sequences arise, for example, in the numerical solution of very large systems of linear or nonlinear equations by fixed-point iterative methods, and \( \lim_{m \to \infty} x_m \) are simply the required solutions to these systems. One common source of such systems is the finite-difference or finite-element discretization of continuum problems.

In most cases, however, the sequences \( \{x_m\} \) converge to their limits extremely slowly. That is, \( s = \lim_{m \to \infty} x_m \) can be approximated with a prescribed level of accuracy by \( x_m \) with very large \( m \). Clearly, this way of approximating \( s \) via the \( x_m \) becomes very expensive computationally. One practical way of tackling this problem effectively is by applying to the sequence \( \{x_m\} \) a suitable convergence acceleration method (or extrapolation method).

More specifically, let us consider the (linear or nonlinear) system of equations

\[
\Psi(x) = 0; \quad \Psi : \mathbb{C}^N \to \mathbb{C}^N,
\]

whose solution we denote \( s \). What is meant by \( x, s, \) and \( \Psi(x) \) are, respectively,

\[
x = [x^{(1)}, \ldots, x^{(N)}]^T, \quad s = [s^{(1)}, \ldots, s^{(N)}]^T; \quad x^{(i)}, s^{(i)} \text{ scalars},
\]

and

\[
\Psi(x) = [\psi_1(x), \ldots, \psi_N(x)]^T;
\]

\[
\psi_i(x) = \psi_i(x^{(1)}, \ldots, x^{(N)}) \text{ scalar functions}. \quad \text{\footnote{Throughout, we use lowercase boldface letters to denote vectors and vector-valued functions, and we use uppercase boldface letters to denote matrices.}}
\]

Then, starting with a suitable vector \( x_0 \), an initial approximation to \( s \), the sequence \( \{x_m\} \) of approximations can be generated by some fixed-point iterative method as in

\[
x_{m+1} = f(x_m), \quad m = 0, 1, \ldots; \quad f : \mathbb{C}^N \to \mathbb{C}^N,
\]

with

\[
f(x) = [f_1(x), \ldots, f_N(x)]^T; \quad f_i(x) = f_i(x^{(1)}, \ldots, x^{(N)}) \text{ scalar functions}.
\]

Here \( x - f(x) = 0 \) is a possibly “preconditioned” form of Equation (1), hence it has the same solution \( s \) [that is, \( \Psi(s) = 0 \) and also \( s = f(s) \)], and, in case of convergence, \( \lim_{m \to \infty} x_m = s \).

One possible form of \( f(x) \) would be

\[
f(x) = x + A(x)\Psi(x),
\]

where \( A(x) \) is an \( N \times N \) matrix, such that \( A(s) \) is nonsingular.

Now, the rate of convergence of the sequence \( \{x_m\} \) in Equation (2) to \( s \) is determined by \( p(F(s)) \), where \( F(x) \) is the Jacobian matrix\footnote{\( F(x) \) is an \( N \times N \) matrix, whose \((i,j)\) entry is \( \partial f_j / \partial x_i \) evaluated at \( x \).} of the vector-valued function \( f(x) \) and \( p(K) \) denotes the spectral radius\footnote{\( p(K) = \max |\mu_i| \), where \( \mu_i \) are eigenvalues of \( K \). Of course, \( K \) is a square matrix.} of the matrix \( K \). It is
known that \( \rho(F(s)) < 1 \) must hold for convergence to take place, and that, the closer \( \rho(F(s)) \) is to zero, the faster the convergence. The rate of convergence deteriorates as \( \rho(F(s)) \) becomes closer to 1, however.

As an example, let us consider the cases in which Equation (1) and Equation (2) arise from finite-difference or finite-element discretizations of continuum problems. For \( s \) [the solution to Equation (1) and Equation (2)] to be a reasonable approximation to the solution of the continuum problem, the mesh size of the discretization should be small enough. However, a small mesh size means a large \( N \). In addition, as the mesh size tends to zero, hence \( N \to \infty \), generally, \( \rho(F(s)) \) tends to 1, as can be shown rigorously in some cases. All this means that, when the mesh size decreases, not only does the dimension of the problem increase, the convergence of the fixed-point method in Equation (2) deteriorates as well. As mentioned above, this problem of slow convergence can be treated efficiently via vector extrapolation methods.

Before going on, we mention that, given a vector sequence \( \{x_m\} \) that converges, a vector extrapolation method computes, directly or indirectly, a "weighted average" of a certain number of the vectors \( x_m \) as an approximation to \( \lim_{m \to \infty} x_m \). This approximation is of the form

\[
\sum_{i=0}^{k} \gamma_i^{(n,k)} x_{m+i},
\]

where \( n \) and \( k \) are integers chosen by the user and the scalars \( \gamma_i^{(n,k)} \), which depend on the \( x_m \) and can even be complex, satisfy

\[
\sum_{i=0}^{k} \gamma_i^{(n,k)} = 1.4
\]

In general, the methods differ in the way they determine the \( \gamma_i^{(n,k)} \). A nice feature of vector extrapolation methods that is also of practical importance is that they take only the vectors \( x_m \) as their input. The way these vectors are generated is irrelevant; they need no other input.

An additional useful property of vector extrapolation methods is that they can even take a divergent sequence of vectors \( \{x_m\} \) into convergent ones, with the right "limits." For example, if \( s \) is the solution to \( x - f(x) = 0 \) and if \( \{x_m\} \) that is obtained from Equation (2) diverges, vector extrapolation methods can produce sequences of approximations that converge to \( s \). This can be proved rigorously when \( f(x) \) is linear. It seems to be true also when \( f(x) \) is nonlinear. In case of divergence, the solution \( s \) is called the antilimit of \( \{x_m\} \).

A detailed review of vector extrapolation methods, which contains the developments up to the early 1980s, can be found in the work of Smith et al. (1). This work discusses (1) 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 (2) three epsilon algorithms, namely, the scalar and vector epsilon algorithms of Wynn (5, 6), and the topological epsilon algorithm of Brezinski (7). When applied to very large systems of equations, polynomial-type methods have been observed generally to be more economical than the epsilon algorithms as far as computation time and storage requirements are concerned. For this reason, we concentrate on MPE and RRE in the present work.

In the next section, we present some preliminaries that justify the development of MPE and RRE via a study of linear systems. Following these preliminaries, in the succeeding six sections, we present an up-to-date review of MPE and RRE, which are 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 Ref. 1. In the third section, we give a detailed discussion of the derivation of MPE and RRE that is easy to understand even by those with limited background. In the fourth section, we provide some elegant determinant representations for them. In the fifth section, we 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. In the sixth section, we describe the most accurate and stable algorithms for their implementation in finite-precision (that is, floating-point) arithmetic; these algorithms are also the most economical as far as their computational cost and storage requirements are concerned. We provide all the details of these algorithms to enable the reader to program them easily. In the seventh section, we provide the analytic theory of MPE and RRE concerning their convergence, acceleration of convergence, error, and numerical stability behavior. The results of this section are important; they are used in the eighth section, where we discuss a mode of usage for MPE and RRE that is known as cycling and that is very suitable for solving nonlinear systems.

In the ninth section, we turn to some additional applications of vector extrapolation methods. When applied to the sequence of partial sums of a vector-valued power series from a vector-valued function, MPE serves as a tool for analytic continuation: it produces vector-valued rational approximations to the function in question that approximate the latter 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 discuss very briefly here, is treated at length in Sidi (10, 11).

Another application of vector extrapolation methods, especially of MPE and RRE, is to the computation of an eigenvector of an arbitrary large and sparse matrix that corresponds 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. Vector extrapolation methods, especially MPE and RRE, can be used effectively in the solution of this problem and hence in PageRank computations. The contents of this part of the paper on computing eigen-
vectors in general and the PageRank in particular are new and originally appeared in the author’s 2004 technical report (12).

For completeness, in the section entitled “Brief Summary of MMPE,” we discuss briefly a third polynomial method, which is known as the modified minimal polynomial extrapolation (MMPE), which was proposed independently by Brezinski (7), Pugachev (13), and Sidi et al. (14).

Again, for completeness, in the last section, we give a brief discussion of the three epsilon algorithms mentioned above.

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 \( C^N \) 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. We refer the reader to the relevant literature for these applications.

Note that, throughout this work, \((x, y)\) will stand for the Euclidean inner product of the vectors \( x, y \in C^m \), that is, \( (x, y) = x^*y \). Similarly, \( \|z\| \) will stand for the standard vector \( l_2\)-norm of \( z \in C^m \), namely, \( \|z\| = \sqrt{z^*z} \). Finally, \( \|A\| \) will stand for the matrix norm of \( A \in C^{m \times m} \) induced by this vector norm. Here, \( s \) is any positive integer.

### PRELIMINARIES AND MOTIVATION

Because one of the main areas of application of vector extrapolation methods is that of numerical solution of large and sparse (linear and nonlinear) systems of equations by fixed-point methods, we will motivate their derivation in this context. We start this by discussing the nature of the vectors \( x_m \) that arise from the iterative method of Equation (2), the function \( f(x) \) there being nonlinear in general. Assuming that \( \lim_{m \to \infty} x_m \) exists, hence that \( x_m \approx s \) for all large \( m \) [recall that \( s \) is the solution to the system \( f(s) = 0 \) and hence to the system \( x = f(x) \)], we expand \( f(x_m) \) in Equation (2) about \( s \), thus obtaining

\[
x_{m+1} = f(s) + F(s)(x_m - s) + O(\|x_m - s\|^2)
\]

as \( m \to \infty \).

Recalling that \( s = f(s) \), and rewriting Equation (3) in the form

\[
x_{m+1} - s = F(s)(x_m - s) + O(\|x_m - s\|^2) \quad \text{as} \quad m \to \infty
\]

we realize that, for all large \( m \), the vectors \( x_m \) and \( x_{m+1} \) satisfy the approximate equation \( x_{m+1} \approx F(s)x_m + |I - F(s)|s \). That is, the sequence \( \{x_m\} \) behaves as if it were being generated by an \( N \)-dimensional linear system of the form \( (I - T)x = b \) through

\[
x_{m+1} = Tx_m + b, \quad m = 0, 1, \ldots
\]

where \( T = F(s) \) and \( b = (I - F(s))s \). This suggests that we should study those sequences \( \{x_m\} \) that arise from linear systems of equations to derive and study vector extrapolation methods. We undertake this task in the next section.

### DERIVATION OF MPE AND RRE

#### Linear Systems and Iterative Methods

Let \( s \) be the unique solution to the \( N \)-dimensional linear system

\[
x = Tx + b.
\]

Writing this system in the form \((I - T)x = b\), it becomes clear that the uniqueness of the solution is guaranteed when the matrix \( I - T \) is nonsingular or, equivalently, when \( T \) does not have 1 as its eigenvalue. Let the vector sequence \( \{x_m\} \) be generated via the iterative scheme

\[
x_{m+1} = Tx_m + b, \quad m = 0, 1, \ldots
\]

Provided \( p(T) < 1 \), \( \lim_{m \to \infty} x_m \) exists and equals \( s \).

