRE become equivalent to some well-known Krylov subspace methods. This is the subject of the following theorem that was proved in Sidi [18]:

**Theorem 5.1.** Consider the linear system in (3.1) and the vector sequence  $\{x_n\}$  generated as in (3.2). Then,  $s_k^{\text{MPE}}$  and  $s_k^{\text{RRE}}$  are the same vectors generated by  $k$  steps of the method of Arnoldi and  $k$  steps of GMRES, respectively, as the latter two methods are applied to the system  $(I - T)x = b$ , starting with the vector  $x_0$ .

The following theorem by Sidi [16] explains why MPE and RRE are true convergence acceleration methods. To this effect, we apply these methods to the vectors  $x_n, x_{n+1}, \dots, x_{n+k+1}$  with  $n > 0$ , instead of applying them to the vectors  $x_0, x_1, \dots, x_{k+1}$ . In the preceding section, we denoted the approximations obtained in this way by  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$ , respectively. We then have the following result:

**Theorem 5.2.** Let the vector sequence  $\{x_n\}$  be such that

$$x_n = s + \sum_{i=1}^p v_i \lambda_i^n, \quad (5.1)$$

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

$$\lambda_i \neq 1 \quad \text{for all } i, \quad (5.2)$$

and are ordered such that

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p|. \quad (5.3)$$

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

$$|\lambda_k| > |\lambda_{k+1}|. \quad (5.4)$$

Then,  $s_{n,k}^{\text{RRE}}$  exists for all  $n$  and  $s_{n,k}^{\text{MPE}}$  exists for all large  $n$ , and

$$\begin{aligned} s_{n,k} - s &= [C_{n,k} + o(1)] \lambda_{k+1}^n \quad \text{as } n \rightarrow \infty, \\ &= O(\lambda_{k+1}^n) \quad \text{as } n \rightarrow \infty, \end{aligned} \quad (5.5)$$

where  $s_{n,k}$  stands for both  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$ , and the vectors  $C_{n,k}$  are uniformly bounded in  $n$ , that is,  $\sup_n \|C_{n,k}\| < \infty$ .

In addition,  $C_{n,k}^{\text{MPE}} = C_{n,k}^{\text{RRE}}$ , so that

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

Let us denote  $\gamma_j$  in (3.12) and in (3.15) by  $\gamma_j^{(n,k)}$ . Then,  $\lim_{n \rightarrow \infty} \gamma_j^{(n,k)}$  all exist, and

$$\lim_{n \rightarrow \infty} \sum_{j=0}^k \gamma_j^{(n,k)} \lambda^j = \prod_{i=1}^k \frac{\lambda - \lambda_i}{1 - \lambda_i}. \quad (5.6)$$

When  $k = p$ , we have the so-called finite termination property:

$$s_{n,p} = s \quad \text{and} \quad \sum_{j=0}^p \gamma_j^{(n,p)} \lambda^j = \prod_{i=1}^p \frac{\lambda - \lambda_i}{1 - \lambda_i}. \quad (5.7)$$



- Remarks.** 1. Since  $x_n - s = O(\lambda_1^n)$  and  $s_{n,k} - s = O(\lambda_{k+1}^n)$  as  $n \rightarrow \infty$ , and  $|\lambda_1| \geq |\lambda_k| > |\lambda_{k+1}|$ , it is clear that  $\{s_{n,k}\}_{n=0}^\infty$  tends to  $s$  faster than  $\{x_n\}$  when  $\{x_n\}$  converges. In case  $\{x_n\}$  does not converge (which happens when  $|\lambda_1| \geq 1$ ),  $\{s_{n,k}\}_{n=0}^\infty$  will converge if  $|\lambda_{k+1}| < 1$ , and it will diverge (but less slowly than  $\{x_n\}$ ) if  $|\lambda_{k+1}| \geq 1$ . Thus, MPE and RRE accelerate the convergence of sequences  $\{x_n\}$  that satisfy the conditions of [Theorem 5.2](#). Thus, in order for MPE and RRE (and the epsilon algorithms mentioned in Section 1) to be effective, convergence of the sequence  $\{x_n\}$  is not necessary. (In case of divergence we call  $s$  the *antilimit* of  $\{x_n\}$ .)
2. Vector sequences  $\{x_n\}$  satisfying the conditions of [Theorem 5.2](#) arise from the iterative technique of (3.2) when the matrix  $T$  is diagonalizable and  $(I - T)$  is nonsingular; in this case, all eigenvalues of the matrix  $T$  are different from 1, the scalars  $\lambda_1, \dots, \lambda_p$  are some or all of the distinct nonzero eigenvalues of  $T$ , and the vectors  $v_i$  are corresponding eigenvectors, that is,  $Tv_i = \lambda_i v_i$ ,  $i = 1, \dots, p$ , and, of course,  $p \leq N$ . Clearly,  $v_i$  are linearly independent. When  $T$  is not diagonalizable, the structure of  $x_n$  becomes more complicated, and so do the results of [Theorem 5.2](#). This general case has been treated completely in Sidi and Bridger [19].
3. As we will see later in this work, when a diagonalizable matrix  $B$  has a largest eigenvalue equal to 1, the power iterations with  $B$  satisfy (5.1) with  $s$  there being an eigenvector corresponding to this eigenvalue. This is precisely the situation in case  $B$  is the Google matrix, and forms the basis for the eigenvalue methods for computing the PageRank.
4. The result of [Theorem 5.2](#) remains the same when MPE and RRE are applied to vector sequences  $\{x_n\}$  in infinite-dimensional inner product spaces such that

$$x_n \sim s + \sum_{i=1}^{\infty} v_i \lambda_i^n \quad \text{as } n \rightarrow \infty,$$

where  $\lambda_i$  are distinct and nonzero, satisfy (5.2), and

$$|\lambda_1| \geq |\lambda_2| \geq \dots \quad \text{and} \quad \lim_{i \rightarrow \infty} \lambda_i = 0$$

instead of (5.3), and there can be only a finite number of  $\lambda_i$  having the same modulus.

Note that [Theorem 5.2](#) does not cover the case  $|\lambda_k| = |\lambda_{k+1}|$ . By different techniques it can be shown that, when  $|\lambda_k| = |\lambda_{k+1}|$ , the convergence result of [Theorem 5.2](#) pertaining to  $s_{n,k}$ , namely,  $s_{n,k} - s = O(\lambda_{k+1}^n)$  as  $n \rightarrow \infty$ , remains essentially the same, provided the sequence  $\{x_n\}$  results from the iterative scheme in (3.2). The relevant result for this case is the subject of the next theorem that was proved originally in Sidi [20].

**Theorem 5.3.** *Let the vector sequence  $\{x_n\}$  be precisely as in [Theorem 5.2](#) with the notation therein; assume only that*

$$|\lambda_k| = |\lambda_{k+1}|,$$

*instead of (5.4). Assume, in addition, that the  $x_n$  have been obtained via  $x_{n+1} = Tx_n + b$ ,  $n = 0, 1, \dots$ , where the matrix  $I - T$  is nonsingular, and that  $s$  is the unique solution to the linear system  $(I - T)x = b$ . Then,*

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

*(i) for RRE unconditionally, and (ii) for MPE provided the eigenvalues of the matrix  $A = I - T$  lie strictly on one side of a straight line through the origin in the complex plane, which happens when  $A + A^*$  is positive definite, for example.*

So far, we have reviewed the convergence properties of the approximations  $s_{n,k}$  for  $n \rightarrow \infty$  while  $k$  is being held fixed. In practice, of course,  $n$  does not tend to infinity. In addition, to avoid excessive storage requirements that may arise on computers on account of increasing  $k$ , MPE and RRE are applied in the so-called *cycling* mode (to be discussed later in Section 7), where  $n$  and  $k$  can be moderately large but are both kept fixed.

The next result, due to Sidi and Shapira [21,22], provides error bounds on  $s_{n,k}^{\text{RRE}}$  for the case in which both  $n$  and  $k$  are being kept fixed. This theorem actually gives the justification for cycling with  $s_{n,k}$  with even moderate *positive*  $n$  rather than fixing  $n = 0$ .

**Theorem 5.4.** *Let  $s$  be the solution to the linear system  $(I - T)x = b$ , where  $I - T$  is nonsingular. Assume that  $T$  is diagonalizable, that is,*

$$T = R\Lambda R^{-1}; \quad \Lambda = \text{diag}(\mu_1, \mu_2, \dots, \mu_N),$$

and let the vector sequence  $\{x_n\}$  be generated via  $x_{n+1} = Tx_n + b, n = 0, 1, \dots$ , precisely as in Theorem 5.2. Define the residual vector associated with the vector  $x$  by  $r(x) = b - (I - T)x$ . Also let

$$\mathcal{P}_k = \{p(z) : p \in \Pi_k, p(1) = 1\}.$$

Then

$$\begin{aligned} \|r(s_{n,k}^{\text{RRE}})\| &= \min_{p \in \mathcal{P}_k} \|T^n p(T)r(x_0)\| \\ &\leq \|T^n p(T)\| \|r(x_0)\| \quad \text{for every } p \in \mathcal{P}_k, \\ &\leq \kappa(R) \Gamma_{n,k}^{\text{spect}(T)} \|r(x_0)\|, \end{aligned}$$

where  $\kappa(R) = \|R\| \|R^{-1}\|$  stands for the condition number of  $R$ ,  $\text{spect}(T)$  stands for the spectrum of  $T$ , and, for any set  $D$  in the complex plane,

$$\Gamma_{n,k}^D = \min_{p \in \mathcal{P}_k} \max_{z \in D} |z^n p(z)|.$$

**Remark.** Theorem 5.4 was originally given in [21], where a similar statement can also be found for MPE. The paper [22] actually treats (restarted) GMRES( $k$ ) with  $n$  initial iterations of Richardson type, which is denoted there GMRES( $n, k$ ). Due to the fact that RRE and GMRES are equivalent in the sense described in Theorem 5.1, the results of [22] can be expressed as in Theorem 5.4 and without any changes.

