Winter 1989

On Vector Sequence Transforms and Acceleration Techniques

Steven L. Hodge
Old Dominion University

ON VECTOR SEQUENCE TRANSFORMS AND ACCELERATION TECHNIQUES

by

Steven L. Hodge

B.S., August 1982, University of South Alabama, Mobile, AL

A Dissertation Submitted to the Faculty of Old Dominion University
in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
in
COMPUTATIONAL AND APPLIED MATHEMATICS

December, 1989

Approved by:

W.D. Lakin (Director)

ABSTRACT

On Vector Sequence Transforms and Acceleration Techniques

Steven Lee Hodge

Old Dominion University, 1989

Director: Dr. William D. Lakin

This dissertation is devoted to the acceleration of convergence of vector sequences. This means to produce a replacement sequence from the original sequence with higher rate of convergence.

It is assumed that the sequence is generated from a linear matrix iteration $x_{i+1} = Gx_i + k$ where G is an $n \times n$ square matrix and x_{i+1}, x_i, and k are $n \times 1$ vectors. Acceleration of convergence is obtained when we are able to resolve approximations to low dimension invariant subspaces of G which contain large components of the error. When this occurs, simple weighted averages of iterates x_{i+l}, $i = 1, 2, \dots k$ where $k < n$ are used to produce iterates which contain approximately no error in the selfsame low dimension invariant subspaces. We begin with simple techniques based upon the resolution of a simple dominant eigenvalue/eigenvector pair and extend the notion to higher dimensional invariant spaces. Discussion is given to using various subspace iteration methods and their convergence. These ideas are again generalized by solving the eigenelement problem for a *projection* of G onto an appropiate subspace. The use of Lanzcos-type methods are discussed for establishing these projections.

We produce acceleration techniques based on the process of generalized inversion. The relationship between the minimal polynomial extrapolation technique (MPE) for acceleration of convergence and conjugate gradient type methods is explored. Further acceleration techniques are formed from conjugate gradient type techniques and a generalized inverse Newton's method.

An exposition is given to accelerations based upon generalizations of rational interpolation and Padé approximation. Further acceleration techniques using Sherman-Woodbury-Morrison type formulas are formulated and suggested as a replacement for the E-transform.

We contrast the effect of several extrapolation techniques drawn from the dissertation on a nonsymmetric linear iteration. We pick the Minimal Polynomial Extrapolation (MPE) as a representative of techniques based on orthogonal residuals, the Vector ϵ-Algorithm (VEA) as a representative vector interpolation technique and a technique formulated in this dissertation based on solving a projected eigenproblem. The results show the projected eigenproblem technique to be superior for certain iterations.

To Dawne

Acknowledgements

I would like to thank Dr. William D. Lakin and the other members of the committee for their encouragement and friendly advice during the writing of this dissertation. I especially appreciate the mathematical insights provided to me by Dr. John Swetits and Dr. John Tweed. I would like to thank Dr. Stan Weinstein for his extensive proofreading of this dissertation at a time I had grown weary of it. I must also recognize Barbara Jeffrey for her help with this dissertation and, most of all, friendship.

I owe a debt of gratitude to the Department of Mathematics at O.D.U., the ICAM program at O.D.U., and the Theoretical Aerodynamics Branch of the Transonics Division at the N.A.S.A. Langley Research Center for the use of their equipment and monetary support.

My parents and brother never ceased to encourage me and I thank them. But most of all, I would like to thank my wife Dawne and son Daniel (5 years old) for their strength, patience, love and understanding during this seemingly infinite project which threatened to rend us apart.

In closing, I would also like to thank my daughter Lauren (at present 10 months old and squalling in her crib) who—although she came a little late to be of much help—may in the future like to see her name in print.

Contents

List of Figures

Chapter 1

Introduction

Mathematicians in the 18th century did not concern themselves with the convergence or divergence of series, which led to contradictions that were not resolved until 1821 when A.L. Cauchy gave firm definitions of the notion of convergence. With this definition, they concerned themselves mostly with convergent series; however, divergent series continued to arise in many problems in analysis. This fomented a systematic study of methods for the summation of divergent series, known as summability techniques or sequence transforms, by mathematicians in the latter 19th century.

Getting usable replacement sequences from divergent sequences is certainly valuable enough to have spawned a branch of mathematics whose general foundation is now the theory of linear transformations, but summability techniques have an additional practical use in the "acceleration" of the convergence of the partial sums of a *convergent* series. "Acceleration" means to produce a replacement sequence with a higher rate of convergence. If a sequence is thought of as the partial sums of some series, then a summability technique can be used for the acceleration of the convergence of a sequence.

Most summability techniques, including the so-called "nonlinear" summability techniques such as the classical Aitken's δ^2 technique, involve a weighted average of partial sums of members of the sequence. In this case, a distinction should be made between classical linear methods where (1) the weights are chosen in advance and whose ability to converge sequences is subject to the guidelines of Toeplitz-Schur type theorems; (2) methods such as Aitken's δ^2 which are nonlinear weighted averages in that the weights are nonlinear functions of the iterates (the process, nonetheless, amounts to a weighted average of the iterates); and (3) "truly nonlinear" methods which differ from the former two. Truly nonlinear methods are not dealt with in this dissertation. In keeping with modern notions of error analysis the first method (weights in advance) will be known as an a' priori method and the second method (weights determined from the particular sequence) will be known as an a'posteriori method.

Both a'priori and a'posteriori methods are attractive summability techniques for vector sequences. They can be applied "globally"—each component of the given vectors in a weighted sum use the same weight—as opposed to "locally" where each component of a vector is regarded as a different sequence and summed with different weights. There is no distinction between local and global techniques in a'priori techniques. This dissertation will be concerned with their use in the context of summing vector sequences. A'priori techniques have been used for years with good effect. A well known example is the Chebychev acceleration technique for linear matrix iterations (cf. chapter five). These techniques, in their purest form, do not take advantage of the developments in the sequence as the iteration progresses ("finely tuned" ones don't need to), but often require some advance information that may be expensive or difficult to obtain accurately. In the case of the Chebychev acceleration accurate estimates are needed on the spectrum of the iterative matrix in order for it to be effective. A'posteriori techniques, such as Shank's transform of chapter four, are adaptive to changes in the sequence and work better for certain sequences than the a'priori techniques, but they are generally more expensive to implement. They also require more memory, but have one great advantage in that a correctly applied a'posteriori technique is able to produce remarkable acceleration unobtainable by a'priori techniques. Another intermediate approach is to apply combinations of both techniques to the same sequence in some methodical fashion.

This dissertation contains a theoretical and numerical investigation of a'posteriori techniques for certain types of problems that commonly arise in numerical analysis. A contribution here will be a theoretical investigation oriented toward a linear algebraic aspect in the formulation of a'posteriori techniques.

Let $\mathcal{B}$ be a Banach space and s a sequence in $\mathcal{B}$. Many acceleration techniques have the characteristic of making an initial assumption about the error structure of the sequence and then algebraically combining sequence members in order to eliminate or minimize all or part of the error. For example, suppose that s_n is a Laplace moment sequence:

$$s_n = s^* + \int_0^\infty e^{-(\alpha+n)t} f(t)dt, \ \Re\alpha > 0. \tag{1.1}$$

In a particular case, Wimp [35] has shown that under the conditions that the set

$$S = \{m | f^{(m)} \neq 0, 0 \leq m \leq r-1\} \tag{1.2}$$

is not empty and $f \in L_2(0,\infty)$, $f^{(r)}(u)e^{-\alpha u} \in L_2(0,\infty)$, that there is a lower triangular array of weights $U = (\mu_{ij})$ such that the sequence

$$\hat{s}_n = \sum_{i=1}^{n} \mu_{ni} s_i \tag{1.3}$$

is accelerated at the rate

$$\hat{s}_n - s^* = n^{j-2r-1/2} O(s^n - s^*) \tag{1.4}$$

where $j = \inf S$. This remarkable acceleration, coupled with the added bonus that the weights satisfy a four term recurrence relation, make the summability method very useful for sequences with error (1.1). It has been shown that no other matrix of weights $\tilde{U}$ will improve convergence more than U [35].

A disadvantage of summability techniques starting from a specified error structure comes in the limitations of the error structure itself, not the technique. In certain situations the error structure is not known or too general to be of any practical use. (The rule of thumb is that *less* definite error structures produce *less* effective summability methods. See the paper by Brezinski [7] for information on the general theory of summation.) If no more information is availiable about the error structure at the onset, there is little that can be done except to wait for terms of the sequence to reveal more about the error than was known in the beginning. This is the approach that will be considered in this dissertation: Get an approximation to the error in the sequence and then use a summability technique based on eliminating the error approximation. The effectiveness of the extrapolation depends on the effectiveness of the error approimation.

The linear algebraic aspect will be in the consideration of dominant "eigenspaces". Assume that the sequence is generated by a linear operator $\mathcal{P}$ and resolve the error by, in the words of Peter Henrici, "quantifying the continuity [of $\mathcal{P}$] in the area of spectral analysis, that is (to use a less fashionable term), to consider an eigenvalue problem for $\mathcal{P}$." Among the tools used for resolving the error will be the numerical techniques developed during the 1960's and 1970's by the ablest numerical analyists of that time. Of primary interest will be sequences which have "dominant invariant subspaces". To explain what is meant by this, suppose that a sequence $\mathbf{s}_n$ in the complex N-space $\mathbf{C}^N$ is to be summed. This is a situation where the numerically largest component of the error lies in a proper subspace of dimension $K \ll N$. In this way, most of the error can be found by solving a more economical lower dimension problem. These tools of eigenvalue estimation will be augmented by the use of methods for projecting matrix problems, including eigenvalue problems, onto proper subspaces of $\mathbf{C}^N$.

This adaptive approach applied to summation of convergent vector sequences is of use for complicated sequence generators for which little analysis can be done to determine the error structure. These types of sequences appear often in the form of large codes in scientific computing for solving large systems of nonlinear partial differential equations. There is enough difficulty in formulating the codes themselves and getting them to converge than to worry about the awesome task of finding the error structure. Fortunately, the error sometimes has a dominant component which depends on a limited number of parameters that can be detected by a good extrapolation routine.

Chapter two involves the formulation of summability methods from the tools developed for the eigenelement problem. Methods that strictly employ eigenvalue approximation and methods that also incorporate projections will be developed. Tried and true eigenelement algorithms will be the building blocks for many of the sequence transforms. It can be argued that this highly developed resource has not been adequately exploited before. In justification of this point, a contrast between the methods of chapter three will be drawn, especially in terms of computational efficiency. In summary, a unifying framework for the construction of vector acceleration techniques will be discussed from the viewpoint of eigenelement/projection techniques.

Chapter three discusses generalizations of the Minimal Polynomial Extrapolation (MPE) of Cabay and Jackson [9] and produces some generalizations. It is shown that these techniques are essentially implementations of a projection technique of chapter two. These techniques turn out to be highly successful in the experiments of chapter five. There is also a discussion of the conjugate gradient technique and the planar conjugate gradient technique. It is shown these techniques can be used to produce extrapolations. In particular, the use of the planar conjugate technique has not been suggested before.

Chapter four will provide a review of certain modern sequence transforms with an emphasis on the Shank's transform and the Brezinski-Haive generalization of it, known as the B-H protocall or the E-transform [34]. In deference to its inventors (Brezinski and Håive), it will be referred to as the B-H protocall here. A discussion of the ϵ-algorithm and its vector version is presented. Motivated by the B.H. transform we suggest a related transform in section 4.3 based upon a sparse linear system solver.