Given the sequence \( x_0, x_1, \ldots \) generated as in Equation (7), let

\[
u_m = \Delta x_m, \quad w_m = \Delta u_m = \Delta^2 x_m, \quad m = 0, 1, \ldots
\]

where \( \Delta x_m = x_{m+1} - x_m \) and \( \Delta^2 x_m = \Delta(x_{m+1}) = x_{m+2} - 2x_{m+1} + x_m \), and define the error vectors \( \epsilon_m \) as in

\[
\epsilon_m = x_m - s, \quad m = 0, 1, \ldots
\]

Using Equation (7), it is easy to show that

\[
u_m = Tu_{m-1}, \quad w_m = Tw_{m-1}, \quad m = 0, 1, \ldots
\]

From this, we have by induction that

\[
u_m = T^m u_0, \quad w_m = T^m w_0, \quad m = 0, 1, \ldots
\]
Similarly, by Equation (7), and by the fact that \( s = Ts + b \), one can relate the error in \( x_m \) to the error in \( x_{m-1} \) via
\[
\epsilon_m = T\epsilon_{m-1},
\]
which, by induction, gives
\[
\epsilon_m = T^m\epsilon_0, \quad m = 0, 1, \ldots.
\]
Of course, \( T^0 = I \) in Equations (11) and (13) and throughout. In addition, \( \epsilon_m, u_m, \) and \( w_m \) can be related via
\[
u_m = (T-1)\epsilon_m, \quad w_m = (T-1)u_m, \quad m = 0, 1, \ldots.
\]
As mentioned in the preceding section, we propose to approximate \( s \) by a “weighted average” of the \( k + 1 \) consecutive vectors \( x_0, x_1, \ldots, x_{n+k} \), for some integers \( n \) and \( k \), as in
\[
s_{n,k} = \sum_{i=0}^{k} \gamma_i x_{n+i}; \quad \sum_{i=0}^{k} \gamma_i = 1.6
\]
(As already mentioned in the Introduction, we do not restrict the scalars \( \gamma_i \) to be real and/or positive; they can be complex in general.) Substituting Equation (9) in Equation (15), and making use of the fact that \( \sum_{i=0}^{k} \gamma_i = 1 \), we obtain
\[
s_{n,k} = \sum_{i=0}^{k} \gamma_i (s + \epsilon_{n+i}) = s + \sum_{i=0}^{k} \gamma_i \epsilon_{n+i}.
\]
From this, we observe that, for \( s_{n,k} \), to be a good approximation to \( s \), the scalars \( \gamma_i \) should be chosen such that the vector \( \sum_{i=0}^{k} \gamma_i \epsilon_{n+i} \), the “weighted average” of the individual error vectors \( \epsilon_n, \epsilon_{n+1}, \ldots, \epsilon_{n+k} \), is small in some sense. Now, by Equation (13),
\[
\sum_{i=0}^{k} \gamma_i \epsilon_{n+i} = \left( \sum_{i=0}^{k} \gamma_i T \right) \epsilon_n = G(T)\epsilon_n; \quad G(z) = \sum_{i=0}^{k} \gamma_i z^i.
\]
Therefore, we should choose the polynomial \( G(z) \) satisfying \( G(1) = 1 \) to ensure that the vector \( G(T)\epsilon_n \) is small. As we will see shortly, we can actually make \( G(T)\epsilon_n \) vanish by choosing \( k \) and the \( \gamma_i \) appropriately, thus obtaining \( s = \sum_{i=0}^{k} \gamma_i x_{n+i} \), with \( \sum_{i=0}^{k} \gamma_i = 1 \). As a preliminary to this, we recall the following definitions and theorems; see, for example, Householder (15, chapter 1, p. 18):

**Definition 3.1.** Given a matrix \( K \in \mathbb{C}^{N \times N} \), the monic polynomial \( Q(z) \) is said to be a minimal polynomial of \( K \)

\footnote{It should be understood that, because they depend on \( n \) and \( k \), the scalars \( \gamma_i \) actually stand for \( \gamma_i^{(n,k)} \), as mentioned in the Introduction. When necessary, we will use the notation \( \gamma_i^{(n,k)} \) in the sequel. When no confusion develops, our notation will be \( \gamma_i \), for short.} if \( Q(K) = 0 \) and \( Q(z) \) has smallest degree. (0 stands for the zero \( N \times N \) matrix.)

**Theorem 3.2.** Given a matrix \( K \in \mathbb{C}^{N \times N} \), the minimal polynomial of \( K \), is unique, and divides the characteristic polynomial of \( K \). [Thus, the degree of \( Q(z) \) is at most \( N \), and all its zeros are eigenvalues of \( K \).]

**Definition 3.3.** Given a matrix \( K \in \mathbb{C}^{N \times N} \) and a nonzero vector \( u \in \mathbb{C}^N \), the monic polynomial \( P(z) \) is said to be a minimal polynomial of \( K \) with respect to \( u \) if \( P(K)u = 0 \) and \( P(z) \) has smallest degree. (0 stands for the zero \( N \)-vector.)

**Theorem 3.4.** (1) Given a matrix \( K \in \mathbb{C}^{N \times N} \) and a nonzero vector \( u \in \mathbb{C}^N \), the minimal polynomial of \( K \) with respect to \( u \) exists, is unique, and divides the minimal polynomial of \( K \), which divides the characteristic polynomial of \( K \). [Thus, the degree of \( P(z) \) is at most \( N \), and its zeros are some or all of the eigenvalues of \( K \).] (2) In addition, if \( P(z) \) is any other polynomial for which \( P(K)u = 0 \), then \( P(z) \) divides \( \tilde{P}(z) \).

Going back to Equation (17), we can make the vector \( G(T)\epsilon_n \) vanish if we choose the polynomial \( G(z) \) to be a constant multiple of \( P(z) \), the minimal polynomial of \( T \) with respect to \( \epsilon_n \) which exists and is unique by part (1) of Theorem 3.4, because \( P(T)\epsilon_n = 0 \). Thus, choosing \( G(z) = P(z)/P(1) \) so that \( G(1) = 1 \) as required, we can obtain from Equations (15)–(17) that
\[
s_{n,k} = \sum_{i=0}^{k} \gamma_i x_{n+i} = s,
\]
with
\[
\sum_{i=0}^{k} \gamma_i = 1.
\]
Here, division by \( P(1) \) is allowed because \( P(1) \neq 0 \) because 1 is not an eigenvalue of the matrix \( T \).

We can obtain the same result in an indirect way also by working backward from \( P(T)\epsilon_n = 0 \). Let
\[
P(z) = \sum_{i=0}^{k} c_i z^i; \quad c_k = 1.
\]
Starting with \( P(T)\epsilon_n = 0 \), and invoking Equation (13), we have
\[
\sum_{i=0}^{k} c_i T^i \epsilon_n = \sum_{i=0}^{k} c_i \epsilon_{n+i} = 0.
\]
which by Equation (9), can be rewritten as
\[
\sum_{i=0}^{k} c_i (x_{n+i} - s) = 0.
\]
Solving this for $s$, we have
\[ s = \sum_{i=0}^{k} \gamma_i x_{n+i}. \]
Here, division by \( \sum_{i=0}^{k} c_i \) is allowed because
\[ \sum_{i=0}^{k} c_i = P(1) \neq 0 \]
because 1 is not an eigenvalue of the matrix $T$. Thus, we have shown once more that
\[ s = \sum_{i=0}^{k} \gamma_i x_{n+i}, \]
with
\[ \sum_{i=0}^{k} \gamma_i = 1. \]

Also, the $\gamma_i$ of the preceding paragraph are now identified as $\gamma_i = c_i / \sum_{j=0}^{k} c_j$, $i = 0, 1, \ldots, k$, the $c_i$ being the coefficients of the minimal polynomial of $T$ with respect to $\epsilon_n = x_n - s$.

What remains is to find the $c_i$. Invoking $c_k = 1$, Equation (19) can be written as
\[ \sum_{i=0}^{k-1} c_i \epsilon_{n+i} = -\epsilon_{n+k}. \]
This is a set of $N$ linear equations in the $k$ unknowns $c_0, c_1, \ldots, c_{k-1}$, with $c_k = 1$. Also, because $k \leq N$, this set is in general overdetermined. Nevertheless, it is consistent and has a unique solution because $P(z)$ exists and is unique. However, to determine the $c_i$ from $\sum_{i=0}^{k-1} c_i \epsilon_{n+i} = -\epsilon_{n+k}$, it seems that we need to know the vectors $\epsilon_n = x_n - s$, $m = n, n + 1, \ldots, n + k$, hence the solution $s$. Fortunately, this is not the case, and we can obtain the $c_i$ solely from our knowledge of the vectors $x_m$. This we achieve as follows: Multiplying Equation (19) by $T$, and invoking Equation (12), we have
\[ 0 = \sum_{i=0}^{k} c_i T \epsilon_{n+i} = \sum_{i=0}^{k} c_i \epsilon_{n+i+1}. \]
Subtracting Equation (19) from Equation (20), we obtain
\[ 0 = \sum_{i=0}^{k} c_i (\epsilon_{n+i+1} - \epsilon_{n+i}) = \sum_{i=0}^{k} c_i (x_{n+i+1} - x_{n+i}), \]
and hence the linear system
\[ \sum_{i=0}^{k} c_i u_{n+i} = 0. \]
Invoking Equation (11), Equation (21) can also be expressed as
\[ 0 = \sum_{i=0}^{k} c_i T^i u_n = P(T) u_n. \]
Actually, using $P(T) u_n = 0$ and the fact that $(T^{-1})$ is nonsingular, we can apply part (2) of Theorem 3.4 to prove the following stronger result:

**Theorem 3.5.** If $u_n = x_{n+1} - x_n \neq 0$, then $P(z)$, the minimal polynomial of $T$ with respect to $\epsilon_n$, is also the minimal polynomial of $T$ with respect to $u_n$.

By Theorem 3.5, even though it is in general overdetermined, the system in Equation (21) is consistent and has a unique solution for the $c_i$. As already mentioned, its matrix is obtained solely from the vectors $x_m$, which are available. Setting $c_k = 1$, we solve this system for $c_0, c_1, \ldots, c_{k-1}$. Following that, with $c_k = 1$, we set
\[ \gamma_i = \frac{c_i}{\sum_{j=0}^{k} c_j}, \quad i = 0, 1, \ldots, k. \]
Again, this is allowed because
\[ \sum_{i=0}^{k} c_i = P(1) \neq 0 \]
by the fact that $I - T$ is not singular, and hence $T$ does not have 1 as an eigenvalue.

Summing up, we have shown that if $k$ is the degree of the minimal polynomial of $T$ with respect to $\epsilon_n$, then there exist scalars $\gamma_0, \gamma_1, \ldots, \gamma_k$, which satisfy
\[ \sum_{i=0}^{k} \gamma_i = 1, \]
such that
\[ \sum_{i=0}^{k} \gamma_i x_{n+i} = s. \]
The $\gamma_i$ are given by
\[ \gamma_i = \frac{c_i}{\sum_{j=0}^{k} c_j}, \quad i = 0, 1, \ldots, k, \]
where the $c_i$ are the unique solution to the linear system
\[ \sum_{i=0}^{k} c_i u_{n+i} = 0 \]
with $c_k = 1$.