In [21], upper and lower bounds on  $\Gamma_{n,k}^{\text{spect}(T)}$  that can be expressed analytically in terms of suitable sets of orthogonal polynomials are given. It is also shown there that, for some special types of spectra, both of the bounds can be expressed analytically in terms of the Jacobi polynomials  $P_k^{(\alpha,\beta)}(z)$  and can easily be computed numerically. In addition, these bounds are quite tight. (See the tables in [21,22].) Below, by  $[u, v]$  we mean the straight line segment between the (complex) numbers  $u$  and  $v$  in the  $z$ -plane. The Jacobi polynomials  $P_k^{(\alpha,\beta)}(z)$  are normalized such that  $P_k^{(\alpha,\beta)}(1) = \binom{k+\alpha}{k}$ ; see Abramowitz and Stegun [23], for example.

1. If  $\text{spect}(T)$  is contained in  $[0, \beta]$  for some possibly complex  $\beta$  ( $1$  is not in  $[0, \beta]$ ), then

$$\Gamma_{n,k}^{\text{spect}(T)} \leq \frac{|\beta|^n}{\left|P_k^{(0,2n)}(2/\beta - 1)\right|} = \frac{|\beta|^{n+k}}{\left|\sum_{j=0}^k \binom{k}{j} \binom{2n+k}{j} (1-\beta)^j\right|}.$$

This bound is a decreasing function of both  $n$  and  $k$  when  $\beta \neq 0$  is real and  $\beta < 1$ . In particular, its decrease as a function of  $k$  becomes faster with increasing  $n$ . Clearly, this bound decreases very quickly in case the spectrum of  $T$  is real nonpositive, that is,  $\beta < 0$ .

2. If  $\text{spect}(T)$  is contained in  $[-\beta, \beta]$  for some possibly complex  $\beta$  ( $1$  is not in  $[-\beta, \beta]$ ), then

$$\Gamma_{n,2v}^{\text{spect}(T)} \leq \frac{|\beta|^n}{\left|P_v^{(0,n)}(2/\beta^2 - 1)\right|}, \quad \Gamma_{n,2v+1}^{\text{spect}(T)} \leq \frac{|\beta|^{n+1}}{\left|P_v^{(0,n+1)}(2/\beta^2 - 1)\right|}.$$

Both for even and odd values of  $k$ , these upper bounds can be unified to read

$$\Gamma_{n,k}^{\text{spect}(T)} \leq \frac{|\beta|^{n+k}}{\left|\sum_{j=0}^v \binom{v}{j} \binom{n+\mu}{j} (1-\beta^2)^j\right|},$$

where

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

This unified bound is a decreasing function of both  $n$  and  $k$  when  $\beta$  is real and  $|\beta| < 1$ . The same holds in case  $\beta$  is purely imaginary and  $|\beta| < 1$ , which happens when  $T$  is a skew-hermitian matrix, for example. In this case, the

upper bound on  $\Gamma_{n,k}^{\text{spect}(T)}$  tends to zero as a function of  $k$  with increasing  $n$  at the best possible rate, because now

$$\Gamma_{n,k}^{\text{spect}(T)} \leq \frac{|\beta|^{n+k}}{\sum_{j=0}^v \binom{v}{j} \binom{n+\mu}{j} (1+|\beta|^2)^j}.$$

When comparing two approximations  $s_{n,k}^{\text{RRE}}$  and  $s_{n',k}^{\text{RRE}}$ , what seems to determine which one is better is the corresponding upper bounds for  $\Gamma_{n,k}^{\text{spect}(T)}$  and  $\Gamma_{n',k}^{\text{spect}(T)}$ . At least from the examples we have given here, it can be concluded that  $s_{n,k}^{\text{RRE}}$  is the better one if  $n > n'$ . In particular,  $s_{n,k}^{\text{RRE}}$  is likely to be better than  $s_{0,k}^{\text{RRE}}$ . We can make use of this observation in applying RRE (and MPE as well) in the cycling mode; this will be discussed in Section 7.

### 6. 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 algorithms would be those that use the normal equations (i)  $U^*Uc = -U^*u_k$  in Definition 3.4 to determine the  $c_i$  for MPE, and (ii)  $W^*W\xi = -W^*u_0$  in (3.17) to determine the  $\xi_i$  for RRE. Since the matrices  $U$  and  $W$  become very ill-conditioned with increasing  $k$ , these algorithms are not stable numerically, and they produce  $s_{n,k}$  with reduced accuracy in floating-point arithmetic. Concerning the solution of least-squares problems via normal equations, see Golub and Van Loan [24]. In this connection, we should also warn the reader that the determinantal expressions given in Theorem 4.1 should not be used for this purpose as they come directly from the normal equations.

Numerically fast and stable and storagewise economical algorithms have been given in Sidi [25], 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.

We switch to the computation of  $s_{n,k}^{\text{MPE}}$  and  $s_{n,k}^{\text{RRE}}$  with  $n \geq 0$ . To discuss the algorithms conveniently, we introduce the notation

$$U_j = [u_n \mid u_{n+1} \mid \cdots \mid u_{n+j}], \quad j = 0, 1, \dots$$

Let us assume that  $U_j$  has full rank, namely,  $\text{rank}(U_j) = j + 1$ . Then, it has a QR-factorization  $U_j = Q_j R_j$ , where  $Q_j \in \mathbb{C}^{N \times (j+1)}$  is unitary and  $R_j \in \mathbb{C}^{(j+1) \times (j+1)}$  is upper triangular with positive diagonal entries,

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

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

Also,  $Q_j$  is obtained from  $Q_{j-1}$  by appending one column (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  $([r_{0j}, r_{1j}, \dots, r_{jj}]^T)$  to the end of the latter. The details of the algorithms are summarized in Tables 1 and 2. Note that, in these algorithms, we need to store only the vector  $x_n$  and the matrix  $Q_k$ , namely, the vectors  $q_0, q_1, \dots, q_k$ . The rest can be overwritten. (Actually,  $q_k$  need not be computed as it is not needed for determining  $s_{n,k}$ .)

The QR-factorization can be carried out inexpensively by applying the modified Gram–Schmidt process (MGS) to the vectors  $u_n, u_{n+1}, \dots, u_{n+j}$ ; see [25]. (For MGS and other orthogonalization processes, see Golub and Van Loan [24].) For the sake of completeness, we reproduce here the steps of MGS. For simplicity of notation, we replace the vectors  $u_{n+i}$  by  $u_i$ .

#### MGS algorithm

Step 1. Compute  $r_{00} = (u_0, u_0)^{1/2}$  and set  $q_0 = u_0/r_{00}$ .

Table 1  
Algorithm for MPE

Step 0.	Input: The vectors $x_n, x_{n+1}, \dots, x_{n+k+1}$ .
Step 1.	<p>Compute <math>u_i = \Delta x_i = x_{i+1} - x_i, i = n, n+1, \dots, n+k</math>.</p> <p>Set <math>U_j = [u_n   u_{n+1}   \dots   u_{n+j}] \in \mathbb{C}^{N \times (j+1)}, j = 0, 1, \dots</math></p> <p>Compute the QR-factorization of <math>U_k</math>, namely, <math>U_k = Q_k R_k</math>.</p> <p>(<math>U_{k-1} = Q_{k-1} R_{k-1}</math> is contained in <math>U_k = Q_k R_k</math>.)</p>
Step 2.	<p>Solve the (upper triangular) linear system</p> $R_{k-1} c = -\rho_k; \quad \rho_k = [r_{0k}, r_{1k}, \dots, r_{k-1,k}]^T, \quad c = [c_0, c_1, \dots, c_{k-1}]^T.$ <p>(Note that <math>\rho_k = Q_{k-1}^* u_{n+k}</math>.)</p> <p>Set <math>c_k = 1</math> and compute <math>\alpha = \sum_{i=0}^k c_i</math>.</p> <p>Set <math>\gamma_i = c_i / \alpha, i = 0, 1, \dots, k</math>.</p>
Step 3.	<p>Compute <math>\xi = [\xi_0, \xi_1, \dots, \xi_{k-1}]^T</math> by</p> $\xi_0 = 1 - \gamma_0; \quad \xi_j = \xi_{j-1} - \gamma_j, \quad j = 1, \dots, k-1.$ <p>Compute <math>s_{n,k}^{\text{MPE}}</math> via</p> $s_{n,k}^{\text{MPE}} = x_n + Q_{k-1} (R_{k-1} \xi).$