Chapter five of this dissertation will involve a short exposition of the formulation of iterative methods from the viewpoint of solving an ordinary differential equation. More importantly, it provides a convenient way to generate realistic iterative methods that will test various sequence transforms produced in this dissertation. A main contribution here is a comparison of the MPE method and the vector ϵ-algorithm.

It is hoped that the examples from this chapter, along with the generality of the approach considered in this dissertation, will convince the working numerical analyst that vector sequence transforms are a simple, effective tool in optimizing the convergence rate of a wide variety of iterative methods.

1.1 Notation

1.1.1 Spaces

$\mathcal{H}$: Hilbert space

$\mathcal{B}$: Banach space

$\mathcal{B}'$: dual space

$\mathrm{B}(\mathcal{B}, \mathcal{B}')$: space of all bounded linear mappings of $\mathcal{B}$ into $\mathcal{B}'$.

$\|T\|$:

$$\|T\| = \sup_{|x| \le 1} \|T(\chi)\|, T \in B, \chi \in \mathcal{B}$$

A^T is the tranpose of A, A^H is the conjugate transpose $\overline{A^T}$. The spectral radius of a matrix A is denoted by $\rho(A)$.

1.1.2 Real and Complex Numbers

$\mathcal{C}^p$ space of ordered complex p-tuples, $p > 1$

$\mathcal{C}$ complex numbers

$\mathcal{R}^p$ space of ordered real p-tuples, $p > 1$

$\mathcal{R}$ real numbers

$\mathcal{R}^0$ nonnegative real numbers

$\mathcal{R}^+$ positive reals

$\mathcal{Z}$ integers

$\mathcal{Z}^0$ nonnegative integers

$\mathcal{Z}^+$ positive integers

m, n, k, r, i, j generally denote integers

1.1.3 Sequences

Sequences will be denoted by simple variables s, t etc. . Individual members will have indices s_i, t_k etc. . Limit points will usually be s_∞, $t_i nfty$.

$\mathcal{A}_s$: a space of sequences

Special Sequences

Δ: $\Delta s_n = s_{n+1} - s_n, \quad n \geq 1:$ If $\Delta x_0 = x_0$, we have that $\quad s_n = \sum_{k=0}^{n} \Delta x_k$

ϵ: $\epsilon_n = s_n - s, \quad s = \lim_{n\to\infty} s_n$

Chapter 2

Matrix Eigenvalue and Projection Techniques for Extrapolation

Let x be a sequence in a Banach space $\mathcal{B}$ which converges to x_∞. This chapter will be devoted to the formulation of vector sequence transforms of a segment

$$x_k, x_{k+1}, x_{k+2}, \cdots, x_{k+n} \tag{2.1}$$

of the sequence x, where $k, n > 0$ based upon the assumption that, over the segment $[k, k+n]$, the error $\epsilon_n = x_n - x_\infty$ is well approximated by a matrix iteration $\epsilon_{i+1} = G\epsilon_i$ where G is a square matrix with no generalized eigenvectors, i.e. G is nondefective. The set of nondefective matrices is dense in the set of all matrices. So in assuming that the sequence error is generated by a matrix, it is not unreasonably restrictive to make the matrix nondefective. Extrapolations will be considered from the viewpoint of removing components of error in dominant eigenspaces; that is, removing the component of error belonging to the eigenspace associated with a subset of eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_r$ of the spectrum of G, where $r \ll K = \text{order}(G)$ and

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_r|. \tag{2.2}$$

Assume now that we have *distinct* eigenvalues $\mu_1 > \mu_2 > \cdots > \mu_{\bar{K}}$ where $\bar{K} \leq K$, and define $M_i \equiv Ker(G - \mu_i I)$. Decompose the real K-space $\mathcal{R}^K$ into the direct sum $\bigoplus_{i=1}^{\bar{K}} M_i$ and let P_i be the projection onto the eigenspace M_i along $\bigoplus_{j\neq i}^{\bar{K}} M_j$. Then, with the reasonable assumption that the initial error ϵ_i has a component in each eigenspace, at the $i+l$ step the error has decomposition

$$\epsilon_{i+l} = \sum_{k=1}^{\bar{K}} \mu_k^l P_k \epsilon_i. \tag{2.3}$$

Hence even if the error ϵ_i has largest component in a subdominant eigenspace M_j where $j > 1$, it is expected that further errors ϵ_{i+l} will have their largest component of error in the dominant eigenspace M_1, their next largest in M_2, and so forth. Moreover, these dominant eigenspaces are the subspaces easiest to identify during the iterative process by the power method and other equivalent eigenelement or projection techniques to be discussed here.

Consider a matrix iteration $x_{i+1} = Gx_i + k$ which approximates a sequence x over the range $[k, k+n]$. Two approaches for extrapolation will be considered in this chapter:

1. Use approximations to eigenelements of G to form an extrapolation based upon removal of components of error in the approximant *dominant eigenspaces.* This will be called an *annihilating polynomial* approach.

2. Project the problem

   $$(I - G)\epsilon_i = r_i, \tag{2.4}$$

   where $r_i = (I - G)x_i + k$ or some other related problem, onto an approximant *dominant eigenspace*, "Solve" the projected problem for ϵ_i to produce approximant solution $\bar{\epsilon}_i$, and produce an approximation to x_∞ of the form $\bar{x}_\infty = x_i - \bar{\epsilon}_i \approx x_\infty$. This is the *error equation/residual* approach.

We start with the first approach in the simple case of a single dominant eigenvalue. Consider a matrix iteration

$$x_{i+1} = Gx_i + k \tag{2.5}$$

where G is an $n \times n$ matrix. If the dominant eigenvalue μ_1 has modulus strictly greater than the moduli of the subdominant eigenvalues, it can be well approximated as a byproduct of the iteration of several methods which fall under the class of simultaneous iteration techniques. One of the simplest such techniques will now be applied to (2.5). Other more elaborate simultaneous iteration techniques will be discussed later.

Assume for simplicity that $x_0 = 0$ and let $\Delta x_i = x_{i+1} - x_i$. Consider (2.5) in the form

$$\begin{aligned} \Delta x_{i+1} &= G\Delta x_i \\ x_{i+2} &= x_{i+1} + \Delta x_{i+1} \end{aligned} \tag{2.6}$$

Let (μ_1, u_1) be the dominant eigenvalue-normalized eigenvector pair and assume $x_i \cdot u_1 \neq 0$. It is well known that as $i \to \infty$, $u_i = \Delta x_i / \|\Delta x_i\|$ tends to the normalized dominant eigenvector u and that for $1 \leq k \leq n$ the ratios

$$\hat{\mu}_i = \frac{\|\Delta x_{i+1}\|_\infty}{\|\Delta x_i\|_\infty} \tag{2.7}$$

or Rayleigh quotients

$$\hat{\mu}_i = u_i^H G u_i \tag{2.8}$$

tend to μ_1 at a rate $\mathcal{O}\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right)$ where μ_2 is the next largest eigenvalue in modulus. It is not necessary for (2.5) to be convergent to get good approximations to the eigenvalues and eigenvectors. If the matrix G is Hermitian ($G = \overline{G^T}$), then the eigenvalue convergence becomes quadratic $\mathcal{O}\left(\frac{|\mu_2|}{|\mu_1|}\right)^{2i}$ when using the Rayleigh Quotient. For details see Golub and VanLoan [14].

We now consider the error in the extrapolation

$$x_e = \frac{x_{i+1} - \hat{\mu}_1 x_i}{1 - \hat{\mu}_1} \tag{2.9}$$

Assume that $\mu_1 \neq 1$ and let $x^* = (I - G)^{-1}k$. Then

$$\begin{aligned}
x^* - x_e &= x^* - \frac{x_{i+1} - \mu_1 x_i - O\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right) x_1}{1 - \mu_1 - O\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right)} \\
&= x^* - \left(\frac{x_{i+1} - \mu_1 x_i}{1 - \mu_1}\right) + O\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right) x_i \\
&= \frac{\epsilon_{i+1} - \mu_1 \epsilon_i}{1 - \mu_1} + O\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right) x_i \\
&= \frac{\bar{\epsilon}_{i+1} - \mu_1 \bar{\epsilon}_i}{1 - \mu_1} + O\left(\left(\frac{|\mu_2|}{|\mu_1|}\right)^i\right) x_i \qquad (2.10)
\end{aligned}$$

where $\bar{\epsilon}_i$ is the component of ϵ_i in $\{u_1\}^\perp$ of smallest ℓ_2 norm. Generally this quantity will be $O(|\mu_2|^i)$; however, this tends to be a large overestimate for large sparse matrix iterations where the eigenvalue μ_2 is close to μ_1. In this case it may happen that the quantity

$$\frac{\bar{\epsilon}_{i+1} - \mu_1 \bar{\epsilon}_i}{1 - \mu_1}$$

(2.10) may be equal to

$$\frac{\bar{\epsilon}_{i+1} - \mu_1 \bar{\epsilon}_i}{1 - \mu_1} + O\left(\left(\frac{|\mu_3|}{|\mu_2|}\right)^i\right)$$

where $\bar{\epsilon}_k$ for $k = i, i+1$ is in $\{u_1, u_2\}^\perp$ where u_2, μ_2 is the second eigenpair and μ_3 is the third eigenvalue. Of course this argument may be extended for any number of clustered eigenvalues, essentially we expect the error to be $O(|\mu_k|^i)$ where μ_k is the first *numerically* significant eigenvalue away from μ_1.

The *stability* of the extrapolation is closely related to the stability of the eigenvalue problem for μ_1. Consider the eigenvalue μ_1 to be simple and pick a left eigenvector ψ_1 of G normalized so that $\|u_1\| = \psi_1^H u_1 = 1$. $\|\psi_1\|$ may be large if G is nonhermitian. Assume a perturbation H of G such that $G' = G + H$ and $\|H\|_2 < \epsilon$ where ϵ is small and set $\epsilon' = \|H\|_2$ so that $\epsilon' \leq \epsilon$. The following theorem is well known (see, for instance, [10]).

Theorem 1 *If ϵ is small enough, there exists a simple eigenvalue μ_1' of G' with an eigenvector u_1' normalized by $\psi_1^H u_1' = 1$, such that*

$$\mu_1' = \mu_1 + \psi_1^H H u_1 + O(\epsilon^2) \tag{2.11}$$

$$u_1' = u_1 - SHu_1 + O(\epsilon^2). \tag{2.12}$$

where $S = \left((G - \mu_1)|_{\{\psi_1\}^\perp}\right)^{-1}(I - P)$ and $P = u_1\psi_1^H$.

From equations (2.11) and (2.12) it is seen that the stability of the eigenvalue μ_1 is dependent on the dominant left hand eigenvector ψ_1 and the stability of the eigenvector is dependent on the generalized inverse of $A - \mu_1 I$ relative to the spectral projection $P = u_1\psi^H$, namely the operator S. For information on generalized inverses see [5] or the introduction in chapter 3 of this dissertation.