At this point, we note that $s$ is the solution to $(I - T)x = b$, whether $\rho(T) < 1$ or not. Thus, with the $\gamma_i$ as determined above, $s = \sum_{i=0}^{k} \gamma_i x_{n+i}$, whether $\lim_{m \to \infty} x_m$ exists or not. (Recall our discussion of antilimits in the Introduction.)
In the sequel, we shall use the notation

\[ U^{(j)}_n = [u_j | u_{j+1} | \cdots | u_{j+s}] \]  

(22)

Thus, \( U^{(j)}_n \) is an \( N \times (s+1) \) matrix, \( u_j, u_{j+1}, \ldots, u_{j+s} \) being its columns. In this notation, Equation (21) reads

\[ U^{(n)}_k \mathbf{c} = 0; \quad \mathbf{c} = [c_0, c_1, \ldots, c_k]^{\top}. \]  

(23)

Of course, dividing Equation (23) by \( \sum_{i=0}^k c_i \), and defining the \( \gamma_i \) as above, we also have

\[ U^{(n)}_k \mathbf{\hat{c}} = 0; \quad \mathbf{\hat{c}} = [\gamma_0, \gamma_1, \ldots, \gamma_k]^{\top}. \]  

(24)

So far, we have observed that \( s \) can be determined via a sum of the form \( \sum_{i=0}^k \gamma_i x_{n+i} \) with \( \sum_{i=0}^k \gamma_i = 1 \) once the minimal polynomial of \( T \) with respect to \( \epsilon_s \) has been determined, and we have seen how this polynomial can be determined. However, the degree of the minimal polynomial of \( T \) with respect to \( \epsilon_s \) can be as large as \( N \) by part (1) of Theorem 3.4. Because \( N \) can be very large in general, determining \( s \) in the way we have described here becomes prohibitively expensive as far as computation time and storage requirements are concerned. [Note that we need to store the vectors \( u_0, u_1, \ldots, u_{n+k} \) and solve the \( N \times k \) linear system in Equation (21)]. Thus, we conclude that attempting to determine \( s \) via a combination of the iteration vectors \( x_n \) cannot be a suitable way after all. Nevertheless, with some twist, we can use the framework developed thus far to approximate \( s \) effectively. To do this, we replace the degree of the minimal polynomial of \( T \) with respect to \( \epsilon_s \) by an arbitrary small integer \( k \) (in practice, we take \( k \ll N \)). This subject is described in the next two subsections.

**Derivation of MPE**

Let us choose \( k \) to be an arbitrary positive integer that is normally (much) smaller than the degree of the minimal polynomial of \( T \) with respect to \( \epsilon_s \). Clearly, the linear system in Equation (23) (with \( c_k = 1 \)) is now inconsistent and hence has no solution for \( c_0, c_1, \ldots, c_{k-1} \) in the ordinary sense. To get around this problem, we solve this system in the least-squares sense, because such a solution always exists. After that, we compute \( \gamma_0, \gamma_1, \ldots, \gamma_k \) precisely as described following Equation (21), and then compute the vector

\[ s_{n,k} = \sum_{i=0}^k \gamma_i x_{n+i} \]

as our approximation to \( s \). The resulting method is known as MPE. Clearly, MPE takes as its input only the integers \( k \) and \( n \) and the vectors \( x_n, x_{n+1}, \ldots, x_{n+k+1} \), hence it can be employed whether these vectors are generated by a linear or nonlinear iterative process.

We can summarize the definition of MPE for an arbitrary sequence of vectors \( \{x_m\} \) through the following steps:

1. Choose the integers \( k \) and \( n \), and input the vectors \( x_n, x_{n+1}, \ldots, x_{n+k+1} \). (Of course, \( k \leq N \), but \( k \ll N \) normally.)

2. Compute the vectors \( u_n, u_{n+1}, \ldots, u_{n+k} \) and form the \( N \times (k+1) \) matrix \( U^{(n)}_{k+1} \). (Recall that \( u_m = x_{m+1} - x_m \).

3. Solve the overdetermined linear system \( U^{(n)}_{k+1} \mathbf{c} = 0 \) in the least-squares sense, subject to the constraint \( \sum_{i=0}^k \gamma_i = 1 \). Here \( \mathbf{\hat{c}} = [\gamma_0, \gamma_1, \ldots, \gamma_k]^{\top} \).
4. Compute the vector

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

as approximation to \( \lim_{m \to \infty} x_m = s \).

A method that is essentially the same as RRE was proposed by Kaniel and Stein (16). In this method, the \( \gamma_i \) are computed as in Step 3 of RRE, but \( \lim_{m \to \infty} x_m = s \) is now approximated by

$$s_{n,k} = \sum_{i=0}^{k} \gamma_i x_{n+i+1}.$$  

**An Exactness Property of MPE and RRE**

From Equations (19) and (20), we realize that \( P(z) = \sum_{i=0}^{k} c_i x_i \), the minimal polynomial of \( T \) with respect to \( \epsilon_n = x_n - s \), satisfies

$$\sum_{i=0}^{k} c_i (x_{m+i} - s) = 0, \quad m = n, n+1, \ldots$$  

(25)

Consequently, it also satisfies

$$\sum_{i=0}^{k} c_i u_{m+i} = 0, \quad m = n, n+1, \ldots,$$

from which the \( c_i \) can be determined uniquely because \( k \) is the smallest possible, and MPE and RRE produce \( s \) exactly.

We can now ignore the fact the vectors \( x_m \) in the preceding subsections were generated by a linear fixed-point iterative method, and concentrate only on vectors \( x_m \) that satisfy the recursion relation in Equation (25) with minimal \( k \), and state the following exactness result:

**Theorem 3.6.** Let the vectors \( x_m \) satisfy Equation (25), with \( c_0 \neq 0 \), \( \sum_{i=0}^{k} c_i \neq 0 \), and minimal \( k \). Let \( \epsilon_m = x_m - s \) for all \( m \), and assume that the vectors \( \epsilon_n, \epsilon_{n+1}, \ldots, \epsilon_{n+k-1} \) are linearly independent. Then the following hold: (1) With \( u_m = x_{m+1} - x_m \) for all \( m \), all the sets \( \{ u_m, u_{m+1}, \ldots, u_{n+k-1} \} \), \( m = n, n+1, \ldots \) are linearly independent. (2) \( s_{n,k} \) obtained by applying MPE and RRE to the sequence \( \{ x_m \} \) exist uniquely and \( s_{n,k} = s \), for all \( m = n, n+1, \ldots \), both for MPE and for RRE.

This result can be viewed as the MPE/RRE analogue of what is known as McLeod’s theorem that concerns the vector epsilon algorithm. We will discuss McLeod’s theorem briefly in the last section.

**DETERMINANT REPRESENTATIONS**

Despite the rather complex procedures by which \( s_{n,k} \) for MPE and RRE have been defined, elegant analytical expressions for them can be given. The determinant representations in the next theorem were given by the author in Ref. 17.

**Theorem 4.1.** With \( u_m \) and \( w_m \) as in Equation (8), define the scalars \( u_{ij} \) by

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

(26)

1. The \( u_{ij} \) for MPE and RRE are the solution to the linear systems

$$u_{n+i}, u_{n+j} = 0, \quad i = 0, 1, \ldots, k-1; \quad \sum_{j=0}^{k} u_{ij} = 1,$$

for MPE

$$w_{n+i}, u_{n+j} = 0, \quad i = 0, 1, \ldots, k-1; \quad \sum_{j=0}^{k} u_{ij} = 1,$$

for RRE.

(27)

2. We have the following determinant representations for \( s_{n,k} \) from MPE and RRE:

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

(28)

where \( D(x_0, x_1, \ldots, x_k) \) is a \((k+1) \times (k+1)\) determinant defined as in

$$D(x_0, x_1, \ldots, x_k) = \begin{vmatrix} u_0 & u_1 & \cdots & u_k \\ u_0 & u_1 & \cdots & u_{0,k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k} \end{vmatrix}$$  

(29)

Note that the determinant \( D(x_n, x_{n+1}, \ldots, x_{n+k}) \) in Equation (28) is vector-valued and is defined via its expansion with respect to its first row. Thus, if \( C_i \) is the cofactor of \( v_i \) in \( D(v_0, v_1, \ldots, v_k) \) defined in Equation (29), then

$$s_{n,k} = \sum_{i=0}^{k} C_i x_{n+i} / \sum_{i=0}^{k} C_i.$$

The determinant representations above have been very useful in analyzing MPE and RRE. They have also been used in Ref. 18 in deriving recursion relations among the various \( s_{n,k} \).

**MPE and RRE in Infinite Dimensional Inner Product Spaces**

Before we continue our discussion of MPE and RRE, it is important to note that these two methods, and other vector extrapolation methods as well, can be applied to vector sequences in infinite dimensional inner product spaces, such as Hilbert spaces, once the inner product in \( \mathbb{C}^N \) has been replaced with that pertinent to the space in which we
are working. This is clear from Equations (26–29). All the developments that we discuss in the next sections, including the convergence theory, remain unchanged in these spaces, subject to some minor modifications. This has been noted in Refs. 14 and 17.

**MPE AND RRE AND KRYLOV SUBSPACE METHODS**

When applied to the solution of the linear system \((I - T)x = b\) in conjunction with the linear fixed-point iteration scheme \(x_{m+1} = Tx_m + b, m = 0, 1, \ldots\). MPE and RRE become mathematically (but not algorithmically) equivalent to two well-known Krylov subspace methods, namely, the method of Arnoldi (8) and GMRES (9). The details follow:

Given a matrix \(S\) and a vector \(u\), let us define the Krylov subspace \(K_m(S; u)\) via

\[
K_m(S; u) = \text{span} \{ Su, i = 0, 1, \ldots, m - 1 \}.
\]

In the method of Arnoldi and GMRES, one starts with the vector \(x_0\) and seeks an approximation \(s_k\) to \(s\), the solution to \(Ax = b\), of the form \(s_k = x_0 + z\), with \(z\) in the Krylov subspace \(K_k(A; r_0)\), and \(r_0 = r(x_0) = b - Ax_0\), such that

\[
(r(s_k), a) = 0 \quad \text{for all } a \in K_k(A; r_0) \quad \text{(for Arnoldi method)}
\]

and

\[
\|r(s_k)\| = \min_{z \in K_k(A; r_0)} \|r(x_0 + z)\| \quad \text{(for GMRES)}.
\]

Here, \((x, y) = x^*y\) and \(\|x\| = \sqrt{x^*x}\) as before. The following theorem was proved in Ref. 19:

**Theorem 5.1.** Consider the linear system \(Ax = b\), where \(A = I - T\), and let \(x_0\) be an arbitrary initial vector. Let the vectors \(x_m\) in MPE and RRE be generated via \(x_{m+1} = Tx_m + b\), and let \(s_{n,k}^{\text{MPE}}\) and \(s_{n,k}^{\text{RRE}}\) be the approximations to the solution \(s\) of \(Ax = b\). Similarly, let \(s_{n,k}^{\text{Arnoldi}}\) and \(s_{n,k}^{\text{GMRES}}\) be the approximations to \(s\) from the method of Arnoldi and from GMRES, respectively, starting with the same initial vector \(x_0\), as explained above. Then there hold

\[
\begin{align*}
\text{MPE:} & \quad s_{n,k}^{\text{MPE}} = s_{n,k}^{\text{Arnoldi}} \\
\text{RRE:} & \quad s_{n,k}^{\text{RRE}} = s_{n,k}^{\text{GMRES}}.
\end{align*}
\]

Recall also that the Arnoldi method reduces (mathematically) to the method of conjugate gradients when the matrix \(A\) is hermitian positive definite.