Table 2  
Algorithm for RRE

Step 0.	Input: The vectors $x_n, x_{n+1}, \dots, x_{n+k+1}$ .
Step 1.	<p>Compute <math>u_i = \Delta x_i = x_{i+1} - x_i, i = n, n+1, \dots, n+k</math>.</p> <p>Set <math>U_j = [u_n   u_{n+1}   \dots   u_{n+j}] \in \mathbb{C}^{N \times (j+1)}, j = 0, 1, \dots</math></p> <p>Compute the QR-factorization of <math>U_k</math>, namely, <math>U_k = Q_k R_k</math>.</p> <p>(<math>U_{k-1} = Q_{k-1} R_{k-1}</math> is contained in <math>U_k = Q_k R_k</math>.)</p>
Step 2.	<p>Solve the linear system</p> $R_k^* R_k d = e; \quad d = [d_0, d_1, \dots, d_k]^T,$ $e = [1, 1, \dots, 1]^T \in \mathbb{C}^{k+1}.$ <p>(This amounts to solving two triangular (lower and upper) systems.)</p> <p>Set <math>\lambda = \left(\sum_{i=0}^k d_i\right)^{-1}</math>. (Note that <math>\lambda</math> is real and positive.)</p> <p>Set <math>\gamma = \lambda d</math>, that is, <math>\gamma_i = \lambda d_i, i = 0, 1, \dots, k</math>.</p>
Step 3.	<p>Compute <math>\xi = [\xi_0, \xi_1, \dots, \xi_{k-1}]^T</math> by</p> $\xi_0 = 1 - \gamma_0; \quad \xi_j = \xi_{j-1} - \gamma_j, \quad j = 1, \dots, k-1.$ <p>Compute <math>s_{n,k}^{\text{RRE}}</math> via</p> $s_{n,k}^{\text{RRE}} = x_n + Q_{k-1} (R_{k-1} \xi).$

Step 2. For  $k = 1, 2, \dots$ , do

Set  $u_k^{(0)} = u_k$

For  $j = 0$  to  $k-1$  do

Compute  $r_{j,k} = (q_j, u_k^{(j)})$  and  $u_k^{(j+1)} = u_k^{(j)} - r_{jk} q_j$ .

end for( $j$ )

Compute  $r_{kk} = (u_k^{(k)}, u_k^{(k)})^{1/2}$  and  $q_k = u_k^{(k)} / r_{kk}$ .

end for( $k$ )

Given the vectors  $x_{n+i}, 0 \leq i \leq k+1$ , and given that the QR-factorization is done with MGS, the operation count for determining  $s_{n,k}$ , whether for MPE or RRE, via our algorithms in Tables 1 and 2, is as follows:  $\frac{1}{2}(k^2 + 5k + 2)$  vector additions,  $\frac{1}{2}(k^2 + 5k)$  scalar–vector multiplications, and  $\frac{1}{2}(k^2 + 3k + 2)$  inner products. All this amounts roughly to  $2k^2N$  floating-point operations.

The quality of  $s_{n,k}$  can be ascertained by looking at the  $l_2$ -norm of the vector  $U_k \gamma$ , which is available at no additional cost through the algorithms in Tables 1 and 2:

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

Here  $r_{kk}$  is the  $(k + 1, k + 1)$  element of the matrix  $R_k$ , and  $\lambda$  is the (positive) scalar computed in Step 2 of the RRE algorithm in Table 2. The justification of this is as follows: When the sequence  $\{x_n\}$  is generated linearly as in (3.2), we have that  $U_k \gamma = r(s_{n,k})$ , where  $r(x) = Tx + b - x$  is the residual vector associated with  $x$ ; hence  $U_k \gamma$  is indeed the residual vector  $r(s_{n,k})$ . 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  $x$ .

Finally, note that the algorithms for MPE and RRE that we have summarized here allow us to compute all  $s_{n,k}$ ,  $k = 1, \dots, k_{\max}$ , without having to orthogonalize the vectors  $u_n, u_{n+1}, \dots, u_{n+k}$  for each  $k$  from the beginning.

### 7. Efficient use of MPE and RRE in solving equation systems

As mentioned already in the preceding section, to determine  $s_{n,k}$ , we need to store  $x_n$  and vectors  $q_0, q_1, \dots, q_k$ , namely 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 look at the sequence  $\{x_n\}$  considered in (5.1) of Theorem 5.2. Because

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

in this case, instead of (5.5) and (5.6), we have

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

and

$$\lim_{n \rightarrow \infty} \sum_{j=0}^k \gamma_j^{(n,k)} \sigma^j = \prod_{i=1}^k \frac{\sigma - \sigma_i}{1 - \sigma_i}; \quad \sigma_i = \lambda_i^r, \quad i = 1, 2, \dots \tag{7.2}$$

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

We now give another justification of this strategy. From the analysis given in [16], it follows that the error  $s_{n,k} - s$  is proportional to  $\prod_{i=1}^k (1 - \lambda_i)^{-1}$  as  $n \rightarrow \infty$ . This shows that the error will be large in case the largest  $\lambda_i$  (or some of them) are close to 1 in the complex plane. In such a case, by (5.6), some of the  $\gamma_j^{(n,k)}$  will be quite large for large  $n$ , despite that fact that  $\sum_{j=0}^k \gamma_j^{(n,k)} = 1$ . (Note also that  $\sum_{j=0}^k |\gamma_j^{(n,k)}| \geq 1$ .) This implies that errors committed in the computation of the vectors  $x_i$  will be magnified in the computation of  $s_{n,k}$  (via (3.12) and (3.15)), which means that the computed  $s_{n,k}$  will be quite different from the exact one. If we could do something to cause the largest  $\lambda_i$  to separate from 1 as much as possible, this may help to solve both problems. One effective way of achieving this is by applying MPE or RRE to the subsequence  $\{x_{rm}\}_{m=0}^\infty$ , which we already discussed in the preceding paragraph. Note that, with  $r > 1$ ,  $\lambda_i^r$  is farther from 1 in the complex plane than  $\lambda_i$  is.

#### 7.1. Extrapolation with cycling

We now present a practical strategy for computing  $s$ , the solution to the (in general nonlinear) system

$$x - F(x) = 0,$$

that involves *cycling* (or *restarting*) that is motivated by Theorem 5.2 and justified by Theorem 5.4 and also by (7.1) and (7.2). 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

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

C2. Apply MPE or RRE to the sequence  $\{y_i\}$  precisely as in Table 1 or Table 2, 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*. We will also denote the  $s_{n,k}$  that is computed in the  $i$ th cycle  $s_{n,k}^{(i)}$ .

A discussion 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) – that is based on Theorem 5.4 is given in [21,22]. Note that, in this mode, the integer  $n$ , just as  $k$ , is kept *fixed* (in Theorem 5.2, we let  $n$  tend to infinity), and the analysis of [21,22] takes this into account and derives upper bounds on the error of  $s_{n,k}$ . As we saw following Theorem 5.4, these upper bounds are expressible in terms of Jacobi polynomials for certain types of spectra of the matrix  $T$ , and they turn out to be quite tight. They also indicate that, with even moderate  $n$ ,  $s_{n,k}$  may be a very good approximations to  $s$  with small  $k$ , hence small storage, and few iterations. Another advantage of applying MPE and RRE in this mode (that is, with  $n > 0$ ) is that it prevents stagnation in the cases where GMRES stagnates. (See the numerical examples in [22].)

Combining cycling with Theorem 5.4, in the notation of Theorem 5.4, we can now state the following result concerning the approximation  $s_{n,k}^{(i)}$  that is computed at the  $i$ th cycle above.

**Theorem 7.1.** *Let  $s$  be the solution to the linear system  $(I - T)x = b$ , where  $I - T$  is nonsingular. Let the vector sequence  $\{x_n\}$  be generated via  $x_{n+1} = F(x_n) = Tx_n + b$ ,  $n = 0, 1, \dots$ , precisely as in Theorem 5.2. Define the residual vector associated with the vector  $x$  by  $r(x) = b - (I - T)x$ . Let  $x_{\text{init}}$  be the vector  $x_0$  in Step C0 of the cycling procedure above, and denote*

$$\omega_{n,k,r} = \min_{p \in \mathcal{P}_k} \|T^{rn} p(T^r)\|; \quad \mathcal{P}_k = \{p(z) : p \in \Pi_k, p(1) = 1\}.$$

Then, under RRE,  $s_{n,k}^{(i)}$  satisfies

$$\|r(s_{n,k}^{(i)})\| \leq \omega_{n,k,r}^i \|r(x_{\text{init}})\|.$$

If  $T$  is diagonalizable, that is,

$$T = R\Lambda R^{-1}; \quad \Lambda = \text{diag}(\mu_1, \mu_2, \dots, \mu_N),$$

then

$$\omega_{n,k,r} \leq \kappa(R) \Gamma_{n,k}^{\text{spect}(T^r)},$$

with

$$\kappa(R) = \|R\| \|R^{-1}\| \quad \text{and} \quad \Gamma_{n,k}^{\text{spect}(T^r)} = \min_{p \in \mathcal{P}_k} \max_{z \in \text{spect}(T)} |z^{rn} p(z^r)|.$$

**Note.** The results we have given above concern the application of MPE and RRE to vector sequences arising from fixed-point iteration of *linear* systems. As far as *nonlinear* systems are concerned, little is known about the convergence of these methods. There have been several attempts at studying the convergence of vector extrapolation methods in general as these are applied to nonlinear systems in the cycling mode. The earliest studies due to Brezinski [26,27], Gekeler [28], and Skelboe [29] have been reviewed in [1]. It follows heuristically from the study in [29], for example, that if MPE and RRE are applied using the cycling strategy above, with  $n = 0$  and  $r = 1$  in Step C0, and with  $k = k_i$  at the  $i$ th cycle in Step C1, where  $k_i$  is precisely the degree of the minimal polynomial of the matrix  $F'(s)$ , the Jacobian matrix of  $F(x)$  at  $x = s$ , with respect to  $s_{0,k_i}^{(i-1)} - s$ , the sequence  $\{s_{0,k_i}^{(i)}\}_{i=0}^{\infty}$  converges quadratically. (Here, we let  $s_{0,k_0}^{(0)}$  be the vector  $x_0$  in Step C0.) A precise study of the topological epsilon algorithm, one of the epsilon algorithms, in the cycling mode has been given by Le Ferrand [30].