Theorem 2 *For the case of a single dominant eigenpair (μ_1, u_1) the extrapolation (2.9) is stable if the eigenvalue problem for G is stable.*

Proof. Assume for simplicity that inner products have been accumulated with enough accuracy so that the round-off error in the iterates $x_0, x_1...$ are negligible. For small enough $\epsilon = \|H\|$, there is a separated simple dominant eigenpair (μ_1', u_1'). Let $\hat{\mu}_1'$ be the power iteration approximation to the dominant eigenvalue of G' after the ith iteration:

$$\hat{\mu}_1' = \mu_1' + O\left(\left(\left|\frac{\mu_2'}{\mu_1'}\right|\right)^i\right)$$

where μ_2' is the next closest eigenvalue in modulus. Check the first coefficient in the extrapolation (2.9):

$$\begin{aligned}\frac{1}{1-\hat{\mu}_i} - \frac{1}{1-\hat{\mu}_1} &= \frac{\hat{\mu}_1 - \hat{\mu}_1'}{(1-\hat{\mu}_1)(1-\hat{\mu}_1')} \\ &= \frac{\psi^H H u_1 + O(\max|\frac{\mu_2}{\mu_1}|^i, |\frac{\mu_2'}{\mu_1'}|^i)}{1 + \psi^H H u_1 + \mu_1\mu_2 + O(\max\{|\frac{\mu_2}{\mu_1}|^i, |\frac{\mu_2'}{\mu_1'}|^i\})}.\end{aligned}$$

The stability is now evident for large enough i. Sufficiency for the other coefficient

$$\frac{\hat{\mu}_1}{1-\hat{\mu}_1}$$

follows in the same fashion. ▌

This method of accelerating convergence by explicitly approximating the dominant eigenvalue seems to be due to Lusternik [13], and has been rediscovered many times.

2.1 Extending Lusternik's Method to a Subspace

To describe the extension of Lusternik's method to subspaces, the notation of subspace convergence needs to be defined. Given a subspace $Y \subset C^N$, where $\dim Y = r$, $\mathbf{Y}$ is an $N \times r$ matrix whose columns are a basis of Y. The basis of Y is completed by the $N - r$ columns of $\mathbf{Z}$ into a basis of C^N. Given a subspace X_k of C^N, dim $X_k \leq r$, $\mathbf{X}_k$ is an $N \times r$ matrix whose columns span X_k.

Definition 1 (Parlett and Poole [26]) *Given a sequence X_k of subspaces of C^n of a vector space where $\dim X_k \geq r$ for all k, X_k converges to Y, denoted $X_k \rightarrow Y$, as $k \rightarrow \infty$ if and only if there exists, for large enough n, matrices of column vectors $\mathbf{X}_k$,$\mathbf{Y}$, $\mathbf{Z}$ and matrices A_k (square, invertable $r \times r$) and B_k $((r_k - r) \times r)$ such that*

$$\mathbf{X}_k = \mathbf{Y} A_k + \mathbf{Z} B_k.$$

where $\|A_k^{-1} B_k\| \to 0$ *as* $k \to \infty$.

Let the eigenvalues of G be ordered by decreasing modulus $|\mu_1| \geq |\mu_2| \geq |\mu_3| \geq .. \geq |\mu_r| > |\mu_{r+1}| \geq .. \geq |\mu_s|$ where there is a strict separation in the rth spot. Suppose that G has a spectrum with a gap in modulus between the rth and $(r+1)$st eigenvalues, $|\mu_1| \geq |\mu_2| \geq ... \geq |\mu_r| > |\mu_{r+1}| \geq ...|\mu_k|$ and that M = span $\{\phi_1, \phi_2, \phi_3, ..., \phi_r\}$ where ϕ_i is the normalized eigenvalue associated with the eigenvalue μ_i. The following theorem due to Parlett and Poole [26] describes when subspace convergence will occur.

Theorem 3 *Suppose that there is a separation in the spectrum of G. Let U be an arbitrary matrix of vectors* $[u_1, u_2, ...u_r]$ *and P be the projection onto the dominant space M parallel to the space of subdominant eigenvectors* $\{\phi_{r+1}, \phi_{r+2} ..., \phi_s\}$. *Then* $G^k U = span(G^k \phi_i)_{1=1,r}$ *converges to M if and only the set* $\{Pu_i\}_{i=1,r}$ *is linearly independent.*

Note that theorem 3 is a statement about a dominant eigenspace, that is, the subspace which dominates the iteration. When a good approximation to the error on the dominant eigenspace is found, an extrapolation may be formed by removing this component of the error from the approximate solution (see section 2.2). Suppose that the column vectors of $G^k\mathbf{U}$ are linearly independent, but some of them are close enough to each other such that for some $i \geq k$, $G^k\mathbf{U}$ has columns that are linearly dependent when represented in the floating point precision of a computer. This phenomena of ill-conditioning leads to no convergence or convergence to a lower order dominant eigenspace—the former being disasterous, the latter wasteful. Further iterations may also produce further dependencies. So it is wise from a numerical viewpoint to produce column vectors of moderate size that are as orthogonal as possible and span the the subspace $G^k\mathbf{U}$ before performing further iterations.

This can be accomplished by producing an orthonormal basis $\mathbf{B}$ from the columns of $G^k\mathbf{U}$ and considering a new subspace iteration

$$G^k\mathbf{B} \qquad k = 1, 2, \ldots . \tag{2.13}$$

If the orthonormal basis is recomputed after each iteration, the classical orthogonal iteration method is produced. The orthogonalization is usually accomplished by means of Householder transformations or the modified Gram Schmidt orthogonalization [14]. The former method is unconditionally stable; the latter is simpler, cheaper and stable enough when the number of iterations is restricted to be under $t/\log_{10}(\|G\|\,\|G^{-1}\|)$ where t is the maximum number of significant figures allowed to be lost [31].

The actual algorithm is a straight forward generalization of the power method. Let p be a chosen integer satisfying $1 \leq p < n$. Given an $n \times p$ starting matrix Q_0 with orthonormal columns, a sequence of matrices $\{Q_k\} \in C^{n\times p}$ is generated as follows:

$$\begin{aligned} \text{For } k &= 1, 2, \cdots \\ Z^k &= A^{l_k} Q_{k-1} \\ Q_k R_k &= Z_k \qquad \text{(QR factorizaton)} \end{aligned} \tag{2.14}$$

where l_k is some integer picked at the kth step. Note that the QR factorization is applied to Z_k, an $n \times p$ matrix, which is not expensive when p is small enough.

Recall that the subspace $G^k\mathbf{U}$ converges if and only if the starting vectors $u_1, u_2, \ldots u_k$ are such that $Pu_1, Pu_2, \ldots Pu_k$ are linearly independent. When this occurs, some of the column vectors may converge to eigenvectors. The convergence is guaranteed in the ith column if $|\mu_{i+1}| > |\mu_i|$ upon which the "rate of convergence" will be $O\left(\frac{|\mu_{i+1}|}{|\mu_i|}\right)$. "Rate of convergence" is defined precisely in Definition 2 p.22 following. The constant in $O\left(\frac{|\mu_{i+1}|}{|\mu_i|}\right)$ depends greatly on the degree of normality of G and $|\mu_{k+1} - \mu_k|$. The convergence can be quite slow if the gap between μ_{k+1} and μ_k is not sufficiently wide. We illustrate this more precisely in the following with results from Golub and Van Loan [14].

First we need a framework which describes the sensitivity of invariant subspaces of a matrix. It should be noted that it is possible for sensitive eigenvectors (eigenvectors unstable under mild perturbations of the matrix), to span a subspace *insensitive* to mild perturbations! (see Golub [14] p.199 ff.) To analyze the behavior of orthogonal iteration, we use the classical *Schur* decomposition [10] which is proved by induction:

Theorem 4 (Schur Decomposition) *If $A \in C^{n\times n}$ then there exists a unitary $Q \in C^{n\times n}$ such*

$$Q^H AQ = T = D + N$$

where $D = diag(\lambda_1, \lambda_2, \dots \lambda_n)$ and $N \in C^{n\times n}$ is strictly upper triangular. Furthermore, Q can be chosen so that the eigenvalues λ_i appear in any order along the diagonal.

Suppose that

$$Q^H AQ = T = \text{ diag}(\lambda_i) + N, \quad |\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n| \tag{2.15}$$

is a Schur decomposition of $A \in C^{n\times n}$ and partition Q, T, and N as follows:

$$Q = \begin{array}{cc} [Q_\alpha, & Q_\beta] \\ p & n-p \end{array} \quad T = \begin{array}{cc} \left[\begin{array}{cc} T_{11} & T_{12} \\ 0 & T_{22} \end{array}\right] & \begin{array}{c} p \\ n-p \end{array} \\ \begin{array}{cc} p & n-p \end{array} & \end{array} \tag{2.16}$$

$$N = \begin{array}{cc} \left[\begin{array}{cc} N_{11} & N_{12} \\ 0 & N_{22} \end{array}\right] & \begin{array}{c} p \\ n-p \end{array} \\ \begin{array}{cc} p & n-p \end{array} & \end{array}$$

If $|\lambda_p| > |\lambda_{p+1}|$, then the subspace $D_p(A)$ defined by

$$D_p(A) = R(Q_\alpha)$$

is a *dominant* invariant subspace. It is the unique invariant subspace associated with the eigenvalues $\lambda_1, ..., \lambda_p$. The following theorem shows that with reasonable assumptions, the subspaces $R(Q_k)$ generated by (2.14) converge to $D_p(A)$ at a rate proportional to $|\lambda_{p+1}/\lambda_p|^k$. First, a definition is needed:

Definition 2 *Let* $\|\cdot\|_F$ *denote the* Frobenius *norm defined by*

$$\|A\|_F = \left[\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right]^{1/2}$$

where $A \in C^{m\times n}$. *We define the* separation *between two square matrices* A *and* B *by*

$$sep(A,B) = \min_{X\neq 0}\frac{\|AX - XB\|_F}{\|X\|_F}$$

for all compatible X *and define for subspaces* S_1 *and* S_2 *the distance to be*

$$dist(P,Q) = \|\pi_1 - \pi_2\|_2 \tag{2.17}$$

where π_i *is the orthogonal projection on* S_i.

sep(A,B) describes the relative distance between A and B modulo similarity transformations which, in effect, describes a distance between $\rho(A)$ and $\rho(B)$ [14]. It estimates the smallest singular value of the transformation

$$X \rightarrow AX - XB.$$

Theorem 5 *Let the Schur decompositon of* $A \in \mathbf{C}^{n\times x}$ *be given by (2.16) and (2.17). Assume that* $|\lambda_p| > |\lambda_{p+1}|$ *and that* $\theta \geq 0$ *satisfies*

$$(1+\theta)|\lambda_p| > \|N\|_F.$$

If $Q_0 \in \mathbf{C}^{n \times p}$ has orthonormal columns and

$$d = \text{dist}[D_p(A^H), R(Q_0)] < 1,$$

then the matrices Q_k generated by 1.15 satisfy

$$\text{dist}[D_p(A), R(Q_k)] \leq$$

$$\frac{(1+\theta)^{n-2}}{\sqrt{1-d^2}} \left[1 + \frac{\|T_{12}\|_f}{\text{sep}(T_{11}, T_{22})}\right] \left[\frac{|\lambda_{p+1}| + \|N\|_f/(1+\theta)}{|\lambda_p| - \|N\|_f/(1+\theta)}\right]^k .$$

When θ is large enough, the theorem essentially shows that

$$\text{dist}[D_p(A), R(Q_k)] \leq c|\lambda_{p+1}/\lambda_p|^k,$$

where d depends on $\text{sep}(T_{11}, T_{22})$ and $A's$ departure from normality. The convergence can be very slow if the gap between $|\lambda_p|$ and $|\lambda_{p+1}|$ is not sufficiently wide. However, it can be seen from the above that with a poorly conditioned set of eigenvectors with only moderate eigenvalue separation an invariant subspace may still show up.