**EFFICIENT IMPLEMENTATION OF MPE AND RRE**

**General Considerations**

We now turn to the numerical implementation of MPE and RRE in floating-point (that is, in finite-precision) arithmetic. Because we are performing computations with vectors and matrices, this topic must be handled with care. In this section, we provide the details of some efficient algorithms for implementing MPE and RRE. In addition to being stable numerically, these algorithms are also timewise economical and have minimal core memory requirements. They were originally developed in Ref. 20, where a FORTRAN 77 code is also included. By following the instructions of this section, the user can program these algorithms himself without having to invoke commercial software packages.

Our algorithms are based on the definitions of MPE and RRE that we presented in the section entitled "Derivation of MPE and RRE." One feature common to both MPE and RRE is that the matrices \(U_k^{(n)}\) enter their definition. In addition, by the fact that \(\sum_{i=0}^{k} \gamma_i = 1\), the vector \(s_{n,k} = \sum_{i=0}^{k} \gamma_i s_{n+i}\), for both MPE and RRE, can be expressed as in

\[
s_{n,k} = x_n + \sum_{i=0}^{k-1} \gamma_i u_{n+i} = U_k^{(n)} \xi;
\]

where

\[
\xi = [\zeta_0, \zeta_1, \ldots, \zeta_{k-1}]^T,
\]

Taking this point into account together with the definitions of MPE and RRE, we realize first that the vectors \(s_{n+1}, \ldots, s_{n+k+1}\) are not needed directly for computing \(s_{n+k}\). Actually, they can be discarded as soon as the \(u_{n+i}\) are computed, which saves storage. This is done by overwriting the vector \(u_{n+i}\) with \(u_{n+i} = \Delta u_{n+i}\) as soon as the latter is computed, for \(i = 1, \ldots, k\). We save only \(x_n\).

In the sequel, we assume that the matrix \(U_k^{(n)}\) has full rank, that is, that rank \(U_k^{(n)} = k + 1\).

One very important aspect of our algorithms is their accurate solution of the relevant least-squares problems, via QR factorizations of the matrices \(U_k^{(n)}\), as in

\[
U_k^{(n)} = Q_k R_k.
\]

Here \(Q_k\) is an \(N \times (k + 1)\) unitary matrix satisfying \(Q_k^* Q_k = I_{(k+1) \times (k+1)}\). Thus, \(Q_k\) has the columnwise partition

\[
Q_k = [q_0 | q_1 | \cdots | q_k],
\]

such that the columns \(q_i\) form an orthonormal set of \(N\)-dimensional vectors, that is, \(q_i^* q_j = \delta_{ij}\). The matrix \(R_k\) is a \((k + 1) \times (k + 1)\) upper triangular matrix with positive diagonal elements. Thus,

\[
R_k = \begin{bmatrix}
r_{00} & r_{01} & \cdots & r_{0k} \\
r_{10} & r_{11} & \cdots & r_{1k} \\
& \ddots & \ddots & \ddots \\
& & r_{k0} & r_{kk}
\end{bmatrix}; \quad r_{ii} > 0, \ i = 0, 1, \ldots, k.
\]
This factorization can be carried out easily and accurately using the modified Gram–Schmidt orthogonalization process (MGS). See, for example, Refs. 20 and 21. For completeness, we give here the steps of MGS as applied to the matrix \( U_k^{(n)} \):

1. Compute \( r_{00} = \| u_0 \| \) and \( q_0 = u_0 / r_{00} \).
2. For \( i = 1, \ldots, k \) do
   - Set \( u_i^{(0)} = u_{ni+i} \).
   - For \( j = 0, \ldots, i - 1 \) do
     - \( r_{jk} = (q_j, u_i^{(j)}) \) and \( u_i^{(j+1)} = u_i^{(j)} - r_{jk} q_j \)
   - Compute \( r_{ii} = \| u_i^{(i)} \| \) and \( q_i = u_i^{(i)}/r_{ii} \)
3. With the QR factorization of \( U_k^{(n)} \), obtain the vectors \( u_i \) in Step 1, while in Step 2, \( u_i^{(0)} \) overwrites \( u_{ni+i} \), \( u_i^{(j+1)} \) overwrites \( u_i^{(j)} \), and \( q_i \) overwrites \( u_i^{(i)} \), as soon as they are computed. Thus, for each \( i = 1, \ldots, k \), the vectors \( u_0, u_1, \ldots, u_k \) all occupy the same storage location both at the time of computation and after the computation has been performed.

Summarizing, we have that at all stages of the computation of \( Q_k \) and \( R_k \), we are keeping only \( k + 2 \) vectors in the memory. Ultimately, we save the vector \( x_0 \) and the matrix \( Q_k \), that is, the vectors \( q_0, q_1, \ldots, q_k \).

Note that \( Q_k \) is obtained from \( Q_{k-1} \) by appending to the latter the vector \( q_k \) as the \((k + 1)\)st column. Similarly, \( R_k \) is obtained from \( R_{k-1} \) by appending to the latter the \( 0 \) vector as the \((k + 1)\)st row and then the vector \( [r_{0k}, r_{1k}, \ldots, r_{kk}]^T \) as the \((k + 1)\)st column.

### Algorithms for MPE and RRE

With the QR factorization of \( U_k^{(n)} \) (hence of \( U_k^{(n)} \)) available, we can give algorithms for MPE and RRE within a unified framework. The algorithms are almost identical; they differ only in the way they compute the \( \gamma_i \). Here are the their steps:

1. Input: \( k \) and \( n \) and the vectors \( x_n, x_{n+1}, \ldots, x_{n+k-1} \).
2. Compute the vectors \( u_{ni+i} = \Delta x_{ni+i}, i = 0, 1, \ldots, k \), and form the \( N \times (k + 1) \) matrix
   \[
   U_k^{(n)} = [u_0 | u_{n+1} | \cdots | u_{n+k}].
   \]
   and form its QR factorization, namely, \( U_k^{(n)} = Q_k R_k \), with \( Q_k \) and \( R_k \) as in Equations (36) and (37), using MGS.
3. Determine the \( \gamma_i \):
   - For MPE
     - With \( p_k = [r_{0k}, r_{1k}, \ldots, r_{k-1}, k] \), solve the \( k \times k \) upper triangular system
     \[
     R_k x' = -p_k; \quad x' = [c_0, c_1, \ldots, c_{k-1}]^T.
     \]
     Set \( c_k = 1 \), and \( \gamma_i = c_i / \sum_{i=0}^{k} c_i, i = 0, 1, \ldots, k \), provided \( \sum_{i=0}^{k} c_i \neq 0 \).
   - For RRE
     - With \( e = [1, 1, \ldots, 1]^T \), solve the \((k + 1) \times (k + 1)\) linear system
     \[
     R_k^T R_k d = e; \quad d = [d_0, d_1, \ldots, d_k]^T.
     \]
     (This amounts to solving two triangular systems: first \( R_k^T a = e \), for \( a = [d_0, a_1, \ldots, a_k]^T \), and following that, \( R_k d = a \) for \( d \).)
     Next, compute
     \[
     \lambda = 1 / \sum_{i=0}^{k} d_i.
     \]
     (Note that \( \lambda \) is always real and positive because \( U_k^{(n)} \) has full rank.)
     Next, set \( \gamma_i = \lambda d_i \), \( i = 0, 1, \ldots, k \).
4. With the \( \gamma_i \) determined, compute \( \xi = [\xi_0, \xi_1, \ldots, \xi_{k-1}] \) via
   \[
   \xi_0 = 1 - \gamma_0; \quad \xi_j = \xi_{j-1} - \gamma_j, \quad j = 1, \ldots, k - 1.
   \]
5. Compute
   \[
   \eta = [\eta_0, \eta_1, \ldots, \eta_{k-1}] = R_{k-1} \xi.
   \]
   Then compute
   \[
   s_{nk} = x_n + Q_{k-1} \eta = x_n + \sum_{i=0}^{k-1} \eta_i q_i.
   \]

Note that the linear systems that we solve in Step 3 are very small in size compared with the dimension \( N \) of the vectors \( x_{ni} \), hence their cost is negligible.

### Error Estimation via Algorithms

We now turn to the assessment of the quality of \( s_{nk} \) as an approximation to \( s \), in case the sequence \( \{x_{ni}\} \) is obtained from the fixed-point iteration \( x_{n+1} = f(x_n) \) of the system \( x = f(x) \). All we know to compute is \( f(x) \) for a given \( x \), and we must use this knowledge for our assessment.

One common way of assessing the quality of a given vector \( x \) as an approximation to \( s \) is by looking at the \( l_2 \)-norm of the residual vector \( r(x) = f(x) - x \) because \( r(s) = 0 \). Thus, the quality of \( s_{nk} \) will be assessed by computing the \( l_2 \)-norm of \( r(s_{nk}) \). What is of interest is the size of \( \|r(s_{nk})\| / \|r(x_0)\| \), namely, the order of magnitude of \( \|r(s_{nk})\| \) relative to that of \( \|r(x_0)\| \). This number shows by how many orders of magnitude the \( l_2 \)-norm of the initial residual \( r(x_0) \) (or of the initial error \( e_0 = x_0 - s \)) has been reduced in the course of extrapolation.

We now show how this can be done by using only some of the quantities already produced by the algorithms given above, at no additional cost.
First, note that, by Equation (2),
\[ r(x_0) = f(x_0) - x_0 = x_1 - x_0 = u_0, \quad \text{hence} \quad ||r(x_0)|| = ||u_0||. \]

- For linear sequences: When the iteration vectors \( x_n \) are generated linearly as in Equation (7), we have \( r(x) = (Tx + b) - x. \) Thus,
\[ r(x_m) = x_{m+1} - x_m = u_m. \]

Invoking also \( \sum_{i=0}^{k} \gamma_i = 1, \) where \( \gamma_i \) are as obtained when applying MPE or RRE, we therefore have
\[ r(s_{n,k}) = \sum_{i=0}^{k} \gamma_i u_{n+i} = U^{(n)}_k \gamma, \quad \text{hence} \quad ||r(s_{n,k})|| = \|U^{(n)}_k \gamma\|. \]

Note also that, in this case,
\[ s_{n,k} = s = -(I - T)^{-1} r(s_{n,k}), \quad \text{hence} \]
\[ ||s_{n,k} - s|| \leq \|(I - T)^{-1}\| \|r(s_{n,k})||, \]

and this provides additional justification for looking at \( ||r(s_{n,k})||. \)

For nonlinear sequences: When the \( x_n \) are generated nonlinearly as in Equation (2), with \( s_n \) close to \( s, \) we have
\[ r(s_{n,k}) = f(s_{n,k}) = s_{n,k} \approx U^{(n)}_k \gamma, \quad \text{hence} \]
\[ ||r(s_{n,k})|| \approx \|U^{(n)}_k \gamma\|. \]

Whether the vectors \( x_n \) are generated linearly or nonlinearly, \( \|U^{(n)}_k \gamma\| \) can be determined in terms of already computed quantities and at no cost, without actually having to compute \( r(s_{n,k}). \) Indeed, we have
\[ \|U^{(n)}_k \gamma\| = \begin{cases} r_{kk}/|k| & \text{for MPE} \\ \sqrt{k} & \text{for RRE} \end{cases} \]

Here, \( r_{kk} \) is the last diagonal element of the matrix \( R_k \) in Equation (37) and \( \lambda \) is the parameter computed in Step 3 of the algorithms in the preceding subsection. Clearly, this computation of \( \|U^{(n)}_k \gamma\| \) can be made a part of the algorithms. For details, see Ref. 20.