## 7.2. Cycling with “frozen” $\gamma_i$

When storage is a problem and the unitary matrix  $Q_{k-1}$  in Step 2 of the MPE and RRE algorithms given in Tables 1 and 2 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 substantially as follows: Perform Steps C1–C3 in the cycling mode  $m$  times ( $m$  is a small integer like 1 or 2), and save the  $\gamma_i$  that are computed (in Step 2 of the MPE and RRE algorithms in Tables 1 and 2) at the  $m$ th cycle. In subsequent cycles, perform Step C2 using

these (frozen)  $\gamma_i$  and set  $s_{n,k} = \sum_{i=0}^k \gamma_i y_{n+i}$ , instead of computing  $s_{n,k}$  by MPE and RRE as described in Tables 1 and 2. This strategy enables the introduction of the vectors  $y_j$  one-by-one without having to save them. In addition, it also avoids the computational (timewise) overhead of MPE and RRE on account of the QR-factorization of the matrix  $U_k$ . The idea behind this is that the  $\gamma_i$ , from one application of Step C2 to the next, do not change very much. This is so because the  $\gamma_i^{(n,k)} \equiv \gamma_i$  associated with  $s_{n,k}$  obtained by MPE and RRE are such that  $\lim_{n \rightarrow \infty} \gamma_i^{(n,k)}$  all exist and satisfy (7.2). We have verified the validity of this shortcut with several numerical examples involving both linear and nonlinear systems with very good results.

### 8. Computation of dominant eigenvectors by MPE and RRE

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.

Because our approach relies on power iterations with the matrix  $A$ , we first summarize the properties of power iterations. This will set the stage for later developments, and fix some of the notation as well.

For simplicity of presentation, we assume again that  $A$  is diagonalizable. Choosing an arbitrary initial vector  $x_0 \in \mathbb{C}^N$ , we compute the vectors  $x_1, x_2, \dots$ , via

$$x_{n+1} = Ax_n, \quad n = 0, 1, \dots \tag{8.1}$$

Then  $x_n$  can be shown to have the spectral decomposition

$$x_n = \sum_{i=1}^q w_i \mu_i^n, \quad n = 1, 2, \dots \tag{8.2}$$

Here, (i)  $\mu_i$  are some or all of the *distinct nonzero* eigenvalues of  $A$ , (ii) for each  $i$ ,  $w_i$  is an eigenvector corresponding to  $\mu_i$ , whether  $\mu_i$  is simple or multiple; that is,  $Aw_i = \mu_i w_i, i = 1, \dots, q$ , and, of course, (iii)  $q \leq N$ . Thus, the  $w_i$  in the summation in (8.2) are linearly independent vectors, and each of the vectors  $w_i \mu_i^n$  in (8.2) is nonzero. Whether a distinct nonzero eigenvalue of  $A$  appears in the spectral decomposition of  $x_n$  depends, of course, on the spectral decomposition of the initial vector  $x_0$ . It does only if at least one of its corresponding eigenvectors is present in the spectral decomposition of  $x_0$ . We also order the  $\mu_i$  in (8.2) as follows:

$$|\mu_1| \geq |\mu_2| \geq \dots \geq |\mu_q|. \tag{8.3}$$

We assume that  $\mu_1$  is the *unique* largest eigenvalue (that is,  $|\mu_1| > |\mu_2|$ ), and that it is known. We are interested in determining the corresponding eigenvector  $w_1$ .

Let us define  $B \equiv \mu_1^{-1} A$ , and do power iterations starting with the vector  $x_0$  as in (8.1), but with the matrix  $A$  there replaced by  $B$ ; that is,

$$x_{n+1} = Bx_n, \quad n = 1, 2, \dots \tag{8.4}$$

Then (8.2) becomes

$$x_n = w_1 + \sum_{i=1}^p v_i \lambda_i^n, \quad n = 1, 2, \dots, \tag{8.5}$$

where

$$p = q - 1; \quad \lambda_i = \mu_{i+1}/\mu_1 \quad \text{and} \quad v_i = w_{i+1}, \quad i = 1, 2, \dots, p, \tag{8.6}$$

with  $\mu_i$  and  $w_i$  precisely as in (8.2). Also, with the  $\mu_i$  ordered as in (8.3), we have

$$1 > |\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p|. \tag{8.7}$$

Thus,  $\lim_{n \rightarrow \infty} x_n = w_1$ , and the rate of convergence is given by  $x_n - w_1 = O(|\lambda_1|^n)$  as  $n \rightarrow \infty$ . Of course, when  $|\lambda_1|$  is close to one, the convergence of  $\{x_n\}_{n=0}^\infty$  will be slow.

By Theorem 5.2, the expansion in (8.5), however, immediately suggests that the convergence of  $\{x_n\}_{n=0}^\infty$  can be accelerated by applying to it appropriate vector extrapolation methods, such as MPE and RRE. We thus have the following theorem:

**Theorem 8.1.** *With  $x_n$  as in (8.4)–(8.7), let  $s_{n,k}$  stand for either  $s_{n,k}^{MPE}$  or  $s_{n,k}^{RRE}$ . Then, also provided that*

$$|\lambda_k| > |\lambda_{k+1}| \quad \text{for some } k < p, \tag{8.8}$$

there holds

$$s_{n,k} - w_1 = O(|\lambda_{k+1}|^n) \quad \text{as } n \rightarrow \infty. \tag{8.9}$$

Clearly, the performance of MPE and RRE on the present problem can be improved by using the cycling strategies described in Section 7.

- Remarks.** 1. The convergence result given in (8.8) and (8.9) remains virtually the same even when the matrix  $A$  is not diagonalizable, provided the largest eigenvalue  $\mu_1$  has only associated eigenvectors but no principal vectors. The precise convergence result for this general case that has been given in [19] is however more involved.
2. Note that, in the problems we are treating here,  $\lim_{n \rightarrow \infty} x_n$  exists and is equal to  $w_1$  since  $|\lambda_1| < 1$ , and we have stated Theorem 5.2 to suit these problems. However, as already stated in Remark 1 following Theorem 5.2, Theorem 5.2 remains valid also when  $\lim_{n \rightarrow \infty} x_n$  does not exist (which happens in case  $|\lambda_1| \geq 1$ ). In this case,  $\lim_{n \rightarrow \infty} s_{n,k}$  exists and equals  $w_1$ , provided  $|\lambda_{k+1}| < 1$ . This observation becomes useful when the matrix  $A$  has several known distinct largest eigenvalues having the same modulus, that is, when (8.3) assumes the form

$$|\mu_1| = \dots = |\mu_m| > |\mu_{m+1}| \geq \dots \geq |\mu_q|,$$

$\mu_1, \dots, \mu_m$  being known. To approximate  $w_i$  for each  $i \in \{1, \dots, m\}$ , we apply MPE or RRE to the sequence  $\{x_n\}$  obtained via the power iterations  $x_{n+1} = B_i x_n, n = 0, 1, \dots$ , where  $B_i = \mu_i^{-1} A$ . In this case,  $x_n$  satisfies

$$x_n = w_i + \sum_{j=1}^p v_j \lambda_j^n, \quad n = 1, 2, \dots,$$

where

$$(\lambda_j, v_j) = \begin{cases} (\mu_j/\mu_i, w_j) & j = 1, \dots, i-1, \\ (\mu_{j+1}/\mu_i, w_{j+1}) & j = i, \dots, p; \quad p = q-1. \end{cases}$$

Note that now

$$1 = |\lambda_1| = \dots = |\lambda_{m-1}| > |\lambda_m| \geq \dots \geq |\lambda_p|.$$

As a result, the sequence  $\{x_n\}$  is bounded, but  $\lim_{n \rightarrow \infty} x_n$  does not exist. Nevertheless, we can approximate  $w_i$  by applying MPE or RRE with  $k \geq m-1$  since  $|\lambda_{m-1}| > |\lambda_m|$ . For example,  $\lim_{n \rightarrow \infty} s_{n,m-1} = w_i$ ; we actually have

$$s_{n,m-1} - w_i = O(|\lambda_m|^n) \quad \text{as } n \rightarrow \infty.$$

### 9. Application of Krylov subspace methods to consistent singular linear systems

Suppose the vector sequence  $\{y_m\}_{m=0}^\infty$  is generated via the iterative process  $y_{m+1} = My_m + b, m = 0, 1, \dots$ , where  $M$  is a square matrix and  $(I - M)$  is nonsingular. As mentioned in Theorem 5.1, the vectors  $s_{0,k}$  produced by the vector extrapolation methods MPE and RRE, as they are being applied to the vector sequence  $\{y_m\}$ , are those that are generated by, respectively, the method of Arnoldi [8] and GMRES of Saad and Schultz [9] for linear systems, as these are being applied to the solution of the linear system  $(I - M)y = b$ , with initial vector  $y_0$ .