2.1.1 Remarks on Methods Based upon Resolving Eigenspaces

As has been shown by Parlett and Poole [26], the subspace iterations accomplished by most of the methods tried in chapter 5, including simultaneous iterations and the QR iteration, are the same in one important aspect:

- *The sequences of subspaces generated by these methods are the same, differing only in the basis in which they are represented.*

Of course the basis representation is very important in the numerical computations with the goal being to avoid the tendency toward ill-conditioning that a primitive power iteration produces by forcing every starting vector not within a subdominant eigenspace toward the most dominant eigenvectors. Moreover, the QR approach to simultaneous iteration converges quadratically for smaller eigenvectors when combined with shifts.[25]

As with the transformation (2.9), the effectiveness of the more general extrapolations will be dependent on both *(1)* the convergence of approximating subspaces to the dominant subspace and *(2)* the stability of the projected eigenvalue or error/residual equation.

The extrapolations used so far will not be effective for matrix iterations with a large number of equimodular (but not equal) eigenvalues unless an inordinately large number of simultaneous iterates are taken. When all are equimodular, subspace convergence, if it takes place at all, is usually too slow to allow for an effective extrapolation [26]. When there are a large number of equimodular eigenvalues, we will reduce expenses by relying on a projection technique that depends upon subspace convergence. It should be noted in closing that this approach is necessary because solving the complete eigenelement problem is often a hazardous undertaking. For example, when a matrix is *derogatory*, i.e. has eigenvalues without unit geometric multiplicity, companion matrix decomposition is inadvisable. Similarly, Jordan decomposition is a formidable step beyond the real Schur decomposition that makes it impractical in numerical analysis. For details see Golub and Van Loan [14].

2.2 Producing an Extrapolation Method from the Simultaneous Iteration Technique

Consider a matrix iteration $x_{i+1} = Gx_i + k$ where G is $n \times n$ and n is large. Consider $U = \text{span}\{\Delta x_0, \Delta x_1, \Delta x_2, ...\Delta x_r\}$ and the subspace iteration
$G^n U$, $n = 1, 2, 3, ...$ Recall Lusternik's method: Let P_S denote a projection into a subspace S. If there is a single dominant eigenvalue λ with associated eigen-

space M, then $P_M\epsilon_{i+1} = \lambda P_M\epsilon_i$ and hence $P_Mx^k = \dfrac{P_Mx_{i+1} - \lambda P_Mx}{1-\lambda}$. In other words, the extrapolation is (in theory) exact on the dominant eigenspace. In reality, the quantity $\dfrac{x_{i+1} - \lambda x_i}{1-\lambda} = y_0$ is computed and there becomes a concern as to the size of the quantity $Q_N\epsilon_{i+1} - \lambda Q_N\epsilon_i$ where Q_N is the projection onto the subdominant eigenspace $(I - P_M)C^n = N$ parallel to M. There is the possibility of amplifying the component of error in subdominant eigenspace. This amplification may be accceptable, however, if the iterative matrix has the effect of diminishing error in the subdominant eigenspace relatively fast compared to the dominant eigenspace.

Consider now the iterative scheme $y_{i+1} = Gy_i + k$ and associated error $\epsilon_{i+1} = G^i\epsilon_i$, $\epsilon_1 = y_1 - x_\infty$. In theory $\|G\epsilon_1\| = \|Q_N(G\epsilon_i)\|$; in practice, however, there will always be round-off error that will cause the reappearance of a component of error in the dominant eigenspace. Assume for now that the round-off error in the dominant eigenspace remains negligible (within a reasonable number of iterations). Then the error that is of most concern in further iterations lies in the subdominant eigenspace. Assume that there are separations between the largest subdominant eigenvalue and further elements of the spectrum

$$|\mu_1| > |\mu_2| > |\mu_3| \geq |\mu_4| \geq \dots \geq |\mu_n|.$$

The error iteration $\bar{\epsilon}_{i+1} = G\bar{\epsilon}_0$ has the form $\mathbf{c}\mu_1^i + \mathbf{d}\mu_2^i + O(\mu_3^i)\mathbf{v}$ where $\mathbf{c}, \mathbf{d}$ and $\mathbf{v}$ are vectors. It is assumed that $\mathbf{c}$ was $\mathbf{0}$ to machine accuracy, hence the term $\mathbf{d}\mu_2^i$ will dominate the iteration if there is large enough separation between $|\mu_2|$ and $|\mu_3|$. It is easily seen by the argument similar to (2.10) that the eigenvalue estimates (2.7) and (2.8) will approximate μ_2 to $O\left(\left(\frac{|\mu_3|}{|\mu_2|}\right)^i\right)$ in the nonhermitian case and $O\left(\left(\frac{|\mu_3|}{|\mu_2|}\right)\right)^{2i}$ in the hermitian case when the Rayleigh quotient is used.) Consequently, we expect the extrapolation

$$w_i = \frac{y_{i+1} - \hat{\mu}_i y_i}{1 - \hat{\mu}_i}$$

similar to (2.9), where $\hat{\mu}_2$ is the numerical approximation to the second eigenvalue, to have convergence rate close to $O(\mu_3)$ if the eigenproblem is stable. This corresponds to the process of "deflation" in the classical power method for eigenvalues and eigenvectors of a square matrix. (see Wilkinson [33]).

If the matrix G is hermitian and the spectrum of G has the form $|\mu_1| > |\mu_2| > ... > |\mu_N|$, the above method extends easily to simultaneous iteration because of the convergence of the orthogonal set of spanning vectors to the actual eigenvectors of G and, consequently, the convergence of the eigenvalues which can be calculated by various ratios, including those described earlier. (Rayleigh quotient and the ℓ_∞ method). To implement the extrapolation, we notice that the iterated polynomial

$$(G - \mu_n)(G - \mu_{n-1})...(G - \mu_1)\epsilon_0 \tag{2.18}$$

has the effect of taking out successive error components in the spans of the eigenspaces associated with the eigenvalues $\mu_1, \mu_2, \mu_3, ..., \mu_n$. There is in theory no specific order in which the factors may be applied since G obviously commutes with itself; however, in practice it seems more stable to use the order given above. The extrapolation then has the form

$$x_\infty = \frac{p(G)x_0}{p(1)} \tag{2.19}$$

where $p(\mu) = \prod_{i=0}^{n} (\mu - \mu_i)$ and use has been made of the fact that $\epsilon_{i+1} = G\epsilon_i$. Again it must be noted that the extrapolation is not exact on subdominant subspaces and, as can be seen from (2.6) and (2.9) an amplification of the error can happen there unless the eigenvalues are tightly clustered. This amplification, when controlled, is often only temporary and may even be beneficial in producing eigenvalue estimates for further extrapolations. Figure 2.1 on the next page is a suggested algorithm for eigenvalue extrapolation.

Figure 2.1: Algorithm for Selective Eigenvalue Extrapolation

1. Generate a number of terms of the base sequence

$$x_n, x_{n+1}, x_{n+2} \ldots x_{n+k}$$

2. Perform subspace iteration with number of iterates.
3. If there is no convergence, exit or go to larger basis and return to step 1.
4. If there is convergence, detemine eigenvalues and extrapolate using

$$y_i = \frac{p(G)}{p(I)} x_{n+i} \quad i = 0, 1, 2, \ldots.$$

It may be advantageous to apply a further extrapolation to the sequence. In fact, to minimize computations the same polynomial may be used, especially if it initially gave a good drop in the norms of the residuals $\|(I-G)x_i - k\| = \|\Delta x_i\|$ and it has certain properties which will now be described . Consider $p(G)\epsilon_n$. $p(G)$ has the same invariant subspaces as G with corresponding eigenvalues, $p(\mu_i)$, $i = 1, 2, ...n \leq N$. If the error $p(G)\epsilon_n$ still has its dominant component in the dominant invariant eigenspace associated with the base sequence, then it is appropriate to consider $p(G)p(G)\epsilon_n$. The transformation now has the form

$$w_i = p(G)^2 x_{n+i} \quad i = 0, 1, 2, ... \tag{2.20}$$

Schematically, we have

$$\begin{tikzcd}
\vdots & & \\
x_n & & \\
x_{n+1} & & \\
x_{n+2} & & \\
\vdots & & \\
x_{n+k} \arrow[r] & y_0 & \\
x_{n+k+1} \arrow[r] & y_1 & \\
x_{n+k+2} \arrow[r] & y_2 & \\
x_{n+k+3} \arrow[r] & y_3 & \\
\vdots & \vdots & \\
x_{n+2k} \arrow[r] & y_k \arrow[r] & w_0 \\
x_{n+2k+1} \arrow[r] & y_{k+1} \arrow[r] & w_1 \\
x_{n+2k+2} \arrow[r] & y_{k+2} \arrow[r] & w_2 \\
\vdots & \vdots & \vdots
\end{tikzcd} \tag{2.21}$$

Note that each vector in the second column is composed of a "window" average of $k+1$ (fixed) vectors in the previous column. This corresponds to a summability method with a matrix of fixed bandwidth. The polynomial p can be used as long as it is effective and enough storage exists to hold $r(k+1)$ vectors where r is the number of columns in the schematic. If there is a k such that $t_0 = p(G)^k \epsilon_n$ has a dominant error component in a subdominant eigenspace, it is time to compute a new polynomial $r(G)$ and consider $(r(G))^\ell t_0$, $\ell = 1, 2, \ldots.$ The polynomial will likely be of different degree since we are working with a different subspace. In practice, there is no way to check the actual error; the residuals or some other approximate error measure is used instead. A general form of these eigenvalue type extrapolations has the form

$$\bar{x}_i = \prod_i (p_i(G))^{k_i} x_{n+i}. \tag{2.22}$$

2.3 Extrapolation with Projection Methods

2.3.1 Introduction

The invariance of dominant eigenspaces leads to the idea of using projection techniques. The basic idea behind projection methods is to approximate the solution to a "large" dimension problem by solving a lower dimensional problem. Consider the example of the representation of a transformation G with respect to a decompositon $M_1 \oplus M_2 \in C^N$ where $M_1 = M$ is the approximate dominant eigenspace and M_2 is some orthogonal complement of M chosen, if it can be done to make the projection problem well conditioned. Let $m_1 = \dim M$ and $m_2 = \dim M_2$. Then $m_1 + m_2 = N$ and the transformation G may be written with respect to the ordered decomposition $M_1 \oplus M_2$ as

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}. \tag{2.23}$$

Here G_{ij} $(i, j = 1, 2)$ is an $m_i \times m_j$ matrix that represents the transformation $P_i G|_{M_j} : M_j \to M_i$ where P_i is the projector on M_i along M_{3-i}. Note that

$P_1 + P_2 = I$. In particular, a projection on M_i along M_2 has the form

$$\begin{bmatrix} I & X \\ 0 & 0 \end{bmatrix} \tag{2.24}$$

and a projection of M_2 along M_1 has the form

$$\begin{bmatrix} 0 & Y \\ 0 & I \end{bmatrix} \tag{2.25}$$

If the space $M_1 = M$ is invariant, the matrix (2.23) has the form

$$\begin{bmatrix} G_{11} & G_{12} \\ 0 & G_{21} \end{bmatrix} \tag{2.26}$$

If the space M_2 is invariant, which will be the case if M_2 is the orthogonal complement of M, the matrix has the form

$$\begin{bmatrix} G_{11} & 0 \\ G_{12} & G_{22} \end{bmatrix}. \tag{2.27}$$

and, finally, if both M_1 and M_2 are invariant, the matrix has the form

$$\begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} \tag{2.28}$$

and we see that $(I - G)\epsilon_i = r_i$ has been decoupled into two problems. This complete decoupling can be accomplished with both spectral projections (projections onto eigenspaces along complementary eigenspaces) and with orthogonal projections when the subspace being projected upon is G-invariant.