**ERROR ANALYSIS FOR MPE AND RRE**

The error and convergence analysis of MPE and RRE has been carried out within the framework of linearly generated vector sequences \( \{x_n\} \) in Refs. 17 and 22–25. This analysis sheds considerable light on what vector extrapolation methods can achieve when they are applied to vector sequences obtained from fixed-point iterative methods on nonlinear systems as well as linear ones.

**Convergence Acceleration Properties of \( s_{n,k} \) as \( n \to \infty \)**

We first consider the convergence properties of MPE and RRE as \( n \to \infty \) while \( k \) is held fixed. The following result that shows that these methods are bona fide convergence acceleration methods was given in Ref. 17.

The technique of the proof is based on that developed by Sidi et al. (14).

**Theorem 7.1.** Assume that the vectors \( x_n \) satisfy
\[ x_n = s + \sum_{i=1}^{p} v_i \lambda_i^{m} n, \quad p \leq N, \]
where the vectors \( v_i \) are linearly independent, and the scalars \( \lambda_i \) are distinct and nonzero and \( \lambda_i \neq 1, \) ordered as in
\[ |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_p|. \]

If also
\[ |\lambda_k| \geq |\lambda_{k+1}|, \]
then the following hold:

1. \( s_{n,k} \) exists for all large \( n \) and \( s_{n,k} \) exists unconditionally for all \( n. \)
2. Both for MPE and RRE, there holds
\[ s_{n,k} - s = (C_{n,k} + O(1)) \lambda_k^{n} = O(\lambda_k^{n}) \text{ as } n \to \infty, \]
where the vectors \( C_{n,k} \) are bounded in \( n, \) that is, \( \sup_n \|C_{n,k}\| < \infty, \) and are the same for MPE and RRE. Thus, the errors \( s_{n,k} - s \) for MPE and RRE are asymptotically equal as \( n \to \infty \) as well.
3. In addition, \( \lim_{n \to \infty} (n,k), \) where \( (n,k) \) stands for \( \gamma_i, \) all exist, and
\[ \lim_{n \to \infty} \sum_{i=0}^{k} \gamma_i^{(n,k)} \lambda_i = \prod_{i=1}^{k} \lambda - \lambda_i. \]

4. Finally, for \( k = p, \) we have the so-called "finite termination property," that is,
\[ s_{n,p} = s \quad \text{and} \quad \sum_{i=0}^{p} \gamma_i^{(n,p)} \lambda_i = \prod_{i=1}^{p} \lambda - \lambda_i. \]

It can be shown that the finite termination property in part (4) of Theorem 7.1 is nothing but the exactness result we stated in Theorem 3.6.

Note that when the \( x_n \) are generated by the iterative procedure in Equation (7), where \( I - T \) is nonsingular and \( T \) is diagonalizable, they are precisely as described in this theorem, with \( \lambda_i \) being some or all of the distinct nonzero eigenvalues of \( T \) and \( \gamma_i \) being eigenvectors corresponding to these eigenvalues. If \( (\mu_i, z_i), \) \( i = 1, \ldots, N, \) are the eigenpairs of \( T, \) that is, \( T z_i = \mu_i z_i, \) for every \( i, \) then \( x_0 - s = \sum_{i=1}^{N} \alpha_i z_i \) for some scalars \( \alpha_i, \) since the \( z_i \) are linearly independent and span \( C^N. \) Therefore, by Equations (9) and (13),
\[ x_n - s = T^{m}(x_0 - s) = \sum_{i=1}^{N} \alpha_i \mu_i^{m} z_i. \]
Then, for $m \geq 1$, zero eigenvalues of $T$ have zero contribution to this summation. In addition, the contributions of equal nonzero eigenvalues can be lumped together to form one single term that is again an eigenvector of $T$. Renaming the distinct eigenvalues that contribute to this sum $\lambda_i$, we obtain precisely the structure of $x_m$ in the theorem. In addition, note that $p$ is the degree of the minimal polynomial of $T$ with respect to $x_m - s$ for all $m \geq 1$ in this case. This is also borne out by Equation (43).

It is also clear from Theorem 7.1 that the sequence $\{x_m\}$ there need not result from a nonsingular linear system $(I - T)x = b$, it can result from some other type of problem as well. In case, the condition $|\lambda_i| > |\lambda_{k+1}|$ in Theorem 7.1 does not hold [then $|\lambda_i| = |\lambda_{k+1}|$ necessarily since $|\lambda_i| \geq |\lambda_{k+1}|$ by the ordering of the $\lambda_i$ in Equation (41)], we need to modify the statement of this theorem. This has been done in Ref. 23, and we state a special case of it next.

**Theorem 7.2.** Assume that the vectors $x_m$ have been generated by the iterative procedure $x_{m+1} = T x_m + b$, $T$ being diagonalizable and $I - T$ being nonsingular. Denote the solution of $(I - T)x = b$ by $s$. Let us order the nonzero distinct eigenvalues $\lambda_i$ of $T$ that contribute to $x_m - s$ via Equation (38) as in Equation (39). If $|\lambda_i| = |\lambda_{k+1}|$, then $s_{n,k}$ exist and are unique for all $n$ and $k$, and there holds

$$s_{n,k} - s = O(\lambda_{r+1}^n) = O(\lambda_{r+1}^m) \quad \text{as } n \to \infty, \quad (44)$$

(1) for RRE unconditionally and (2) for MPE provided the hermitian part of the matrix $\alpha A$, where $\alpha$ is some appropriate nonzero scalar in $\mathbb{C}$ and $A = I - T$, is positive definite.

One conclusion that can be drawn from Theorem 7.2 is that if

$$|\lambda_r| > |\lambda_{r+1}| > \ldots > |\lambda_{r+v+1}| \geq \ldots,$$

then there holds

$$s_{n,k} - s = O(\lambda_{r+1}^n) \quad \text{as } n \to \infty, \quad \text{for } r + 1 \leq k \leq r + v - 1.$$

Thus, if

$$|\lambda_1| = \cdots = |\lambda_r| > |\lambda_{r+1}| \geq \cdots,$$

then

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

while

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

We would like to emphasize here that the results of Theorems 7.1 and 7.2 are not the same. For example, in Theorem 7.1, we have that $s_{n,k}^{MPE} - s$ and $s_{n,k}^{RRE} - s$ are asymptotically equal as $n \to \infty$, whereas no such result holds in Theorem 7.2.

Concerning the results of Theorems 7.1 and 7.2, we can offer the following interpretation: Starting from Equation (38), we first observe that the contribution of the small $\lambda_i$ 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 $\lambda_i$. What remains is the contribution from the intermediate $\lambda_i$, starting with $\lambda_{k+1}$.

In connection with Theorems 7.1 and 7.2, we note that we have not assumed that $\lim_{n \to \infty} x_m$ exists. Clearly, $\lim_{n \to \infty} x_m$ exists and is equal to $s$ provided $|\lambda_1| < 1$. In case $\lim_{n \to \infty} x_m$ does not exist, we have that $|\lambda_1| \geq 1$ necessarily; in this case, $s$ is the antilimit of $\{x_m\}$. Now, if we use the $x_m$, as they are, to approximate $s$, we have the error

$$\epsilon_m = x_m - s = O(\lambda_{1}^m) \quad \text{as } m \to \infty.$$  

Thus, provided $|\lambda_{k+1}| < 1$, we have that $\lim_{n \to \infty} s_{n,k} = s$, whether $\lim_{n \to \infty} x_m$ exists. Furthermore, when the sequence $\{x_m\}$ converges (to $s$), $s_{n,k}$, converges (to $s$) faster, provided $|\lambda_{k+1}| < |\lambda_1|$, which clearly holds in Theorem 7.1. This means that MPE and RRE accelerate the convergence of the sequence $\{x_m\}$.

For more general versions of Theorems 7.1 and 7.2 involving nondiagonalizable matrices $T$, see Refs. 22 and 23.

**Error Analysis for $s_{n,k}$ with Fixed $n$ and $k$**

So far, we have reviewed the convergence properties of the approximations $s_{n,k}$ for $n \to \infty$ while $k$ is being held fixed. Of course, $n$ cannot be increased indefinitely in practice. Similarly, $k$ cannot be increased too much because of storage limitations [recall from the algorithms of the sixth section that we need to store the $N \times (k + 1)$ matrix $Q_k$]. Because of these reasons, MPE and RRE are applied in the so-called cycling mode (to be discussed later in the eighth section), where $n$ and $k$ can be chosen moderately large but are both kept fixed. Bounds on the error in $s_{n,k}^{MPE}$ and $s_{n,k}^{RRE}$, as functions of $n$ and $k$, have been presented by Sidi and Shapira (24, 25).

The next result from Refs. 24 and 25 provides an error bound on $s_{n,k}^{RRE}$ 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 $n = 0$. An essentially identical result for $s_{n,k}^{MPE}$ under the condition that the matrix $I - T$ has a positive definite hermitian part (cf. Theorem 7.2), is given in Ref. 24.

**Theorem 7.3.** Let $s$ be the solution to the linear system $(I - T)x = b$, where $I - T$ is nonsingular. Assume that $T$ is

---

8The hermitian part of a matrix $K \in \mathbb{C}^{n \times n}$ is the matrix $K_{H} = \frac{1}{2}(K + K^*)$.

9Recall that $s_{r} - s = O(\lambda_{r}^n)$ as $n \to \infty$, and that $s_{r+v} - s = O(\lambda_{r+v+1}^n)$ as $n \to \infty$, already as described in Theorem 7.1.
and let the vector sequence $\{x_m\}$ be generated via $x_{m+1} = Tx_m + b$, $m = 0, 1, \ldots$ Define the residual vector associated with the vector $x$ by $r(x) = b - (1 - T)x$. Let also $P_k = \{p(z) : p \in \mathbb{P}_k, p(1) = 1\}$. Then

$$
\|r(s_{n,k})\| = \min_{p \in P_k} \|T^n p(T)(r(0))\| \\
\leq \left( \min_{p \in P_k} \|T^n p(T)\| \right) \|r(0)\| \\
\leq \|T^n p(T)\| \|r(0)\| \quad \text{for every } p \in P_k \\
\leq \kappa(R)^\ast \|T^n p(T)\| \|r(0)\|, 
$$

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

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

Remark. The paper (25) actually treats GMRES $(k)$, the restarted GMRES, with $n$ initial iterations of Richardson type, which is denoted there GMRES$(n,k)$. Because RRE and GMRES are equivalent in the sense described in Theorem 5.1 in the fifth section, the results of Ref. 25 can be expressed as in Theorem 7.3 and without any changes.

Now, in case spect $(T) \subset D$, we have that $\Gamma_{n,k}^{\text{spect}(T)} \leq \Gamma_{n,k}^D$. This fact has been used in Ref. 24 to derive upper and lower bounds on $\Gamma_{n,k}^{\text{spect}(T)}$ that can be expressed analytically in terms of suitable sets of orthogonal polynomials. For some special types of spectra, both bounds can be expressed analytically in terms of the Jacobi polynomials $P_k^{(n,\beta)}(z)$ and can easily be computed numerically. In addition, these bounds are tight. (See the tables in Refs. 24 and 25.) Below, by $[u,v]$ we mean the straight line segment between the (complex) numbers $u$ and $v$ in the $z$-plane. The Jacobi polynomials $P_k^{(n,\beta)}(z)$ are normalized such that $P_k^{(n,\beta)}(1) = \binom{k + \alpha}{k}$; see Abramowitz and Stegun (26), for example.