In another paper by Sidi [31], it is shown that this applies also when the matrix  $(I - M)$  is singular, which happens when 1 is an eigenvalue of  $M$ , provided the system  $(I - M)y = b$  is consistent. It also follows from [31] that, when the eigenvalue 1 of  $M$  is not defective, that is, it has only corresponding eigenvectors but no principal vectors, then the vectors  $s_{0,k}$  obtained from these methods approximate  $y'_0 + (I - M)^\# b$ , where  $y'_0$  is that part of  $y_0$  that lies in the null space of  $(I - M)$  and  $(I - M)^\#$  is the group inverse of  $(I - M)$ . Thus, if  $b = 0$ , then we obtain approximations



to  $y'_0$  only. For group inverses, and other generalized inverses too, see Ben-Israel and Greville [32] and Campbell and Meyer [33].

Let us now go back to the matrix  $B$  considered in Section 8 in (8.4)–(8.7) and Theorem 8.1. Then, the matrix  $B^r$ , just as  $B$ , has 1 as its largest eigenvalue. In addition, this eigenvalue is nondefective. Hence  $(I - B^r)x = 0$ , being a homogeneous singular system, is also consistent. Therefore, the vectors  $s_{0,k}$  obtained by applying MPE or RRE to the sequence  $\{x_{rm}\}_{m=0}^{\infty}$ , are the same as those that are obtained by applying, respectively, the method of Arnoldi or GMRES to the system  $(I - B^r)x = 0$ , with initial vector  $x_0$ . That is,  $M = B^r$ ,  $b = 0$ ,  $y_0 = x_0$ , and  $y'_0 = w_1$  in the preceding paragraph.

Also note that the system  $(I - B^r)x = 0$  can be obtained by preconditioning the matrix  $(I - B)$  with the (nonsingular) matrix  $K = I + B + B^2 + \dots + B^{r-1}$ , since  $KB = I - B^r$ . Thus, what we have here is (polynomially) preconditioned Krylov subspace methods applied to the solution of  $(I - B)x = 0$ .

We note that Golub and Greif [34,35] apply the Arnoldi method and Arnoldi-type methods that involve the singular value decomposition to the consistent singular system  $(B - I)x = 0$  to compute the PageRank. It seems likely that the polynomial preconditioner we have proposed here may help accelerate the convergence further.

## 10. Application of MPE and RRE 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 real  $N \times N$  matrix  $A$  is *column-stochastic* if it is nonnegative and the sum of the elements in each of its columns is 1. Thus,  $A$  is column-stochastic if

$$a_{ij} \geq 0 \quad \text{for all } i \text{ and } j;$$

$$\sum_{i=1}^N a_{ij} = 1, \quad j = 1, \dots, N.$$

Therefore, if  $A$  is column-stochastic, then  $\rho(A) = 1$ , and  $A$  has  $\mu_1 = \rho(A) = 1$  as its eigenvalue, the corresponding left eigenvector being  $e = [1, 1, \dots, 1]^T$ . (As usual,  $\rho(A)$  stands for the spectral radius of  $A$ .) It is known that  $A$  has a nonnegative right eigenvector  $w_1$  that corresponds to the eigenvalue  $\mu_1$ . If  $A$  is also irreducible, then this eigenvector is positive. These results are well known in the theory of nonnegative matrices. See, for example, the book by Berman and Plemmons [36, Chapter 2].

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 also column-stochastic. In addition,  $E$  is of the form  $E = ue^T$ , where  $e = [1, 1, \dots, 1]^T$  as before, and  $u$  is a nonnegative vector such that  $e^T u = 1$ . Interestingly, whether  $u$  is positive or nonnegative,  $\mu_1 = 1$  is always simple, and this result follows from the paper [37], where it is proved that the second eigenvalue  $\mu_2$  of  $A$  satisfies  $|\mu_2| \leq c < 1$ . Thus, even if  $P$  has a multiple eigenvalue equal to 1,  $A$  has a simple eigenvalue equal to 1. Yet, in two other papers [38,39] by Langville and Meyer, the following is proved concerning the eigenvalues of  $A$ : If  $1, \mu'_2, \mu'_3, \dots, \mu'_N$  are the eigenvalues of  $P$ , then the eigenvalues of  $A$  are  $1, c\mu'_2, c\mu'_3, \dots, c\mu'_N$ . For another short proof of this result, see Eldén [40].

In summary, the largest eigenvalue of the Google matrix  $A$  is  $\mu_1 = 1$  and this eigenvalue is simple and the corresponding (right) eigenvector  $w_1$  is positive. (Thus, our matrix  $A$  is simply the matrix  $B$  of Section 8.) The PageRank is simply  $w_1$ , normalized such that the sum of its components is 1.

Now, the matrix  $P$  is very sparse, the number of nonzero elements in each of its rows being  $O(1)$ . This means that the cost of computing a matrix-vector product  $Pw$  is  $O(N)$  arithmetic operations. The cost of computing the product  $Ew$  is also the same because  $Ew = (u^T w)e$ . As a result, the cost of computing the matrix-vector product  $Aw$  is  $O(N)$  arithmetic operations. From this, we see that the matrix  $A$ , despite the fact that it is *not* a sparse matrix, is *behaving* like one in the computation of  $Aw$ . Thus, methods that are based on power iterations can be most useful in PageRank computations.

Because storage is a crucial problem in PageRank computations, we must strive to use methods that give accurate results quickly and require little storage. These aims can be realized by applying MPE and RRE in the cycling mode

to the sequence of power iterations  $\{x_m\}$ ,  $x_{m+1} = Ax_m$ ,  $m = 0, 1, \dots$ , precisely as in Section 7.1, by tuning the integer parameters  $n$ ,  $k$ , and  $r$  appropriately. One other strategy can be to apply MPE and RRE in the cycling mode with frozen  $\gamma_i$ , as explained in Section 7.2. This way we need to save only one vector and do not have the overhead involved in implementing MPE and RRE through the full algorithms in Tables 1 and 2.

Note that, when the initial vector  $x_0$  is chosen to be positive and satisfies  $e^T x_0 = 1$ , all the power iterations  $x_m$ ,  $m = 1, 2, \dots$ , are also positive and satisfy  $e^T x_m = 1$ , because  $A$  is column-stochastic. This implies that the vector  $\lim_{m \rightarrow \infty} x_m = w_1$  is exactly the PageRank. Since  $\sum_{i=0}^k \gamma_i = 1$  in both MPE and RRE,  $s_{n,k}$  is also positive and satisfies  $e^T s_{n,k} = 1$  for both MPE and RRE. In other words, MPE and RRE converge to  $w_1$ , the PageRank vector.

### Numerical example

We have applied MPE and RRE to Web matrices of different sizes, small and large, and observed the same pattern of behavior in all cases. To end this section, we report the application of MPE and RRE to the computation of the PageRank associated with a matrix  $P'$  that results from a link graph containing  $N = 281\,903$  nodes, with roughly 2.3 million links. The link graph was generated from a crawl of the `stanford.edu` domain created in September 2002 by the Stanford WebBase project. (See [41].)

The matrix  $P'$  has all its nonzero columns summing to unity. It is first modified by replacing its zero columns by the vector  $\frac{1}{N}e$ , the resulting matrix  $P$  being column-stochastic. The matrix  $A = cP + (1 - c)E$  is formed with  $E = \frac{1}{N}ee^T$ .

In our numerical computations, we took several values of  $c \in (0, 1)$ . Here we report those results obtained with  $c = 0.95$  and  $c = 0.99$ . We have applied MPE and RRE to the power iterations  $\{x_m\}$ , where  $x_{m+1} = Ax_m$ ,  $m = 0, 1, \dots$ , starting with the vector  $x_0 = \frac{1}{N}e$ . We have employed the cycling strategy precisely as described in Section 7.1, with  $(n, k, r) = (0, 10, 1)$ ,  $(0, 10, 3)$ ,  $(10, 10, 1)$ ,  $(10, 10, 3)$ . The results of our computations are shown in Figs. 1 and 2. Note that the norms of residuals associated with MPE and RRE shown in these figures are those of  $x_0, x_1, \dots, x_{(n+k+1)r-1}$  ( $x_{(n+k+1)r}$  not included) and of  $s_{n,k}$  (instead of  $x_{(n+k+1)r}$ ), for each cycle. The residual associated with a vector  $x$  is defined to be the vector  $r(x) = Ax - x$ .