In simultaneous iteration it is unlikely, especially with reorthogonalization for a nonsymmetric matrix iteration, that the spanning vectors will converge to eigenvectors. Consequently, it is difficult to find accurate eigenvalues to use in the extrapolation (2.20). However, a single eigenvalue extrapolation is often cheap enough that it is worth a try, especially in a large sparse linear iteration with a tightly clustered spectrum.

Another recourse is to try solving a projected eigenvalue problem. Here the eigenproblem is projected onto a subspace and then the projected eigenproblem is solved completely. Consider a simultaneous iteration G^kU where U is a span of column vectors. Usually U will have dimension n much smaller than $N =$ order (G). Often it consists of spans of successive matrix iterates or some algebraic combination of them, making it a Krylov type method. Consider a matrix π_n of orthogonal projection of C^n onto U so that we have a low dimension problem

$$\begin{aligned} &\text{find values } \ \mu_n \ \text{ and } \ \theta_n \neq 0 \in G^kU \\ &\text{such that} \\ &\pi_n(G\theta_n - \mu_n\theta_n) = 0. \end{aligned} \tag{2.29}$$

In terms of actual matrices in the hermitian case, this is accomplished by picking an orthonormal basis B of G^kU forming a matrix P_n, whose columns are the elements of the basis, and then solving

$$(P_k^H G P_k - \mu P_k^H P_k)\xi_n = 0. \tag{2.30}$$

The matrix $P_k^H G P_i$ is of "small" dimension $k \times k$ which makes the solution of (2.25) computationally practical. A third recourse to be explored is the projection of the residual equation

$$\pi_n((I - G)\epsilon_i - r_i)$$

upon some subspace.

2.3.2 Avoiding Amplification by the Incorporation of Eigenvectors

First, projections will be accomplished with a direct incorporation of approximate eigenvectors. Consider the iteration $x_{i+1} = Gx_i + k$ where G is square. Suppose that G is nondefective and that an approximate eigenvalue μ and eigenvector θ have been found. The calculation of the approximate eigenvectors is often a necessary part of the method, such as in the simultaneous iteration methods and their projected versions. They may also be obtained at a reasonable cost by extensions of various methods such as a projected version of the well known QR algorithm or Stewart's algorithm [31] where the eigenvectors are computed by the method of shifted iteration or shifted inverse iteration [14].

Let $M=$ span $\{\theta\}$. Assuming the invariance of M, an approximation to $(I-G)^{-1}|_M$ is given by the $\frac{1}{1-\mu}$, which can be regarded as a single element matrix with respect to the basis $\{\theta\}$. Let $r_i = (I-G)x_i - k$ and $\epsilon_i = x_i - x_\infty$ where $x_\infty = Gx_\infty + k$. Then $\epsilon_i = (I-G)^{-1}r_i$. When the matrix is Hermitian, in which case the subdominant eigenspace is $M^\perp$, the residual is projected onto M by using the orthogonal matrix projection associated with the normalized θ, namely the matrix $\theta\theta^H$. Write $r_i = \theta\theta^H r_i + m$ where $m \in M^\perp$. Then $\epsilon_i = (r_i\theta^H)\theta/(1-\mu) + (I-G)^{-1}m$ or

$$x_\infty = x_i + \frac{\theta\theta^H r_i}{1-\mu} + (I-G)^{-1}m. \tag{2.31}$$

The iteration $(I-G)^{-1} = \left(\sum_{1=0}^{\infty} G^i\right) m$ will not necessarily have a faster asymptotic rate of convergence than the iteration $x_{i+1} = Gx_i + k$, but if G is close to a Hermitian matrix, it can be expected to have a reasonable improvement in convergence rate, at least for a while. Without considering the accuracy of the eigenvalue, the expression (2.31) is accurate to $O(\|(I-\theta\theta^H)\chi\|)$ where χ is the actual dominant eigenvector of G (Chatelin [10] p.53). The expression $(I-G)^{-1}m$ must be evaluated. One of the many possible approximations is the partial sums of a series solution to $(I-G)^{-1}m$ with, perhaps, further extrapolations performed on it.

Now a generalization of (2.31) will be presented for hermitian matrices. Suppose that a k-vector simultaneous iteration has been performed to produce approximate eigenvalues $\mu_1^{(n)}, \mu_2^{(n)}, ..., \mu_n^{(n)}$, to exact eigenvalues $\mu_1, \mu_2, \mu_3, ...\mu_k$, and appropriate approximate normalized eigenvectors

$$\theta_1^{(n)}, \theta_2^{(n)}, \theta_3^{(n)}, ...\theta_k^{(n)} \tag{2.32}$$

to exact normalized eigenvectors $\theta_1, \theta_2, \theta_3, ...\theta_k$. It is known that the hermitian property of the matrix and a separation in the spectrum $|\mu_k| > |\mu_{n+1}|$ are sufficient conditions to get this convergence. Let M=span $\{\theta_1^{(n)}, \theta_2^{(n)}...\}$. Decompose the residual into approximant dominant and subdominant eigenspace components:

$$r_n = \sum_{i=1}^{k} \theta_i \theta_i^H r_n + m \qquad m \epsilon M^{\perp}. \tag{2.33}$$

then, similar to (1.26)

$$x_\infty = x_n + \sum_{i=1}^{k} \frac{1}{1-\mu_i} \theta_i^H r_n + (I-G)^{-1} m \tag{2.34}$$

As before the actual extrapolation results from making an approximation to $(I-G)^{-1}m$. Let $A_n(G)$ be an approximation to $(I-G)^{-1}$ which depends on the iteration. Then a sequence transform is

$$y_n = x_n + \sum_{i=1} \frac{1}{1-\mu_i} \theta_i^H r_n + A_n(G) m \tag{2.35}$$

A similar transform may be applied to the sequence y_n in (2.35) to produce a multilevel transform similar to (2.21).

When the matrix G is nonhermitian and is not close to a hermitian matrix, a different approach is needed.

Let ν be the number of vectors in the orthogonal iteration and consider the projected problem

$$(Q_\nu^H G Q_\nu - \mu Q_\nu^H Q_\nu)\eta = 0 \tag{2.36}$$

formed from a similarity transformation by the $n \times \nu$ matrix Q_ν whose columns are the orthonoralized vectors for the iteration. This problem is to be solved completely. In general, eigenvalue methods for determining the eigenvalues of a general nonsymmetric matrix cost $O(m^3)$ floating point operations (flops) where m is the order of the matrix. To solve for additional selected eigenvectors (generally by the method of inverse iteration) requires $O(m^2)$ flops per eigenvector. If the Jordan structure of the matrix is not easily discernable, which sometimes happens with matrices of large defect, it may become impractical to determine the eigenvectors and generalized eigenvectors. In this extreme, eigenvalue extrapolations become impractical; however, a dominant eigenspace may be revealed upon which to project the error/residual equation. Some details, mostly theoretical, for establishing a projection will be summarized here. We also summarize some well known methods in which the projections of the matrices are represented in convenient matrix forms (such as the Lanzcos tridiagonalization). We give another related formulation, in the form of the "MPE" method, of effectively establishing a projection onto an approximant dominant eigenspace in the next chapter. Here, as before, we force the residual of the extrapolation to be orthogonal to a subspace, but in this situation a linear variety (a translate of a subspace) is considered directly and the actual subspace is determined indirectly. This formulation turns out to be the most stable of the projection accelerations on the moderate size vector sequences of chapter 5.

Suppose that a dominant eigenspace M has emerged during the course of an iteration $x_{i+1} = Gx_i + k$. It is desirable to construct a well conditioned basis B for M. To accomplish this, it may be necesssary to factor the matrix consisting of the columns of simultaneous iterates into a product of an orthogonal matrix Q and a matrix

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

where R_1 is upper triangular. Common methods for this are Householder orthogonalization and the modified Gram-Schmidt methods. As an added bonus the Householder method also provides a basis for the orthogonal complement $M^{\perp}$. (See Golub and Van Loan [14] for details). Now the problem

$$(I - G)\epsilon_i = r_i \tag{2.37}$$

is to be projected onto M. Suppose that a well conditioned basis B' has been picked to represent G_{11}. The basis B' can be orthonormalized cost effectively to a basis B for small dimension dominant invariant subspaces. The projected matrix problem is find $\bar{\epsilon}_i \in X_i =$ span B such that

$$\pi_i((I - G)\bar{\epsilon}_i - \bar{r}_i) = 0 \tag{2.38}$$

where π_i is the orthogonal projection onto $X_i=$ span B. The actual matrix problem is formulated by picking a matrix of orthogonal projection onto B, namely $\pi_n = QQ^H$ where the columns of Q (or the rows of Q^H) are the elements of B. Applying to (2.37) and invoking linear independence of the elements of B leads to the system

$$Q^H((I - G)Q\eta_i - Q^H r_i) = 0. \tag{2.39}$$

Let $B = Q^H(I - G)Q$ and $\bar{r}_i = Q^H r_i$ so that the system (2.39) is expressed as $B\eta_i = \bar{r}_i$. We then have that $\bar{\epsilon}_i = Q\eta_i$ is the approximation to ϵ_i from the subspace X_i. An extrapolation then is formed from

$$x_\infty \approx y_i = x_i - \bar{\epsilon}_i. \tag{2.40}$$

which is exact on the dominant invariant subspace. Further extrapolations take the form

$$x_\infty \approx y_j = x_j + G^{j-i}\bar{\epsilon}_i. \tag{2.41}$$

The relation (2.39) is not exact on subdominant invariant subspaces and amplifications of error may result there.

The inverse computed in (2.39) could also be reused to produce an extrapolation

$$y_j = x_j + Q(B)^{-1}Q^H r_j. \tag{2.42}$$

There are many variations of the extrapolations (2.41) and (2.42) above. For instance, in (2.42) the extrapolation may proceed in the form described or at some point, a new extrapolation

$$w_j = y_j + Q(B)^{-1}Q^H s_j \tag{2.43}$$

where $s_j = k - (I - G)y_j$.

Schematically, the extrapolation can be represented by the following:

$$\begin{tikzcd}
\vdots & & & \\
x_i \arrow[r] & y_i \arrow[r] & w_i \arrow[r] & {} \\
x_{i+1} \arrow[r] & y_{i+1} \arrow[r] & w_{i+1} \arrow[r] & {} \\
x_{i+2} \arrow[r] & y_{i+2} \arrow[r] & w_{i+2} \arrow[r] & {} \\
x_{i+3} \arrow[r] & y_{i+3} \arrow[r] & w_{i+3} \arrow[r] & {} \\
\vdots & \vdots & \vdots &
\end{tikzcd} \tag{2.44}$$

In contrast to scheme (2.21), this extrapolation needs only a certain number of "startup" iterates upon which it may proceed along any path in the schematic.