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

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

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 spect $(T)$ is contained in $[-\beta, \beta]$ for some possibly complex $\beta$ (1 is not in $[-\beta, \beta]$), then

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

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

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

where

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

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 $\Gamma_{n,k}$ tends to zero monotonically as a function of $k$, its rate of decrease becoming larger with increasing $n$, at the best possible rate, because now

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

When comparing two approximations $s_{n,k}^{\text{RE}}$ and $s_{n,k}^{\text{RRE}}$, what seems to determine which one is more accurate is the corresponding upper bounds for $\Gamma_{n,k}^{\text{spect}(T)}$ and $\Gamma_{n,k}^{\text{spect}(T)}$. The examples we have given here suggest that $s_{n,k}^{\text{RE}}$ is the better one if $n > n'$. In particular, $s_{n,k}^{\text{RE}}$, with $n > 0$ is likely to be better than $s_{n,k}^{\text{RRE}}$ (equivalently, GMRES$(n,k)$ with $n > 0$ is better than GMRES$(k)$). We can make use of this observation in applying RRE (and MPE as well) in the cycling mode; this will be discussed in the eighth section.

Numerical Stability of MPE and RRE

An important issue related to the application of extrapolation methods in finite-precision arithmetic is that of numerical stability. This issue concerns the propagation of input errors into the output. In our case, this means the propagation of errors in the $x_m$ into $s_{n,k}$. A practical measure of this is the quantity

$$
\Gamma_{n,k} = \sum_{i=0}^k |\eta_i^{(n,k)}|. 
$$

Clearly, $\Gamma_{n,k} \geq 1$ because $\sum_{i=0}^k |\eta_i^{(n,k)}| = 1$.

This assertion can be justified heuristically as follows: Let us assume that $\eta_n$ is the error committed in computing
\( \mathbf{x}_m \) (which can be the result of roundoff, for example). Thus, \( \mathbf{s}_m = \mathbf{x}_m + \eta_m \) is the computed \( \mathbf{x}_m \). Assuming that the \( \gamma_i^{(n,k)} \) are not affected by these errors, \( \mathbf{s}_{n,k} \), the “computed” \( \mathbf{s}_{n,k} \), is given by

\[
\mathbf{s}_{n,k} = \sum_{i=0}^{k} \gamma_i^{(n,k)} \mathbf{x}_{n+i} = \sum_{i=0}^{k} \gamma_i^{(n,k)} (\mathbf{x}_{n+i} + \eta_{n+i}) = \mathbf{s}_{n,k} + \sum_{i=0}^{k} \gamma_i^{(n,k)} \eta_{n+i}.
\]

(49)

Therefore,

\[
\| \mathbf{s}_{n,k} - \mathbf{s}_{n,k} \| = \| \sum_{i=0}^{k} \gamma_i^{(n,k)} \eta_{n+i} \| = \sum_{i=0}^{k} \| \gamma_i^{(n,k)} \| \| \eta_{n+i} \|
\]

\[
\leq \Gamma_{n,k} \max_{0 \leq i \leq k} \| \eta_{n+i} \|.
\]

(50)

Let us now assume that the relative errors in the computed \( \mathbf{x}_m \) are bounded by some fixed positive scalar \( \beta \); that is, \( \| \eta_m \| \leq \beta \| \mathbf{x}_m \| \). Let us assume, in addition, that \( \{ \mathbf{x}_m \} \) converges, so that \( \max_{0 \leq i \leq k} \| \eta_{n+i} \| \) is bounded. Then

\[
\| \mathbf{s}_{n,k} - \mathbf{s}_{n,k} \| \leq \beta \Gamma_{n,k} \left( \max_{0 \leq i \leq k} \| \mathbf{s}_{n,k} \| \right).
\]

(51)

Now, \( \beta \) and \( \max_{0 \leq i \leq k} \| \eta_{n+i} \| \) are independent of the extrapolation method, hence are fixed. Because \( \{ \mathbf{x}_m \} \) converges, \( \max_{0 \leq i \leq k} \| \eta_{n+i} \| \) is of the order of \( \| \mathbf{s} \| \) and \( \| \mathbf{s}_{n,k} \| \). Therefore, for all practical purposes, \( \beta \Gamma_{n,k} \) is the relative error in \( \mathbf{s}_{n,k} \).

Suppose that the \( \mathbf{x}_m \) have been computed to machine accuracy. This implies that \( \beta \) is the rounding unit of the floating-point arithmetic being used; let \( \beta \) be of order \( 10^{-p} \), for some \( p > 0 \). If now \( \Gamma_{n,k} \) is of order \( 10^q \) for some \( q > 0 \), then \( \mathbf{s}_{n,k} \) agrees with \( \mathbf{s}_{n,k} \) up to \( p - q \) decimal digits. In other words, \( q \) decimal digits have been lost in the course of computation.

In view of this discussion, we say that MPE or RRE is stable if

\[
\sup_n \Gamma_{n,k} < \infty.
\]

(52)

Let us now assume that the sequence \( \{ \mathbf{x}_m \} \) is as in Theorem 7.1. By part (3) of this theorem, the \( \gamma_i^{(n,k)} \) satisfy Equation (42) and hence \( \lim_{n \to \infty} \Gamma_{n,k} \) exists. In addition, by Theorem 1.4.3 in Sidi (27), \( \Gamma_{n,k} \) also satisfies

\[
\lim_{n \to \infty} \Gamma_{n,k} \leq \prod_{i=1}^{k} \frac{1 + |\lambda_i|}{|1 - \lambda_i|} < \infty.
\]

(53)

Thus both MPE and RRE are stable under the conditions of Theorem 7.1. Equality holds in (53) when \( \lambda_1, \ldots, \lambda_k \) are all positive or all negative; in addition, when they are all negative, \( \lim_{n \to \infty} \Gamma_{n,k} = 1 \), which is the most ideal case. When one or more of these \( \lambda_i \) are very close to 1 in the complex plane, \( \Gamma_{n,k} \) becomes very large, and this limits the accuracy that can be reached by \( \mathbf{s}_{n,k} \). This suggests that we should aim at iterative methods for which the largest eigenvalues are not too close to 1. In the absence of such an iterative scheme, we can apply MPE and RRE to the sequence \( \{ \mathbf{x}_m \} \) with some integer \( r > 1 \). By Equation (38),

\[
\mathbf{x}_{m+r} = \mathbf{s} + \sum_{i=1}^{r} \mathbf{y}(\lambda_i^r)^m \text{ as } n \to \infty.
\]

(54)

Note that, if \( |\lambda_i| < 1 \) but \( \lambda_i \) is close to 1, then \( |1 - \lambda_i^r| > |1 - \lambda_i| \) for \( r > 1 \). Thus MPE and RRE are likely to produce better accuracy when applied to \( \{ \mathbf{x}_m \} \) with some integer \( r > 1 \) than when applied to \( \{ \mathbf{x}_m \} \). For example, the approximations \( \mathbf{s}_{n,k} \) produced by applying MPE and RRE to the sequence \( \{ \mathbf{x}_m \} \), under the conditions of Theorem 7.1 and provided \( \lambda_i^r \) are distinct, are such that

\[
\mathbf{s}_{n,k} - \mathbf{s} = O(|\lambda_i^{r+1}|) \text{ as } n \to \infty
\]

(55)

and

\[
\lim_{n \to \infty} \mathbf{x}_{n+1} = \sum_{i=1}^{r} \mathbf{y}(\lambda_i^r)^m \lim_{n \to \infty} \Gamma_{n,k} \leq \prod_{i=1}^{r} \frac{1 + |\sigma_i|}{|1 - \sigma_i|};
\]

\[
\sigma_i = \lambda_i^r, \quad i = 1, 2, \ldots
\]

(56)

From Equation (55), it is clear that better stability and accuracy can be obtained using the same storage (the same \( k \)) with \( r > 1 \). We can also maintain a given level of accuracy by increasing \( r \) and decreasing \( k \) simultaneously.

### MPE AND RRE VIA CYCLING

As we have already discussed, one important issue that we confront when applying vector extrapolation methods to large-scale problems is that of storage. In the algorithms presented in the sixth section, for example, we need to store the matrix \( \mathbf{Q}_n \), which forms the bulk of the stored quantities. Using the same storage, we can obtain approximations of better accuracy by applying the extrapolation methods to the sequence \( \{ \mathbf{x}_m \} \) of the \( \{ \mathbf{x}_m \} \) instead of \( \{ \mathbf{x}_m \} \), as was pointed out at the end of the preceding section.

In this section, we discuss the strategy that is known as cycling or restarting, which we alluded to above, when MPE and RRE are being applied to the solution of the system \( \mathbf{x} = \mathbf{f}(\mathbf{x}) \) in conjunction with the iterative scheme \( \mathbf{x}_{m+1} = \mathbf{f}(\mathbf{x}_m) \), \( m = 0, 1, \ldots \). In this strategy, \( n \) and \( k \) are held fixed. Here are the steps of cycling:

1. Choose integers \( n \geq 0, \quad k \geq 1 \) and \( r \geq 1 \), and an initial vector \( \mathbf{x}_0 \).
2. Compute the vectors \( \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r \) via \( \mathbf{x}_{m+1} = \mathbf{f}(\mathbf{x}_m) \), and save

\[
\mathbf{y}_n, \mathbf{y}_{n+1}, \ldots, \mathbf{y}_{n+k}, \mathbf{y}_{n+k+1}; \quad \mathbf{y}_i = \mathbf{x}_{ri}, \quad i = 0, 1, \ldots
\]

3. Apply MPE or RRE to the vectors \( \mathbf{y}_n, \mathbf{y}_{n+1}, \ldots, \mathbf{y}_{n+k+1} \) precisely as in the sixth section, with end result \( \mathbf{s}_{n,k} \).
4. If \( \mathbf{s}_{n,k} \) satisfies accuracy test, stop.
5. Otherwise, set \( \mathbf{x}_0 = \mathbf{s}_{n,k} \), and go to Step C1.
We will call each application of Steps C1–C3 a cycle. We will also denote the \( s_{i+1} \) that is computed in the \( i \)th cycle \( s_{k,i}^{(m)} \).

The analysis of MPE and RRE, as these are being applied to nonlinear systems \( \mathbf{x} = \mathbf{f}(\mathbf{x}) \) in the cycling mode with \( n = 0 \) and \( r = 1 \), has been considered in Skelboe (28). As has been shown in Ref. 1, the results of this work are only heuristic, however. What these say is that, when \( k \) in the \( i \)th cycle is chosen to be the degree \( k \) of the minimal polynomial of \( F(s_{k,i}^{(r-1)}) \) [recall that \( \mathbf{F}(\mathbf{x}) \) is the Jacobian matrix of \( \mathbf{f}(\mathbf{x}) \) at \( \mathbf{x} \)], with respect to \( s_{k,i}^{(r-1)} \) – \( s \), the sequence \( \{s_{k,i}^{(r)}\} \) converges to \( s \) quadratically. However, we must recall that, since \( k \) can be as large as \( N \) and is not known exactly, this usage of cycling is not useful practically for large-scale problems we are interested in solving. In other words, trying to achieve quadratic convergence from MPE and RRE via cycling is not realistic. We may achieve linear but fast convergence by choosing even moderate values for \( n \) and \( k \) with cycling.