The graphs in these figures confirm first that, in the cycling mode, MPE and RRE do accelerate the convergence of power iterations, in the sense that when they employ a given number of power iterations, they achieve higher accuracy than the iteration vectors they employ. They also confirm that with the same  $n$ ,  $k$ , and  $r$ , MPE and RRE produce very similar results, as expected by Theorem 5.2.

Next, the graphs show that MPE and RRE, with the same  $k$  and  $r$ , have achieved the same accuracy both with  $n = 0$  and with  $n = 10$ , using about the same number of power iterations. However, the number of cycles with  $n = 10$  is much smaller than (about half) that with  $n = 0$ , hence take a smaller amount of computing time when the time overhead of cycling is taken into account. Similarly, with the same  $k$  and  $n$ , MPE and RRE have achieved the same accuracy both with  $r = 1$  and with  $r = 3$ , again using about the same number of power iterations. Now the results with  $r = 3$  are obtained in a smaller number of cycles (about one-third of that necessary with  $r = 1$ ), hence take a smaller amount of computing time. Both of these observations confirm our statements of Section 7 concerning the strategy of cycling.

## 11. Generalization of the method of Kamvar et al. and connection with MPE

In their paper [12] on the computation of the PageRank, Kamvar et al. have developed an extrapolation method, which they call *quadratic extrapolation*. This method accelerates the convergence of the sequence  $\{x_m\}$  generated as in (8.4) to a scalar multiple of the vector  $w_1$  in (8.5).

With a proper change of notation and indexing, quadratic extrapolation is the  $k = 2$  case of the general method we propose next. Here are the steps of this method:

- Input: The vectors  $x_n, x_{n+1}, \dots, x_{n+k+1}$ .
- Let  $y_i = x_{n+i+1} - x_n$ ,  $i = 0, 1, \dots, k$ , and solve the minimization problem

$$\min_{d_0, d_1, \dots, d_{k-1}} \left\| \sum_{i=0}^k d_i y_i \right\|, \quad \text{subject to } d_k = 1. \quad (11.1)$$

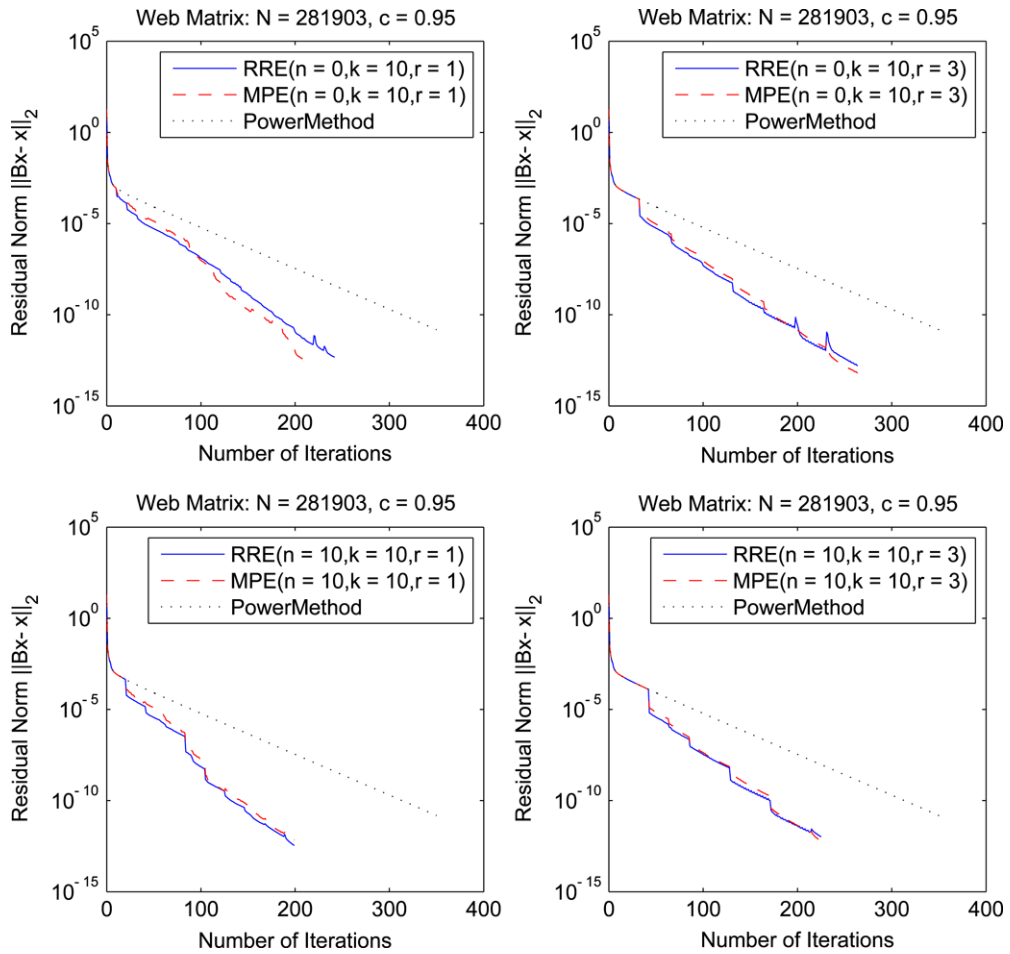


Fig. 1. Application of MPE, RRE, and the power method to the Google matrix of dimension  $N = 281,903$  of Section 10 with  $c = 0.95$ .

- Compute  $\hat{c}_0, \hat{c}_1, \dots, \hat{c}_k$  via

$$\hat{c}_i = \sum_{j=i}^k d_j, \quad i = 0, 1, \dots, k. \quad (\text{Thus, } \hat{c}_k = d_k = 1.) \quad (11.2)$$

- Set

$$\hat{s}_{n,k} = \sum_{i=0}^k \hat{c}_i x_{n+i+1}. \quad (11.3)$$

A numerically stable and storagewise economical implementation can be designed along the lines of MPE and RRE given in Tables 1 and 2. Here are the steps of this implementation:

- S1. Form the QR-factorization of the matrices  $Y_j = [y_0 \mid y_1 \mid \dots \mid y_j]$ , namely,  $Y_j = \hat{Q}_j \hat{R}_j$ , for example, by MGS. Thus,

$$\hat{Q}_j = [\hat{q}_0 \mid \hat{q}_1 \mid \dots \mid \hat{q}_j] \in \mathbb{C}^{N \times (j+1)}; \quad \hat{Q}_j^* \hat{Q}_j = I_{(j+1) \times (j+1)},$$

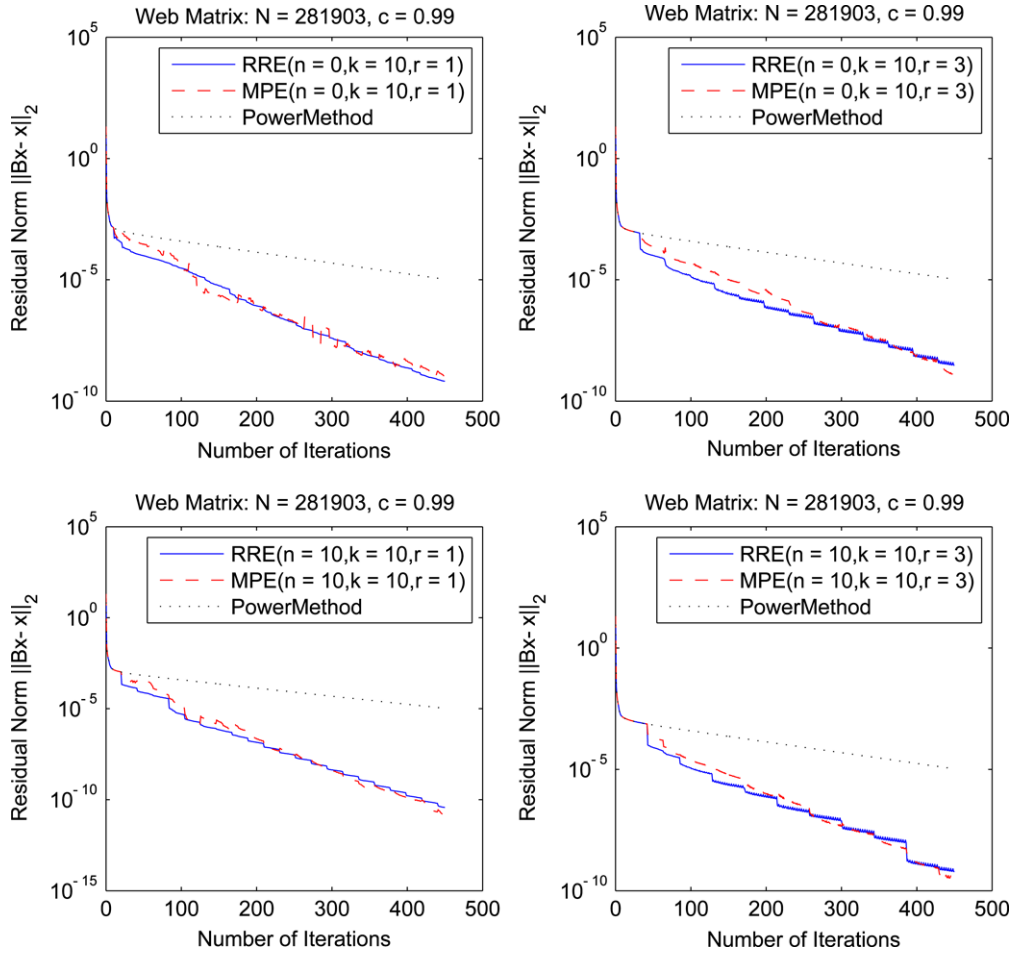


Fig. 2. Application of MPE, RRE, and the power method to the Google matrix of dimension  $N = 281,903$  of Section 10 with  $c = 0.99$ .