2.3.3 On Approximate Eigenelements from Projections

As was done in the presence of a simple eigenvalue/eigenvector pair, the effectiveness of extrapolations based on more elaborate spectral approximations with and without the use of projection techniques can be deduced form perturbation and truncation error results which are well known. Perturbation results describe the sensitivity of the eigenvalue problem or error/residual equation to natural machine errors and are generally dependent on the conditioning of the eigensystem and the geometric and algebraic multiplicities of the eigenvalues.

It was shown by Golub and Wilkinson in [15] that small geometric with large algebraic multiplicities can cause severe problems which makes the computation of the Jordan canonical form (or other canonical forms) an impractical procedure in general. Truncation error results are important because of the realities of computation: Numerical methods for eigenelements tend to be iterative processes which must be stopped by some criteria. In the case of the QR algorithm, for example, it is by the closeness of the iterative matrix to an upper diagonal matrix. In general, we expect the extrapolation to be stable if the associated eigenproblem or error/residual equation is stable and the accuracy is of the same order as those given by the truncation error results for the associated problem.

The following theorems from [10] describe the relationship between eigenvalues of projected problems and the eigenvalues themselves in terms of the closeness of the initial subspace U. By π_n we mean the orthogonal projection onto U.

Theorem 6 *Let P be the spectral projection upon a dominant invariant eigenspace of dimension r and suppose that the r vectors $\{Px_i\}_{i=1,r}$ are independent. Then for any eigenvector θ_i of G there exists a particular $u_i \epsilon U$ such that $Pu_i = \theta_i$ and*

$$\|(I - \pi_n)\theta_i\|_2 \leq \|\phi_i - u_i\|_2 \left(\frac{| u_{r+1} |}{| u_i |} + \epsilon_n \right)^n$$

where $\epsilon_n \rightarrow 0$ as $n \rightarrow \infty$.

If G is Hermitian, $\|\theta_i - u_i\|_2 = \tan \Theta_i$ where Θ_i the acute angle between θ_i and u_i. Consider the subspace $X_k = G^k U$ and the eigenproblem

$$G\phi^{(n)} - \mu^{(h)}\phi^{(k)} \bot X_k.$$

i.e. the problem has been projected orthogonally into X_k. Let P_k be a projection onto an orthonormal basis of X_k and pick ξ_k such that $\phi^{(n)} = P_k\xi_k$. Then ξ_k

is a solution of

$$P_k^H(GP_k - \mu^{(h)}P_k)\xi_k = 0.$$

In other words, $\{\xi_k\}$ and $\mu^{(h)}$ are eigenvalues and eigenvectors of the matrix $P_k^H GP_k$. As $k \to \infty$, $P_k^H GP_k \to P^H GP$ where $P^H GP$ is the restriction of G to M expressed in terms of a basis of the dominant eigenspace. Hence the ith eigenpair $\mu_i^{(k)}, \phi_i^{(k)}(1 \leq i \leq r)$ converges to μ_i, ϕ_i $(1 \leq i \leq r)$. The following theorem is relevant to convergence and rate of convergence of the Galerkin method. In particular, we state a theorem similar to theorem 4 for the eigenvalue/eigenvector via a projected matrix iteration.

Theorem 7 (Chatelin-Saad [10]) *If the vectors Px_i, $1 \leq i \leq r$, are independent and $|\mu_r| > |\mu_{r+1}|$, the method of orthogonal iteration is convergent. If, moreover, the ith dominant eigenvalue is simple, the rate of convergence of the ith eigenpair is of the order $|\mu_{r+1}/\mu_r|$, for $i = 1, 2, \ldots, r$. If A is hermitian, the rate of convergence is squared to $|\mu_{r+1}/\mu_r|^2$.*

As before, subspace convergence, not necessarily simple eigenspace convergence, is being described here. Spectral decompositions of matrices are in general more expensive than L-U decompositions, so for eigenvector/eigenvalue extrapolations to be more useful than error/residual extrapolations, the size of the subspace for the projection should be small or the extrapolations reiterated before recomputing.

2.4 Extensions to Lanczos type methods

Assume that G is hermitian of order N. Up to now, fixed subspaces of the form G^nU where U is a subspace spanned by an initial set of column vectors has been considered. The dimension of the subspaces G^nU are fixed and equal to the dimension of U. In the generalized Lanczos idea, an increasing sequence of subspace, $X_n = \text{span}\ \{U, GU, ..., G^{n-1}U\}$, $n = 1,2,3,...,v < N$, is considered for projecting the matrix G onto.

Consider first a single vector x and $X_n = \text{span}\ \{x, Gx, ..., G^{n-1}x\}$ for $n = 1,2,...,v < N$. The Lanczos algorithm is an iterative method for representing the orthogonal projection onto X_n of $G|_{X_n}$ in terms of a tridiagonal matrix with respect to an orthonormal basis. In N iterations then, the matrix G would be tridiagonalized. Generally the Lanczos methods have a problem with numerical round-off, so often the Householder method in used in conjunction with the Lanczos method to reorthogonalize the basis in case of a loss of orthogonality. For purposes here, however, the method will be used with a small number of steps for projection extrapolations. In particular, the estimated eigenvalues can be found by applying one of several well known algorithms to the hermitian tridiagonal matrix. These include the method of bisection, the quotient difference algorithm and the symmetric QR algorithm. For purposes of extrapolation the Lanczos method is useful in estimating a number n_0 of the larger eigenvalues of G.

A brief summary of the idea behind the Lanczos method and the accuracy of the approximations for the eigenelements and solution of linear systems obtained after $\mu < N$ steps will now be given. This method is used for a small number of steps because of strong sensitivity to a gradual accumulation of round off error that results in a loss of orthogonality. Its most attractive feature here is its iterative means for computing an orthogonal projection of G onto the increasing sequence of Krylov subspaces $X_n = \{x, Gx, G^2x, ..., G^nx\}$. If at any stage it is decided that the subspace X_n is large enough for some $u = n_0 < N$, a switch may be made to a fixed size subspace iteration. Figure 2.2 on the next page has a sequence transfrom version of the Lanczos method.

First, we treat convergence of eigenelements. The convergence of the eigenvalues depends, not surprizingly, on the choice of the starting Krylov vector x.

Figure 2.2: Lanczos Method

1. Generate a subspace $X_\mu = \{x, Gx, G^2x, \ldots, G^\mu x\}$ $\mu \leq N-1$.

2. Set $v_1 = x/\|x\|_2$, $a_1 = v_1^H G v_1$, $b_1 = 0$,

3. for $j = 1, 2, \ldots, \mu \leq n-1$, do

$$x_{j+1} = Gv_j - a_j - b_j v_{j-1}, \quad b_{j+1} = \|x_{j+1}\|_2;$$

$$v_{j+1} = x_{j+1}/\|x_{j+1}\|_2, \qquad a_{j+1} = v_{j+1}^H G v_{j+1}.$$

3a. (Polynomial Acceleration) Determine the eigenelements of the matrix T_μ with diagonal elements a_i, $i = 1, \ldots, \mu$ and off-diagonal elements b_i, $i = 2, \ldots, n$. Use 2.19 or 2.31.

3b. (Error Equation Acceleration) Solve the projected problem

$$(I - T_\mu)\eta = Q^H r_i$$

and update using $x_1 = x_0 + Q\eta$.

The Lanczos process itself amounts to an approximation of the eigenelements of a $k \times k$ matrix with respect to an orthonormal basis A' representing $A|_E$ where E is the subspace spanned by $(\theta_i^H x)\theta_i$ for $i = 1, k$ and θ_i is the ith normalized ith eigenvector of G. The eigenelements of G' are simple because of the tridiagonal nature of the construction [10]. The following theorem is adapted from [29].

Theorem 8 *If $(\theta_i^H x)\theta_i \neq 0$ then $0 \leq \mu_i - \mu_i^{(n)} \leq k\beta_{in}^2$ and $\|\theta_i - \theta_i^{(n)}\|_2 \leq k\beta_{in}$ where $\theta_i^{(n)}, \mu_i^{(n)}$ is the ith simple eigenvector, eigenvalue of A', k is a constant and*

$$\beta_{in} = \tan(\theta_i, x)\frac{K_i}{P_{n-1}\left(1 + 2\frac{\mu_i - \mu_{i+1}}{\mu_{i+1} - \mu_{\min}}\right)}$$

$$K_i = \begin{cases} 1 & i = 1 \\ \prod_{j=1}^{i-1} \frac{\mu_j - \mu_{\min}}{\mu_j - \mu_i} & i > 1 \end{cases}$$

and P_i is the ith Chebychev polynomial on $[-1, 1]$.

From the previous theorem it can be seen that the eigenelement bounds may be weakened in the case of tightly clustered eigenvalues. One way to get around this in the case of hermitian matrix iterations is the block-Lanczos technique which will now be described.

In the *Block Lanczos technique* an initial subspace U spanned by k orthogonal vectors $x_1, x_2, x_3, x_4, ...x_\mu$ replace x in the Lanczos algorithm. Hence an increasing sequence of subspaces

$$X_\mu = \{U, AU, ...A^\mu U\}$$

is considered in analogy with the Kyrlov subspace $\{x, Ax, \ldots, A^\nu x\}$ of the Lanczos method. The block-Lanczos technique iteratively produces a projecton of G onto X_μ in the form of a block triangular matrix $\overset{\square}{T}_\mu$ with respect to an orthonormal basis which spans X_μ. The difference is that the orthonormal matrices are now produced in blocks as opposed to individually as in the Lanczos method.

The block-Lanczos method has an advantage over the unblocked version in the situation when repeated eigenvalues arise, in which case Lanczos would have convergence problems. This is analogous to what happens with respect to single vector and simultaneous iteration. The algorithm is quite similar to the Lanczos algorithm (see figure 2.3 next page). We do not worry here about the situation in which an exact invariant subspace is computed, since this is unlikely when these techniques are used as extrapolators and the material is adequately covered in [14] and [10]. The convergence rates of the block Lanczos method are derived similar to those of the Lanczos method.

Theorem 9 (Underwood [14]) *Let A be an $n \times n$ real symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2, \geq \ldots, \geq \lambda_n$ and corresponding orthonormal eigenvectors $\theta_1, \theta_2, \ldots, \theta_n$. Let $\mu_1 \geq \mu_2 \geq, \ldots, \geq \mu_p$ be the p largest eigenvalues of the matrix $\overline{T_j}$ obtained after j steps of the block Lanzcos iteration. If $Z_1 = [z_1, z_2, \ldots, z_p]$ and $\cos(\phi_p) = \sigma_p(Z_1^T X_1) > 0$, then for $k = 1, \ldots, p$*

$$\lambda_k \geq \mu_k \geq \lambda_k - \epsilon_k^2 \tag{2.45}$$

where

$$\epsilon_k^2 = \frac{(\lambda_1 - \lambda_k)\tan(\phi_p)}{\left[p_{j-1}\left(\frac{1+\gamma_k}{1-\gamma_k}\right)\right]^2} \quad \gamma_k = \frac{\lambda_k - \lambda_{p+1}}{\lambda_k - \lambda_n} \tag{2.46}$$

and $P_{j-1}(z)$ is the $(j-1)$st Chebychev polynomial.