APPLICATIONS

Solution of Large-Scale Nonlinear Systems

As we mentioned earlier, one of the main uses of vector extrapolation methods has been in the solution of large-scale systems of nonlinear equations \( \psi(\mathbf{x}) = 0 \) with solution \( s \), especially those that arise from discretization of nonlinear continuum problems, via fixed-point iterative procedures of the form \( \mathbf{x}_{m+1} = \mathbf{f}(\mathbf{x}_m) \). In this connection, we would like to emphasize that vector extrapolation methods, in the cycling mode, are effective when the iteration procedure \( \mathbf{x}_{m+1} = \mathbf{f}(\mathbf{x}_m) \) chosen is convergent and reasonably good, in the sense that only a few of the eigenvalues of \( \mathbf{F}(\mathbf{s}) \), the Jacobian matrix of \( \mathbf{f}(\mathbf{x}) \) at \( \mathbf{x} = \mathbf{s} \), are near 1 in the complex plane. Vector extrapolation methods do not perform that well when most eigenvalues of \( \mathbf{F}(\mathbf{s}) \) cluster near 1 in the complex plane. Thus, the iterative method should be chosen such that the eigenvalues of \( \mathbf{F}(\mathbf{s}) \) are scattered in the interior of the unit disk, very few of them being near 1.

As discussed in the preceding section, the simplest way of achieving this goal is by applying the extrapolation methods to the subsequence \( \{\mathbf{x}_m\} \) with some integer \( r \geq 2 \). Another simple way is by defining the iteration vectors via

\[
\mathbf{x}_{m+1} = \mathbf{x}_m + w[\mathbf{f}(\mathbf{x}_m) - \mathbf{x}_m] = (1 - w)\mathbf{x}_m + w\mathbf{f}(\mathbf{x}_m),
\]

\( m = 0, 1, \ldots \),

for some suitable scalar \( w \). See, for example, Ref. 20 (section 7).

In applying MPE and RRE to finite difference solutions of elliptic PDEs, such as the two-dimensional Poisson equation \( \partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 = \phi(x, y) \), a good iterative method would be the ADI (alternating direction implicit) method. For this method, see Stoer and Bulirsch (29), Morton and Mayers (30), or Ames (31), for example.

Vector-Valued Rational Approximations

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

For this, we apply the extrapolation methods to the sequence \( \{\mathbf{x}_m(z)\} \), where

\[
\mathbf{x}_m(z) = \sum_{i=0}^m \mathbf{u}_i z^i, \quad m = 0, 1, \ldots
\]

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 (10). We give a brief informal description of the subject, in the context of MPE, next.

When MPE is used for this purpose, we obtain the approximations \( s_{n,k}(z) \) (now called SMPE approximations) that are given as in

\[
s_{n,k}(z) = \frac{D(z^k \mathbf{x}_n(z), z^{k-1} \mathbf{x}_{n+1}(z), \ldots, z^0 \mathbf{x}_{n+k}(z))}{D(z^k, z^{k-1}, \ldots, z^0)},
\]

(57)

where \( D(v_0, v_1, \ldots, v_k) \) is a \((k + 1) \times (k + 1)\) determinant defined as in

\[
D(v_0, v_1, \ldots, v_k) = \begin{vmatrix} v_0 & v_1 & \cdots & v_k \\ v_{0,0} & v_{0,1} & \cdots & v_{0,k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k} \end{vmatrix} 
\]

\( u_{i,j} = (\mathbf{u}_{n+i}, \mathbf{u}_{n+j}) \).

Thus, \( s_{n,k}(z) \) is also of the form

\[
s_{n,k}(z) = \frac{\sum_{i=0}^k c_i z^{k-i} \mathbf{x}_{n+i}(z)}{\sum_{i=0}^k c_i z^{k-i}}
\]

(59)

with appropriate scalar constants \( c_i \). Note that the numerator polynomial has degree at most \( n + k \), whereas the denominator polynomial has degree \( k \) when the cofactor of \( v_0 \) in \( D(v_0, v_1, \ldots, v_k) \) is nonzero.

The sequence \( \{s_{n,k}(z)\}_{n=0}^\infty \) (with fixed \( k \)) has a very nice convergence property, which we discuss briefly. Suppose that \( \mathbf{u}(z) \) is analytic in an open disc \( D_r = \{z \in \mathbb{C} : |z| < r\} \) and meromorphic\(^{10}\) in a larger open disc \( D_R = \{z \in \mathbb{C} : |z| < R\} \). This implies that the series \( \sum_{i=0}^\infty \mathbf{u}_i z^i \) converges to \( \mathbf{u}(z) \) only for \(|z| < r\), and diverges for \(|z| \geq r\). Under some additional condition that has to do with the Laurent expansions of \( \mathbf{u}(z) \) about its poles, the rational approximations \( s_{n,k}(z) \) obtained from the series \( \sum_{i=0}^\infty \mathbf{u}_i z^i \) 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 \( \mathbf{u}(z) \) uniformly in every compact subset of \( D_R \) excluding the poles of \( \mathbf{u}(z) \). In

\(^{10}\)A function \( \mathbf{u}(z) \) is said to be meromorphic in a domain \( D \) of the complex \( z \)-plane if its only singularities in \( D \) are poles.
being an eigenvector of being some or all of the residues being related to corresponding eigenvectors and principal vectors. The poles of \( s_{n,k}(z) \) turn out to be the so-called Ritz values resulting from the method of Arnoldi for eigenvalues. For details on precise convergence properties and rates of convergence, see Sidi (11).

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 (32). In this paper, the space we are working in is infinite dimensional, and the definitions of MPE and RRE remain unchanged, as mentioned in the fourth section on determinant representations on MPE and RRE.

### Computation of Eigenvectors of Known Eigenvalues

An immediate application of vector extrapolation in general, of MPE and RRE in particular, is to the computation of the eigenvector that corresponds to the largest eigenvalue of a large and sparse matrix when this eigenvalue is nondefective, that is, it has only corresponding eigenvectors but no corresponding principal vectors. Let \( A \in \mathbb{C}^{N \times N} \) have eigenvalues \( \mu_1, \ldots, \mu_N \), which we order as in

\[
|\mu_1| > |\mu_2| \geq |\mu_3| \geq \cdots \geq |\mu_N|.
\]

Assume that \( \mu_1 \) is known and a corresponding (unknown) eigenvector is required. To solve this problem, we choose an initial vector \( x_0 \), and perform the power iterations

\[
x_{m+1} = \mu_1^{-1} Ax_m, \quad m = 0, 1, \ldots.
\]

It is easy to show that [see, for example, Sidi (12, 33)], provided \( x_0 \) has a contribution from the eigenvalue \( \mu_1 \), the vectors \( x_m \) are precisely as in Theorem 7.1, with \( x \) there being an eigenvector of \( A \) that corresponds to \( \mu_1 \) and \( \lambda_1 \) there being some or all of \( \mu_{i+1} / \mu_i \). (Clearly, \( |\lambda| < 1 \) for all \( i \), hence \( x_m \) converges to the required eigenvector.) Thus, MPE and RRE can be applied to the sequence of power iterations \( x_m \) to accelerate its convergence. Indeed, both methods apply exactly as described in Theorem 7.1. In addition, they can be applied in the cycling mode, as described in the previous section. Vector extrapolation methods are attractive for this application because their computational cost is low due to the sparseness of the matrix involved that enables us to compute the sequence \( \{x_m\} \) via \( x_{m+1} = \mu_1^{-1} Ax_m \) inexpensively.

One recent application of this has been to the computation of the PageRank of the Google matrix. The PageRank is the (unique) eigenvector of the Google matrix that corresponds to the largest eigenvalue \( \mu_1 = 1 \) that is also known to be simple. It is positive and is normalized such that the sum of its components is 1. (See Brin and Page (34).) This problem was treated by Kann et al. (35) using a method the authors denoted quadratic extrapolation. This method has been generalized in (12, 33); quadratic extrapolation turns out to be the \( k = 2 \) case of this generalization. In addition, this generalization turns out to be closely related to MPE, in the sense that if \( s_{n,k} \) and \( s_{n,h} \) are the vectors obtained by applying MPE and the new method, respectively, to the sequence of power iterations obtained from the Google matrix \( A \), then \( s_{n,k} = s_{n,h} \). In this application, the computation of the sequence \( \{x_m\} \) via \( x_{m+1} = Ax_m \) can be achieved inexpensively because of the sparseness of the Google matrix. The use of vector extrapolation methods in general, and MPE and RRE in particular, in the cycling mode with arbitrary \( n, k, \) and \( r \) for computing the PageRank was first suggested in Ref. 12. This approach is illustrated with numerical examples in Ref. 33. The results obtained in Ref. 33 show clearly that this approach to PageRank computation is very effective.

### BRIEF SUMMARY OF MMPE

We now present a very brief discussion of the modified minimal polynomial extrapolation (MMPE) that is based on Sidi et al. (14). Without more explanation, we will be using the notation of the third section throughout.

The MMPE approximation \( s_{n,k} \) from \( \{x_m\} \), namely,

\[
s_{n,k}^{\text{MMPE}} = \sum_{i=0}^{k} \gamma_i x_{n+i},
\]

is essentially defined by requiring that the \( \gamma_i \) be the solution to the linear system

\[
(q_i, U_k^{(n)} \gamma) = 0, \quad i = 0, 1, \ldots, k-1; \quad \sum_{j=0}^{k} \gamma_j = 1,
\]

where \( q_0, q_1, \ldots, q_{k-1} \) are \( k \) fixed linearly independent vectors in \( \mathbb{C}^N \).

A determinant representation for \( s_{n,k}^{\text{MMPE}} \) has been given in Ref. 14, this representation being exactly of the form shown in Equations (28) and (29) of Theorem 4.1, with \( \gamma_{i,j} = (q_i, u_{i+j}) \) in Equation (26).

The convergence theory of the sequences \( \{s_{n,k}^{\text{MMPE}}\} \) under the conditions of Theorem 7.1 has been given in Ref. 14. This theory shows that, for all large \( n \), \( s_{n,k}^{\text{MMPE}} \) exist uniquely [part (1) of Theorem 7.1], and that the rest of the results of Theorem 7.1 (parts (2)–(4)) are valid for MMPE without any changes, provided

\[
\begin{pmatrix}
(q_0, v_1) & (q_0, v_2) & \cdots & (q_0, v_k) \\
(q_1, v_1) & (q_1, v_2) & \cdots & (q_1, v_k) \\
\vdots & \vdots & \ddots & \vdots \\
(q_{k-1}, v_1) & (q_{k-1}, v_2) & \cdots & (q_{k-1}, v_k)
\end{pmatrix} \neq 0. \tag{61}
\]
A more complete theory under conditions on \( \{x_m\} \) more general than those in Theorem 7.1, has been presented in Sidi and Bridger (22).