and

$$\widehat{R}_j = \begin{bmatrix} \widehat{r}_{00} & \widehat{r}_{01} & \cdots & \widehat{r}_{0j} \\ & \widehat{r}_{11} & \cdots & \widehat{r}_{1j} \\ & & \ddots & \vdots \\ & & & \widehat{r}_{jj} \end{bmatrix} \in \mathbb{C}^{(j+1) \times (j+1)}; \quad \widehat{r}_{ii} > 0 \text{ for all } i.$$

Then solve the linear system

$$\widehat{R}_{k-1}d = -\widehat{\rho}_k;$$

where

$$\widehat{\rho}_k = [\widehat{r}_{0k}, \widehat{r}_{1k}, \dots, \widehat{r}_{k-1,k}]^T, \quad d = [d_0, d_1, \dots, d_{k-1}]^T.$$

(Note that  $\widehat{\rho}_k = \widehat{Q}_{k-1}^* y_k$ .)

S2. Set  $d_k = 1$  and compute  $\widehat{c} = [\widehat{c}_0, \widehat{c}_1, \dots, \widehat{c}_k]^T$  by (11.2).

S3. Set

$$\widehat{s}_{n,k} = \left( \sum_{i=0}^k \widehat{c}_i \right) x_n + \widehat{Q}_k (\widehat{R}_k \widehat{c}).$$

Steps S1 and S2 in this algorithm essentially generalize those given in [12]. Step S3 (that is new) is the result of the following observation and of the fact that  $Y_k = \widehat{Q}_k \widehat{R}_k$ :

$$\begin{aligned} \widehat{s}_{n,k} &= \left( \sum_{i=0}^k \widehat{c}_i \right) x_n + \sum_{i=0}^k \widehat{c}_i y_i \\ &= \left( \sum_{i=0}^k \widehat{c}_i \right) x_n + Y_k \widehat{c}. \end{aligned}$$

Thus, on account of Step 3 in the new algorithm, we need to store only the vector  $x_n$  and the matrix  $\widehat{Q}_k$ , namely, the vectors  $\widehat{q}_0, \widehat{q}_1, \dots, \widehat{q}_k$ , a total of  $k + 2$  vectors in  $\mathbb{C}^N$ . This should be compared with the algorithm of [12], in which both  $x_{n+1}, \dots, x_{n+k+1}$  and  $\widehat{q}_0, \widehat{q}_1, \dots, \widehat{q}_k$ , a total of  $2k + 2$  vectors in  $\mathbb{C}^N$ , need to be saved.

The following theorem shows that this method and MPE are different and yet almost identical.

**Theorem 11.1.** *When the vector sequence  $\{x_m\}$  is obtained through power iterations with the matrix  $B$  via  $x_{m+1} = Bx_m, m = 0, 1, \dots$ , the  $\widehat{s}_{n,k}$  and  $s_{n,k}^{\text{MPE}}$  are related in the sense that*

$$\widehat{s}_{n,k} = \alpha \left( B s_{n,k}^{\text{MPE}} \right); \quad \alpha = \sum_{i=0}^k c_i \quad \text{as in Table 1.} \tag{11.4}$$

**Proof.** Noting that

$$y_i = \sum_{j=0}^i u_{n+j}, \quad i = 0, 1, \dots; \quad u_i = \Delta x_i, \quad i = 0, 1, \dots,$$

and recalling (11.2), we can express the vector  $\sum_{i=0}^k d_i y_i$  as in

$$\sum_{i=0}^k d_i y_i = \sum_{i=0}^k \widehat{c}_i u_{n+i}.$$

Since  $d_0, d_1, \dots, d_{k-1}$  determine  $\widehat{c}_0, \widehat{c}_1, \dots, \widehat{c}_{k-1}$  uniquely, and vice versa, this means that the minimization problem in (11.1) is equivalent to

$$\min_{\widehat{c}_0, \widehat{c}_1, \dots, \widehat{c}_{k-1}} \left\| \sum_{i=0}^k \widehat{c}_i u_{n+i} \right\|, \quad \text{subject to } \widehat{c}_k = 1. \tag{11.5}$$

Comparing (11.5) with (3.10), we realize that  $\widehat{c}_i = c_i, i = 0, 1, \dots, k$ . Substituting this in (11.3), we obtain  $\widehat{s}_{n,k} = \sum_{i=0}^k c_i x_{n+i+1}$ . Recalling the fact that  $x_{m+1} = Bx_m$ , we finally have

$$\widehat{s}_{n,k} = B \left( \sum_{i=0}^k c_i x_{n+i} \right).$$

The result now follows by the fact that  $s_{n,k}^{\text{MPE}} = \alpha^{-1} \sum_{i=0}^k c_i x_{n+i}$  with  $\alpha$  as in (3.11). ■

**Remark.** With the  $\widehat{c}_j$  computed as above, if we modify the definition of  $\widehat{s}_{n,k}$  in (11.3) slightly so that  $\widehat{s}_{n,k}$  is now given by

$$\widehat{s}_{n,k} = \sum_{i=0}^k \widehat{c}_i x_{n+i},$$

then we would have precisely  $\widehat{s}_{n,k} = \alpha s_{n,k}^{\text{MPE}}$ .

In view of the result of Theorem 11.1, we can use Theorem 5.2, which concerns the approximations  $s_{n,k}$  produced by MPE and RRE, to state the following convergence theorem for  $\widehat{s}_{n,k}$ .

**Theorem 11.2.** *Let the vectors  $x_m$  satisfy the conditions of Theorem 8.1. Then*

$$\|\alpha^{-1}\widehat{s}_{n,k} - w_1\| \leq \|B\| \|s_{n,k}^{\text{MPE}} - w_1\|, \tag{11.6}$$

and

$$\alpha^{-1}\widehat{s}_{n,k} = w_1 + O(|\lambda_{k+1}|^n) \quad \text{as } n \rightarrow \infty. \tag{11.7}$$

**Proof.** By Theorem 11.1 and by the fact that  $Bw_1 = w_1$ , we have

$$\widehat{s}_{n,k} = \alpha B \left( s_{n,k}^{\text{MPE}} - w_1 \right) + \alpha B w_1 = \alpha B \left( s_{n,k}^{\text{MPE}} - w_1 \right) + \alpha w_1,$$

from which

$$\alpha^{-1}\widehat{s}_{n,k} - w_1 = B \left( s_{n,k}^{\text{MPE}} - w_1 \right).$$

Now apply Theorem 5.2 to  $s_{n,k}^{\text{MPE}}$ . ■

Note that, if we use the vector  $l_1$ -norm in Theorem 11.2, then we have  $\|B\| = 1$  because  $B$  is column-stochastic. Hence (11.6) becomes

$$\|\alpha^{-1}\widehat{s}_{n,k} - w_1\| \leq \|s_{n,k}^{\text{MPE}} - w_1\|.$$

In a subsequent work [41], Kamvar, Haveliwala, and Golub have observed that, when using quadratic extrapolation, many web pages converge quickly, while relatively few pages take much longer time to converge. Furthermore, the pages that converge slowly are generally those pages with high PageRank. Taking this into account, Kamvar, Haveliwala, and Golub have proposed an adaptive method, in which the effective Google matrix is much smaller after many of the pages have converged; as a result, in subsequent applications of quadratic extrapolation, the computing time is reduced considerably. Needless to say, this adaptive method can be used just as effectively with MPE and RRE and all other extrapolation methods.