Careful examination of theorem 8 and theorem 9, especially the constants γ_k above reveals the superior convergence of the block method when there are clustered eigenvalues. In particular the following results follow [14]:

a. the amount of work required to compute $\overline{T_j}$ is proportional to p^2; and

b. the overhead associated with a Lanczos step from $\overline{T}_j$ to $\overline{T}_{j+1}$ does not increase much with increased p.

Figure 2.3: Block Lanczos Method

1. Generate a subspace $X_\mu = \{U, GU, G^2U, \ldots, G^\mu\}$ $\mu \leq N-1$ from a starting orthonormal set $U = \{x_1, x_2, \ldots x_r\}$ $r \ll N$. Let Q_0 be the $N \times r$ matrix $(x, x_2, \ldots, x_r)$.

2. Set $\overset{\square}{A}_1 = Q_0^H G Q_0$; $\overset{\square}{B}_1 = 0$.

3. For $j = 1, 2, \ldots, n-1$ do $D_j = GQ_{j-1} - Q_{j-1} \overset{\square}{G}_j - Q_{j-2} \overset{\square}{B}_j^H$; perform the orthonormalization of D_j, $D_j = Q_j R_j$, where R_j is an $\mu \times \mu$ regular triangular matrix; and set $\overset{\square}{B}_{j+1} = R_j$, $\overset{\square}{G}_{j+1} = Q_j^H G Q_j$.

3a. (Polynomial Acceleration) Determine the eigenelements of the matrix $\overset{\square}{T}_\mu$ with diagonal blocks $\overset{\square}{A}_i$, $i = 1, \ldots, \mu$ and off- diagonal blocks $\overset{\square}{B}_i$, $i = 2, \ldots, n$. Use 2.19 or 2.31.

3b. (Error Equation Acceleration) Solve the projected problem

$$(I - T_\mu)\eta = Q^H r_i$$

and update using $x_1 = x_0 + Q\eta$.

Figure 2.4: Arnoldi Method

```
1. Generate a subspace $X_\mu = \{x, Gx, G^2x, \ldots, G^\mu\}$  $\mu \leq N - 1$.

2. Set $v_1 = x/\|x\|_2$,  $h_{11} = v_1 G v_1$,  $h_{21} = 0$,

3. for $j = 1, 2, \ldots, \mu \leq n - 1$, do

       $x_{j+1} = Gv_j - \sum_{i=1}^{j} h_{ij} v_j$,  $h_{j+1,j} = \|x_{j+1}\|_2$;
       $v_{j+1} = x_{j+1}/\|x_{j+1}\|_2$,  $h_{i,j+1} = v_{j+1}^H G v_{j+1}$ for $i \leq j + 1$..

3a. (Polynomial Acceleration) Determine the eigenelements of the
    upper-hessenberg matrix $H_\mu$. Use 2.19 or 2.31.

3b. (Error Equation Acceleration) Solve the projected problem

        $(I - H_\mu)\eta = Q^H r_i$

    and update using $x_1 = x_0 + Q\eta$.
```

Use with the extrapolation is summarized in table 2.3.

Generalizing the Lanczos method to non-hermition matrices leads to the Arnoldi algorithm. Like the Lanczos algorithm , the Arnoldi method realizes a projection of G onto the Krylov subspace $X_n = \{x, Gx, ..., G^{n-1}x\}$ in which G is represented by an *upper hessenberg matrix* $H_n = (h_{ij})$, $h_{ij} = 0 \;\; i > j + 1$, with respect to an orthonormal basis $\{q_i\}_{i=1}^{n}$. As in the Lanczos method, the basis $\{q_i\}_{i=1}^{n}$ and matrix H_n are computed iteratively. Again, we do not worry about small degree minimal polynomials because their existence is unlikely. The algorithm for the Arnoldi method is given in figure 1.4.

Complete analysis of the rate of convergence of the Arnoldi method is a difficult problem in complex approximation theory and the theory is not as complete as for the Lanczos and block Lanczos methods. The degree of convergence of the eigenelements of the orthogonal projection G_n of a nonhermitian matrix G, which the Arnoldi algorithm realizes, can be bounded by the constant $c\epsilon_i^{(n)}$ where $\epsilon_i^{(n)} = \min_{p\in P_{n-1},\ p(\mu_i)=1} \max_{j\neq i} |p(\mu_j)|$ when the problem is not too ill conditioned. (See Chatelin [10]). A block Arnoldi method is also possible for the case when the eigenvalues of G are clustered much in the same way the block Lanczos method was used to resolve tightly clustered eigenvalues for the Hermitian matrix iterations.

To understand when the Arnoldi method will be effective, let π_n be the orthogonal projection onto $X_n = \{x, Gx, ...G^k x\}$. Such a projection is formed implicitly by the Arnoldi method. The problem

$$\pi_n(I - G)x_n = \pi_n k \tag{2.47}$$

is solved for $x_n \epsilon X_n$. Let $A = I - G$ and $x_\infty = A^{-1}k$. To bound the error $x_n - x_\infty$, note that $x_n = (\pi_n A|_{X_n})^{-1}\pi_n b$ and $x_\infty = \pi_n x_\infty + (I - \pi_n)x_\infty$. Hence,

$$\begin{aligned} x_n - x_\infty &= \left[(\pi_n A|_{X_n})^{-1}\pi_n A - \pi_n - (I - \pi_n)\right] x_\infty \\ &= \left[(\pi_n A|X_n)^{-1}\pi_n A\,(I - \pi_n) - (I - \pi_n)\right] x_\infty \\ &= \left[\left((\pi_n A|_{X_n})^{-1}\pi_n A - I\right)(I - \pi_n)\right] x_\infty \end{aligned}$$

Hence,

$$\|x_n - x_\infty\|_2 \leq \|((\pi_n A|_{X_n})^{-1}\pi_n A - I)\|_2 \text{ dist } (x_\infty, X_n) \tag{2.48}$$

where dist $(x_\infty, X_n) = \min_{x\epsilon X_n} \|x - x_\infty\|_2$. If $\|\pi_n(A|_{X_n})^{-1}\pi_n A\|_2 \leq M$, which says roughly that $\pi_n A$ should not differ greatly from $\pi_n A \pi_n$, then

$$\|x_n - x_\infty\|_2 \leq (1 + M) \text{ dist } (x_\infty, X_n). \tag{2.49}$$

An advantage of the Lanczos and Arnoldi methods for extrapolations is the convenient form in which the matrix projections are represented. This leads to highly efficient LU decompositions for solving the projected linear systems. For all methods, the projections are used in a very incomplete form–the order of X_n does not get close to the order of the system.

In summary, the projections all have the form

$$H_n x_n = (Q_n^H (I - G) Q_n) x_n = Q_n^H k$$

or

$$(H_n - \mu) x_n = (Q_n^H (G - \mu) Q_n) x_n = 0$$

where H_n is tridiagonal (upper Hessenberg) in the Lanczos (Arnoldi) case. In the blocked versions the matrices are blocked. When a system is solved, an appropriate lower upper LU-type or iterative method is used which takes advantage of the special forms. The situation is similar for the eigenelement extrapolations. For more details of the Lanczos methods for linear symmetric systems see Golub and Van Loan [14] where it is also shown that the Lanczos method for a symmteric positive definite system is equivalent to the well known conjugate gradient technique. This algorithm requires a matrix multiplication and 5n flops per iteration, which makes it highly efficient. The Arnoldi algorithm, with its greater range of application, will cost more storage and flops.

The methods considered so far were all *orthogonal* projection methods. When $(I - G)$ or $(G - \mu I)$ is nonhermitian, non-orthogonal (or oblique) projections can be considered as well. The problems can be presented in an abstract setting as follows. Given two sequences of subspaces X_n and Y_n with $\dim X_n = \dim Y_n$ approximate the problems by

1. Find $\mu \in C$, where $0 \neq \phi_n \in X_n$ such that

$$G\phi_n - \mu_n \phi_n \text{ is orthogonal to } Y_n \qquad (2.50)$$

or

2. Find $\epsilon_n \in C$, where $0 \neq \epsilon_n \in X_n$ such that

$$(I - G)\epsilon_n - r_n \text{ is orthogonal to } Y_n. \qquad (2.51)$$

A classic example of an oblique projection method is the incomplete biorthogonalization method of Lanczos. In this method, given x and y such that $x^H y = 0$, the subspaces used are $X_n = \{x, Bx, \ldots B^{n-1}x\}$ and $Y_n = \{y, B^H y, \ldots (B^H)^{n-1} y\}$ where $B = G$ or $B = (I - G)$. Matrices generated are tridiagonal in this case and a projection can be accomplished in an iterative scheme similar to the Lanczos method.

To understand the matrix problem for oblique projections, let $Q_n Q_n^H$ be the matrix of orthogonal projection onto X_n and $P_n P_n^H$ be the matrix of orthogonal projection onto y_n. Then the matrix version of the problem is

Find $\mu_n \in C$ and $0 \neq \xi_n \in C^v$ such that

$$P_n^H G Q_n \xi_n - \mu_n P_n^H Q_k \xi_n = 0. \qquad (2.52)$$

If $(P_n^H Q_n)^{-1}$ exists, the problem may be reformulated as

$$(P_n^H Q_n)^{-1} P_n^H G Q_n \xi_n - \mu_n = 0. \qquad (2.53)$$

Methods for solving the generalized eigenproblem (2.52), including the standard QZ algorithm, are described in Golub and Van Loan [14].

Chapter 3

Extrapolations Based on Generalized Inverses

To this point, the methods were considered based upon (1) determination of eigenelements, (2) determination of eigenelements of a projected problem upon a space spanned by dominant eigenelements or (3) solving a projected residual equation on a dominant eigenspace. Approaches (2) and (3) were essentially *least squares* type approaches. In this chapter we are concerned with the minimal polynomial extrapolation technique (MPE) due to Cabay and Jackson [9] and improved by Sidi, Smith and Ford [30] which uses a different approach that is more concerned with the sequence of iterates. When ℓ_2 minimization is used, it is equivalent to a projection of chapter 2; however, the form of this method makes is much easier to extend to norms other than ℓ_2.

3.1 The MPE Method

Essentially, the MPE technique is a futile attempt to find an annihilating polynomial of a vector x_0 with respect to the matrix G; that is, a polynomial $p(z)$ such that

$$p(G)x_0 = \sum_{i=1}^{r} c_i G^i x_0 = 0, \qquad c_0 = 1. \tag{3.1}$$

In most circumstances, factoring such a polynomial does not in general produce accurate estimates of the eigenvalues (see Wilkinson [33]). The purpose of finding such a polynomial can be illustrated by considering the matrix iteration

$$x_{i+1} = Gx_i + k \qquad i = 1,2 \tag{3.2}$$

with fixed point

$$x_\infty = Gx_\infty + k. \tag{3.3}$$

Let $\epsilon_i = x_i - x_\infty$ and $\Delta x_i = x_{i+1} - x_i$.

Definition 3 *The annihilating monic polynomial $p(z)$ of G with respect to a vector v_0, i.e. $p(G)v_0 = 0$, of smallest degree will be called the* minimal polynomial of G with respect to v_0.

It is easily shown that the minimal polynomial is unique.

The following theorem is well known (see for instance Sidi, Smith and Ford [30]). Since the matrix iterations considered here are nonsingular, assume that zero is not a root of $p(z)$.