As we have observed, the vectors \( \mathbf{q}_i \) that enter the construction of MMPE are fixed. It is possible to replace these vectors by \( \mathbf{q}_i^{(n)} \) that depend on \( n \). We then get what the author has called generalized MPE (GMPE) in Ref. 19. Obviously, GMPE contains MPE, RRE, and MMPE with \( \mathbf{q}_i^{(n)} = \mathbf{u}_{n+i}, \mathbf{q}_i^{(n)} = \mathbf{w}_{n+i}, \) and \( \mathbf{q}_i^{(n)} = \mathbf{q}_i \), respectively. Then,

\[
\mathbf{s}_{GMPE}^{n,k} = \sum_{i=0}^{k} \gamma_i x_{n+i},
\]

with the \( \gamma_j \) defined as the solution to

\[
(q_i^{(n)}, u_k^{(n)} \gamma) = 0, \quad i = 0, 1, \ldots, k - 1; \quad \sum_{j=0}^{k} \gamma_j = 1.
\]

Of course, a determinant representation for \( \mathbf{s}_{GMPE}^{n,k} \) exists and is given as in Equations (28) and (29) of Theorem 4.1, with \( \mathbf{u}_{i,j} = (q_i^{(n)}, u_{n+j}) \) in Equation (26).

**BRIEF SUMMARY OF EPSILON ALGORITHMS**

**Scalar Epsilon Algorithm**

Let the scalar sequence \( \{x_m\} \) be such that

\[
x_m \sim s + \sum_{i=1}^{\infty} a_i \lambda_i^m \text{ as } m \to \infty,
\]

where \( a_i \) are nonzero, \( \lambda_i \) are distinct, nonzero, and \( \lambda_i \neq 1 \), such that

\[
|\lambda_1| > |\lambda_2| > \cdots; \quad \lim_{i \to \infty} \lambda_i = 0.
\]

Here \( s \) is \( \lim_{m \to -\infty} x_m \) when the latter exists; it is the antilimit of \( \{x_m\} \) otherwise. Of course, the sequence converges provided \( |\lambda_i| < 1 \). To accelerate the convergence of the sequence \( \{x_m\} \), Shankos (36) developed a nonlinear transformation, by which we define

\[
e_k(x_n) = \frac{E(x_n, x_{n+1}, \ldots, x_{n+k})}{E(1,1,1,1)},
\]

where

\[
E(u_0, u_1, \ldots, u_k) = \begin{vmatrix}
u_0 & u_1 & \cdots & u_k \\u_n & u_{n+1} & \cdots & u_{n+k} \\
u_{n+1} & u_{n+2} & \cdots & u_{n+k+1} \\\vdots & \vdots & \ddots & \vdots \\u_{n+k-1} & u_{n+k} & \cdots & u_{n+2k-1}
\end{vmatrix};
\]

\( u_i = x_{i+1} - x_i \).

Here, \( e_k(x_n) \) are approximations to \( s \). In case Equation (62) assumes the form

\[
x_m = s + \sum_{i=1}^{k} a_i \lambda_i^m, \quad m = 0, 1, \ldots,
\]

for some finite \( k \), we have \( e_k(x_n) = s \) for all \( n \). This is the exactness result for the Shanks transformation.

This transformation was proposed earlier by Schmidt (37) for constructing the solution of linear systems from fixed-point iterations; compare Equation (62) with Equation (38) satisfied by fixed-point iteration vectors generated linearly.

A very elegant and fast algorithm for constructing the \( e_k(x_n) \) was developed by Wynn (5), and it reads

\[
e_{-1}^{(n)} = 0, \quad e_0^{(n)} = x_n, \quad n = 0, 1, \ldots,
\]

\[
e_{k+1}^{(n)} = e_{k-1}^{(n)} + \frac{1}{e_k^{(n)} - e_{k-1}^{(n)}}, \quad n, k = 0, 1, \ldots.
\]

Here, \( e_{2k}^{(n)} = e_k(x_n) \) and \( e_{2k+1}^{(n)} = 1/e_k(\Delta x_n) \). Of course, \( \Delta x_n = x_{n+1} - x_n \).

A different algorithm, denoted the FS/qd algorithm, that is as efficient as the epsilon algorithm, was recently proposed by Sidi in chapter 21 of Ref. 27. In this algorithm, we first compute two sets of quantities, \( \{e_k^{(n)}\} \) and \( \{q_k^{(n)}\} \), via the qd algorithm:

\[
e_0^{(n)} = 0, \quad q_1^{(n)} = \frac{u_{n+1}}{u_n}, \quad n = 0, 1, \ldots; \quad (u_i = x_{i+1} - x_i)
\]

\[
e_k^{(n)} = q_{k+1}^{(n+1)} - q_k^{(n)} + e_k^{(n+1)}, \quad q_k^{(n+1)} = \frac{e_{k+1}^{(n+1)} - q_k^{(n)}}{e_k^{(n)}},
\]

\[
n = 0, 1, \ldots; \quad k = 1, 2, \ldots.
\]

Once the \( e_k^{(n)} \) and \( q_k^{(n)} \) have been computed, we compute the \( e_k(x_n) \) as follows:

\[
M_0^{(n)} = \frac{x_n}{u_n}, \quad N_0^{(n)} = \frac{1}{u_n}, \quad n = 0, 1, \ldots; \quad (u_i = x_{i+1} - x_i)
\]

\[
M_k^{(n)} = \frac{M_{k+1}^{(n+1)} - M_k^{(n+1)}}{e_k^{(n)}}, \quad N_k^{(n)} = \frac{N_{k+1}^{(n+1)} - N_k^{(n+1)}}{e_k^{(n)}},
\]

\[
(\text{for some finite } k)
\]

The convergence analysis of the sequence \( \{e_k(x_n)\}_{k=0}^{\infty} \) with fixed \( k \), for sequences \( \{x_m\} \) whose members behave as in Equations (62) and (63), with \( \lambda_i \) all positive or all negative, was given by Wynn (38). Wynn’s results were later extended by Sidi (39) to the case of general complex \( \lambda_i \), and also to sequences \( \{x_m\} \) whose members behave in a more general fashion than that given in Equations (62) and (63); see also chapter 16 in Ref. 27. One of the results of Ref. 39, which is a generalization of Wynn’s results, reads as follows:

**Theorem 11.1.** Provided \( |\lambda_k| > |\lambda_{k+1}| \) in Equations (62) and (63), there holds

\[
e_k(x_n) - s = O(\lambda_k^{n-k}) \quad \text{as } n \to \infty.
\]

For a detailed study of the Shanks transformation, see chapter 16 in Ref. 27.
Theorem 11.1 shows that the Shanks transformation is an effective convergence acceleration method when applied to scalar sequences \(\{x_m\}\) that behave as in Equations (62) and (63). Now, the vector sequences \(\{x_m\}\) in Theorems 7.1 and 7.2 behave exactly as in Equations (62) and (63) componentwise. This suggests that the Shanks transformation can be applied to these sequences componentwise to accelerate their convergence. When applied in the context of vector sequences in this way, the Shanks transformation is known as the scalar epsilon algorithm (SEA). Note that, SEA does not couple the different components of the vectors \(x_m\).

**Vector Epsilon Algorithm**

Another interesting vector method to accelerate the convergence of \(\{x_m\}\) that is based on the epsilon algorithm itself was proposed by Wynn (6). This method is known as the vector epsilon algorithm (VEA), and is defined as follows:

\[
\begin{align*}
\epsilon_{n+1}^{(n)} &= 0, & n &= 0, 1, \ldots, \\
\epsilon_{k+1}^{(n)} &= \epsilon_{k}^{(n)} + (\epsilon_{k+1}^{(n+1)} - \epsilon_{k}^{(n+1)}), & n, k &= 0, 1, \ldots, \\
\end{align*}
\]

Here, \(x^{-1}\) is the Samelson inverse of the vector \(x\), given by \(x^{-1} = \frac{x^\dagger x}{x^\dagger x} = \frac{x}{\|x\|^2}\), and \(x^\dagger\) is simply the complex conjugate of \(x\). With VEA, we take \(\epsilon_{2k}^{(n)}\) to be approximations to the limit or antilimit of \(\{x_m\}\). Clearly, unlike SEA, VEA couples all the components of the vectors \(x_m\).

The properties of VEA have been studied extensively via the so-called vector-valued Padé approximants: Graves-Morris (40, 41) has shown that, under the conditions of Theorem 3.6, the VEA approximations from the sequence \(\{x_m\}\) satisfy \(\epsilon_{2k}^{(n)} = s, m = n, n + 1, \ldots\). This result is known as McLeod’s theorem. It was conjectured by Wynn (42), and was first proved by McLeod (43) with the \(\epsilon_i\) restricted to be real numbers; see Ref. 1, page 211.

A determinantal formula for the \(\epsilon_{2k}^{(n)}\) has been given by Graves-Morris and Jenkins (44). This formula shows that \(\epsilon_{2k}^{(n)}\) is a “weighted average” of the \(x_m\) exactly as described in the Introduction even though this is not clear from Equation (69).

The convergence of the sequence \(\{\epsilon_{2k}^{(n)}\}_{n=0}^{\infty}\) has been analyzed by Sidi, Ford, and Smith (14) and Sidi and Bridger (22). It is proved in Ref. 14 that, under the conditions of Theorem 7.1, the rate of convergence of \(\epsilon_{2k}^{(n)}\) to \(s\) as \(n \to \infty\) is the same as those of \(s_{n,k}\) from MPE and RRE. Specifically, for all large \(n\), \(\epsilon_{2k}^{(n)}\) exist uniquely [part (1) of Theorem 7.1], and that the rest of the results of Theorem 7.1 [parts (2)–(4)] are valid for TEA without any changes, provided

\[
w \cdot y_i \neq 0, \quad i = 1, \ldots, k.
\]

**Remarks on Application of Epsilon Algorithms**

Clearly, all three epsilon algorithms can be applied in the cycling mode, just as MPE and RRE. It has been observed in various applications that, when used in the cycling mode, SEA is rather unstable, hence it is not recommended for vector sequences. For this reason, we comment only on VEA and TEA in the sequel.

From the various convergence analyses mentioned above, and from Theorem 3.6 and McLeod’s theorem, we conclude that the sequences \(\{\epsilon_{2k}^{(n)}\}_{n=0}^{\infty}\) from VEA and TEA, have convergence rates that are the same as those of \(s_{n,k}\) from MPE and RRE. It is clear from the recursion relations in Equations (69) and (70) that to determine \(\epsilon_{2k}^{(n)}\) we need the vectors \(x_0, x_1, \ldots, x_{2k+1}\), a total of \(2k + 1\) vectors. To compute \(x_{2k}\), we need \(x_0, x_1, \ldots, x_{2k+1}\), a total of \(k + 2\) vectors. Thus, VEA and TEA require practically twice as many vectors \(x_m\) to achieve the same kind of accuracy as MPE and RRE. This can already be of concern when the cost of computing each \(x_m\) is high. Also, we need to...
store $2k + 1$ vectors to obtain $e_k(n)$, whereas $k + 2$ vectors need to be stored to obtain $s_k(n)$. When $N$, the dimension of the vectors $x_m$, is very large, storage problems can occur with VEA more than with MPE and RRE. In addition, with the vectors $x_m$ already available, the cost of computing $e_k(n)$ is higher than the cost of computing $s_k(n)$ when the latter are computed via the algorithms of the sixth section; for the exact operation counts, see Sidi (20). For all these reasons, it seems reasonable to prefer MPE and RRE over epsilon algorithms in general.

BIBLIOGRAPHY

5. P. Wynn, On a device for computing the $e_m(S_n)$ transformation, Mathematical Tables and Other Aids to Computation, 10: 91–96, 1956.
AVRAM SIDI  
Technion—Israel Institute of Technology  
Haifa, Israel