**Remark.** Recently, the PageRank computation has been also treated in a paper by Brezinski and Redivo Zaglia [42, Section 6]. Just as our present work and its predecessor [13], [42] too is concerned with computing the PageRank via vector extrapolation methods. In particular, in subsection 6.1 of their paper, Brezinski and Redivo Zaglia make an attempt at generalizing the method of Kamvar et al., their end result being the vector  $\mathbf{r}_c^{(k,n)}$  in Theorem 6.1. In Theorem 6.2 of [42], a determinant representation of  $\mathbf{r}_c^{(k,n)}$  that is expressed in terms of the power iterations  $\mathbf{r}_c^{(n)} = A_c \mathbf{r}_c^{(n-1)}$  is given. (Note that our matrix  $B$  and vectors  $x_n$  are, respectively, the matrix  $A_c$  and the vectors  $\mathbf{r}_c^{(n)}$  in [42].) A close examination of this determinant representation and its comparison with our determinant representation of  $s_{n,k}^{\text{MPE}}$  given in (4.1)–(4.3), namely,

$$s_{n,k}^{\text{MPE}} = \frac{\begin{vmatrix} x_n & x_{n+1} & \cdots & x_{n+k} \\ (\Delta x_n, \Delta x_n) & (\Delta x_n, \Delta x_{n+1}) & \cdots & (\Delta x_n, \Delta x_{n+k}) \\ (\Delta x_{n+1}, \Delta x_n) & (\Delta x_{n+1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+1}, \Delta x_{n+k}) \\ \vdots & \vdots & \ddots & \vdots \\ (\Delta x_{n+k-1}, \Delta x_n) & (\Delta x_{n+k-1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+k-1}, \Delta x_{n+k}) \\ 1 & 1 & \cdots & 1 \\ (\Delta x_n, \Delta x_n) & (\Delta x_n, \Delta x_{n+1}) & \cdots & (\Delta x_n, \Delta x_{n+k}) \\ (\Delta x_{n+1}, \Delta x_n) & (\Delta x_{n+1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+1}, \Delta x_{n+k}) \\ \vdots & \vdots & \ddots & \vdots \\ (\Delta x_{n+k-1}, \Delta x_n) & (\Delta x_{n+k-1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+k-1}, \Delta x_{n+k}) \end{vmatrix}}{\begin{vmatrix} (\Delta x_n, \Delta x_n) & (\Delta x_n, \Delta x_{n+1}) & \cdots & (\Delta x_n, \Delta x_{n+k}) \\ (\Delta x_{n+1}, \Delta x_n) & (\Delta x_{n+1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+1}, \Delta x_{n+k}) \\ \vdots & \vdots & \ddots & \vdots \\ (\Delta x_{n+k-1}, \Delta x_n) & (\Delta x_{n+k-1}, \Delta x_{n+1}) & \cdots & (\Delta x_{n+k-1}, \Delta x_{n+k}) \end{vmatrix}},$$

reveal that the numerator determinant of  $\mathbf{r}_c^{(k,n)}$  is, in our notation, identical to the numerator determinant of  $s_{n,k-1}^{\text{MPE}}$ . That is,  $\mathbf{r}_c^{(k,n)}$  is nothing but a constant multiple of  $s_{n,k-1}^{\text{MPE}}$ . The approximations to the PageRank obtained by proper (and essential) normalization of the vectors  $s_{n,k-1}^{\text{MPE}}$  and  $\mathbf{r}_c^{(k,n)}$  are thus identical. Hence, unlike our method in this section, the method of [42] does not generalize quadratic extrapolation.

## 12. Concluding remarks

In this work, we gave a survey of MPE and RRE, two most effective and widely used vector extrapolation methods. We discussed their derivation, convergence properties, stable and economical algorithms for their implementation, and effective strategies for their use in different situations. We also showed that they can be used in computing eigenvectors corresponding to dominant eigenvalues when these eigenvalues are known. As an application of this, we discussed the use of MPE and RRE in computing the Google PageRank. We also gave a true generalization of the quadratic extrapolation method of Kamvar et al. for PageRank computation, and developed an algorithm for this generalization that is stable and has minimal storage requirements. We showed that this method is very closely related to MPE, and proved its convergence, and provided its precise rate of convergence at the same time. Finally, we showed that standard Krylov subspace methods such as the method of Arnoldi and GMRES can also be used to compute the PageRank, and we provided a polynomial preconditioner for them as well.

## Acknowledgements

I would like to express my gratitude to the late Professor Gene H. Golub for drawing my attention to References [34] and [35]. I would also like to thank Mr. Eran Treister for carrying out the PageRank computations reported in Section 10 of this work.

## References

- [1] D.A. Smith, W.F. Ford, A. Sidi, Extrapolation methods for vector sequences, *SIAM Rev.* 29 (1987) 199–233.
- [2] 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.
- [3] 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.
- [4] M. Mešina, Convergence acceleration for the iterative solution of the equations  $X = AX + f$ , *Comput. Methods Appl. Mech. Engrg.* 10 (1977) 165–173.
- [5] P. Wynn, On a device for computing the  $e_m(S_n)$  transformation, *Math. Tab. Other Aids Comput.* 10 (1956) 91–96.
- [6] P. Wynn, Acceleration techniques for iterated vector and matrix problems, *Math. Comp.* 16 (1962) 301–322.
- [7] C. Brezinski, Généralisations de la transformation de Shanks, de la table de Padé, et de l' $\epsilon$ -algorithme, *Calcolo* 11 (1975) 317–360.
- [8] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.* 9 (1951) 17–29.
- [9] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual method for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (1986) 856–869.
- [10] A. Sidi, Rational approximations from power series of vector-valued meromorphic functions, *J. Approx. Theory* 77 (1994) 89–111.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] D. Shanks, Nonlinear transformations of divergent and slowly convergent sequences, *J. Math. Phys.* 34 (1955) 1–42.
- [15] 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).
- [16] 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).
- [17] W.F. Ford, A. Sidi, Recursive algorithms for vector extrapolation methods, *Appl. Numer. Math.* 4 (1988) 477–489.
- [18] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, *J. Comput. Appl. Math.* 22 (1988) 71–88.
- [19] A. Sidi, J. Bridger, Convergence and stability analyses for some vector extrapolation methods in the presence of defective iteration matrices, *J. Comput. Appl. Math.* 22 (1988) 35–61.
- [20] A. Sidi, Convergence of intermediate rows of minimal polynomial and reduced rank extrapolation tables, *Numer. Algorithms* 6 (1994) 229–244.
- [21] 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. The report can be downloaded from the URL: [http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19920024498\\_1992024498.pdf](http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19920024498_1992024498.pdf).
- [22] A. Sidi, Y. Shapira, Upper bounds for convergence rates of acceleration methods with initial iterations, *Numer. Algorithms* 18 (1998) 113–132.
- [23] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, in: *Nat. Bur. Standards Appl. Math. Series*, vol. 55, US Government Printing Office, Washington, DC, 1964.

- [24] G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, London, 1996.
- [25] A. Sidi, Efficient implementation of minimal polynomial and reduced rank extrapolation methods, *J. Comput. Appl. Math.* 36 (1991) 305–337 (Originally appeared as NASA TM-103240 ICOMP-90-20).
- [26] C. Brezinski, Application de l' $\epsilon$ -algorithme à la résolution des systèmes non linéaires, *C. R. Acad. Sci. Paris* 271 A (1970) 1174–1177.
- [27] C. Brezinski, Sur un algorithme de résolution des systèmes non linéaires, *C. R. Acad. Sci. Paris* 272 A (1972) 145–148.
- [28] E. Gekeler, On the solution of systems of equations by the epsilon algorithm of Wynn, *Math. Comp.* 26 (1972) 427–436.
- [29] S. Skelboe, Computation of the periodic steady-state response of nonlinear networks by extrapolation methods, *IEEE Trans. Circuits Syst.* 27 (1980) 161–175.
- [30] H. Le Ferrand, The quadratic convergence of the topological epsilon algorithm for systems of nonlinear equations, *Numer. Algorithms* 3 (1992) 273–284.
- [31] A. Sidi, Application of vector extrapolation methods to consistent singular linear systems, *Appl. Numer. Math.* 6 (1990) 487–500.
- [32] A. Ben-Israel, T.N.E. Greville, *Generalized inverses: Theory and applications*, in: CMS Books in Mathematics, 2nd ed., Springer-Verlag, New York, 2003.
- [33] S.L. Campbell, C.D. Meyer Jr., *Generalized Inverses of Linear Transformations*, Pitman, London, 1979.
- [34] G.H. Golub, C. Greif, Arnoldi-type algorithms for computing stationary distribution vectors, with applications to PageRank, Technical Report SCCM-04-15, Scientific Computing/Computational Mathematics Program, Stanford University, 2004.
- [35] G.H. Golub, C. Greif, An Arnoldi-type method for computing PageRank, *BIT* 46 (2006) 759–771.
- [36] A. Berman, R.J. Plemmons, *Nonnegative Matrices in Mathematical Sciences*, in: Classics in Applied Mathematics, vol. 9, SIAM, Philadelphia, 1994.
- [37] T.H. Haveliwala, S.D. Kamvar, The second eigenvalue of the Google matrix, Technical Report, Stanford University, 2003.
- [38] A.N. Langville, C.D. Meyer, Fiddling with PageRank, Technical Report, Department of Mathematics, North Carolina State University, 2003.
- [39] A.N. Langville, C.D. Meyer, Deeper inside PageRank, Technical Report, Department of Mathematics, North Carolina State University, 2004.
- [40] L. Eldén, The eigenvalues of the Google matrix, Technical Report LiTH-MAT-R-04-01, Linköping University, 2004.
- [41] S. Kamvar, T. Haveliwala, G. Golub, Adaptive methods for the computation of PageRank, *Linear Algebra Appl.* 386 (2004) 51–65.
- [42] C. Brezinski, M. Redivo Zaglia, The PageRank vector: Properties, computation, approximation, and acceleration, *SIAM J. Matrix Anal. Appl.* 28 (2006) 551–575.