Theorem 10 *The minimal polynomial of G with respect to Δx_i and ϵ_i are the same.*

Proof $(I - G)\epsilon_i = \Delta x_i$, hence $P(G)\Delta x_i = 0 \Leftrightarrow P(G)\epsilon_i = 0$ since G commutes with $(I - G)$ and $(I - G)^{-1}$. ∎

In general matrix iterations, the vectors Δx_i are known quantities while the vectors ϵ_i are unknown. The MPE technique consists of determining the coefficients of $p(z)$ by "solving" the (usually overdetermined) system

$$p(G)\Delta x_0 = 0 \tag{3.4}$$

and then extrapolating by solving $p(G)\epsilon_0$ for x_∞ (note that $G\epsilon_i = \epsilon_{i+1}$):

$$x_\infty = \frac{p(G)}{p(1)} x_0 \tag{3.5}$$

where $p(1)$ is $p(z)$ evaluated at the scalar value 1. Unfortunately, for the large vector iterations found in the numerical examples in chapter 5 of this dissertation, it is unlikely that there is any polynomial $p(z)$ of reasonably small degree such that

$$p(G)\Delta x_0 = 0. \tag{3.6}$$

So in keeping with the classical idea of Lanzcos, Cabay and Jackson recommend finding a polynomial of the form

$$p(G)\Delta x_0 = \delta \tag{3.7}$$

where δ is some small vector quantity. This is just a linear system

$$\left[G^k \Delta x_0, G^{k-1}\Delta x_0, \ldots, \Delta x_0\right] \mathbf{c} = \delta \tag{3.8}$$

where $\mathbf{c}$ are the coefficients of the polynomial. Keep in mind that the system is not necessarily square. In the underdetermined case, this may be accomplished by finding

$$\min_{p \in P(r)} \|p(G)\Delta x_0\| \tag{3.9}$$

where $\| \cdot \|$ some norm (usually ℓ_2) and $P(r)$ is the set of polynomials with complex coeficients of degree less than r. Generally r is not so large as to make the problem impractically expensive. We start with the ℓ_2 and the vector of coefficients $[1, c_1, c_2, ..., c_r]$ will be the vector among those that solve (3.7) with minimal ℓ_2 norm.

Sidi, Smith and Ford solve (3.7) by considering a solution of minimal ℓ_2 norm which minimizes

$$\|Gc - \delta\| \tag{3.10}$$

where $G = [\Delta x_1, \Delta x_2, ..., \Delta x_r]$, $c = [c_1, c_2, ...c_r]^T$ and $\delta = \Delta x_0$. Actually any vector Δx_i for $i = 0, 1, \ldots r$ may be isolated as the δ term and it may be advantageous to minimize

$$\|Gd - \delta\| \tag{3.11}$$

where $G = [\Delta x_0, \Delta x_i, ... \Delta x_{r-1}]$, $c = [d_1, d_2, ..., d_r]^T$ and $\delta = -\Delta x_r$. Since in many sequences the latter terms are smaller and more likely to be similar, this approach would tend to avoid rank deficiency in the matrix G.

Once the polynomial p(z) is found, the *assumption* $p(G)\epsilon = 0$ is made and the extrapolation is made by solving $p(G)\epsilon = 0$ for x_∞ to produce in actuality an approximation y_i to x_∞:

$$y_i = \frac{p(G)}{p(1)} x_i = q(G) x_i. \tag{3.12}$$

Since the extrapolated vectors y_i are linear combinations of the iterates, y_i lies in the Krylov subspace $\{x_i, Gx_i, G^2 x_i, ..., G^r x_i\}$ and one would suspect that if $p(z)$ is an approximate minimal polynomial for $x_i - x_\infty$ $i = 0, 1, 2$ then it also be an approximate minimal polynomial for $y_i - x_\infty$ $i = 0, 1, 2, \ldots$. Hence the extrapolation may be reapplied to y_i to produce

$$z_i = [q(G)]\, y_i = q(G)^2 x_i \tag{3.13}$$

Figure 3.1 on the next page describes the extrapolation. Three columns are used for simplicity; however, the actual extrapolation has no such limitations.

At any point in the iteration, an extrapolant, say z_k, may be used as a starting vector for the iteration

$$x_{i+1} = Gx_i + k \ , \ x_0 = z_k. \tag{3.15}$$

It has been found that it is beneficial to do this until (1) there is an increase in some norm of successive residuals or (2) storage limitations become a factor. For further analysis, an explicit representation of the transform (3.11) minimized with ℓ_2 norm is needed. Let $(M \overset{i}{\leftarrow} \delta)$ denote the matrix M with its ith column replaced by δ. Solving $\|Gc - \delta\|$ gives

$$\begin{aligned} c_0 &= 1 \\ c_i &= \frac{\det(M \overset{i}{\leftarrow} \delta)}{\det M} \quad i = 1, \ldots \end{aligned} \tag{3.16}$$

Figure 3.1: Repeated Application of MPE

$$\begin{tikzcd}
x_0 & & \\
x_1 & & \\
x_2 & & \\
\vdots & & \\
x_r \arrow[r] & y_0 & \\
x_{r+1} \arrow[r] & y_1 & \\
x_{r+2} \arrow[r] & y_2 & \\
\vdots & \vdots & \\
x_{2r} \arrow[r] & y_r & \\
x_{2r+1} \arrow[r] & y_{r+1} & \\
x_{2r+2} \arrow[r] & y_{r+2} & \\
\vdots & \vdots & \\
x_{3r} \arrow[r] & y_{2r} \arrow[r] & z_0 \\
\vdots & \vdots & \vdots
\end{tikzcd} \tag{3.14}$$

where

$$M = \begin{bmatrix} (\overline{\Delta x_1}, \Delta x_1) & (\overline{\Delta x_1}, \Delta x_2) & \cdots & (\overline{\Delta x_1}, \Delta x_r) \\ (\overline{\Delta x_2}, \Delta x_1) & (\overline{\Delta x_2}, \Delta x_2) & \cdots & (\overline{\Delta x_2}, \Delta x_r) \\ \vdots & & \vdots & \vdots \\ (\overline{\Delta x_r}, \Delta x_1) & (\overline{\Delta x_r}, \Delta x_2) & \cdots & (\overline{\Delta x_r}, \Delta x_r) \end{bmatrix}$$

and

$$\delta = [(\overline{\Delta x_1}, \Delta x_{r+1}), (\overline{\Delta x_2}, \Delta x_{r+1}), \ldots, (\overline{\Delta x_r}, \Delta x_{r+1})]^T.$$

Hence in (3.11)

$$p(z) = \sum_{i=0}^{k} \gamma_i z^i, \quad \gamma_i = \frac{c_i}{\sum_{k=1}^{r} c_k} \quad i = 0, \ldots, k \tag{3.17}$$

and

$$
\begin{aligned}
y_i &= \sum_{n=0}^{k} \gamma_n x_{n+i}. \\
&= \frac{D(x_n, x_{n+1}, \ldots, x_{n+r})}{D(1,1,\ldots,1)}.
\end{aligned}
\tag{3.18}
$$

where

$$
D(\sigma_1, \sigma_2, \ldots, \sigma_{r+1}) = \det \begin{bmatrix} \sigma_1 & \sigma_2 & \ldots & \sigma_r \\ u_{1,1} & u_{1,2} & \ldots & u_{1,r+1} \\ u_{2,1} & u_{2,2,} & & u_{2,r+1} \\ \vdots & \vdots & & \vdots \\ u_{r,1} & u_{r,2} & & u_{r,r+1} \end{bmatrix}
\tag{3.19}
$$

where $u_{i,j} = (\overline{\Delta x_i}, \Delta x_j)$. Starting at the nth term of the sequence, using r vectors, the transform is easily generalized to

$$
y_{n,m,r} = \frac{\mathcal{D}(x_n, x_{n+1}, \ldots, x_{n+r})}{\mathcal{D}(1,1,\ldots,1)}.
\tag{3.20}
$$

where

$$
\mathcal{D}(\sigma_1, \sigma_2, \ldots, \sigma_r) = \det \begin{bmatrix} \sigma_1 & \sigma_2 & \ldots & \sigma_r \\ u_{m,m} & u_{m,m+1} & \ldots & u_{m,m+r} \\ u_{m+1,m} & u_{m+1,m+1} & & u_{m+1,m+r} \\ \vdots & \vdots & & \vdots \\ u_{m+r-1,m} & u_{m+r-1,m+1} & & u_{m+r-1,m+r} \end{bmatrix}.
\tag{3.21}
$$

This interesting determinantal form turns up again in the next chapter. In fact, for certain symmetric matrices $\mathbf{U} = (u_{i,j})$, these are just *Padé approximants* of a certain formal power series!

Sidi has done a direct and extensive convergence analysis for a single column MPE method applied to the vector sequence $\{x_n\}$ over a general complex inner product space B which has an asymptotic expansion of the form.

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

where the λ_i are scalar quantities and the rest are vector quantities and $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots$.

The convergence analysis consists of writing the extrapolation in the form of a ratio of two determinants

$$s_{n,k} = \frac{D(x_n x_{n+1}, \dots, x_{n+r})}{D(1, 1, \dots, 1)} \tag{3.22}$$

and then doing an extensive analysis using the multilinearity of the determinants and Hadamard type inequatities on the error

$$s_{n,k} - x_\infty = \frac{D(x_n - x_\infty, x_{n+1} - x_\infty, \dots, x_{n+1} - x_\infty)}{D(1, 1, 1, \dots, 1)}. \tag{3.23}$$

His analysis showed that MPE is an accelerator where

$$\frac{\|s_{n,k} - s\|}{\|x_{n+k+1} - s\|} = O\left[\left(\frac{\lambda_{k+1}}{\lambda_i}\right)^n\right] \tag{3.24}$$

when there is a separation in the spectrum $|\lambda_k| > |\lambda_{k+1}|$. Also in this paper is a stability analysis that shows that the coefficients of the semi-iterative method produced by MPE have a bounded absolute sum for a fixed value of k as $n \to \infty$. That is, the extrapolation $s_{n,k}$ is weighted average of a $k+1$ length window of sequence values:

$$\begin{array}{rl} s_{n,k} = & \sum_{j=0}^{k} \gamma_j^{(n,k)} x_{n+j} \\ \text{and} & \sum_{j=0}^{k} \gamma_j^{(n,k)} = 1 \end{array} \tag{3.25}$$

and "stability" is shown be bounding

$$\sup_n \sum_{j=0}^{k} \left|\gamma_j^{(n,k)}\right| < \infty. \tag{3.26}$$

This is roughly equivalent to showing one of the regularity conditions in the Toeplitz theorem [34]. It is apparent that this definition of stability is the adequate definition for the description of the evolution of small perturbations in $x_n, x_{n+1}, \dots, x_{n+k}$.

Usually, the regularity over all C_S, points out a limitation of the sequence transfrom. Again it is the old addage: Transforms which are highly stable tend to be limited in their effectiveness on particular types of error structures because of their broad scope. Borderline stable techniques or even unstable transforms, the "violent" summability techniques in the terminology of G.H. Hardy, tend to produce more dramatic results on the sequences for which they are intended. It is apparent that for most residuals Δx_i for converging iterative methods that we have $\gamma_j^{(n,k)} \to 0$ as $k \to \infty$, so that MPE behaves as a regular transform.

It is apparent that the reextrapolation based upon the coefficients determined from (5.7) will also be stable, but there is no evidence to indicate that there will be an improved rate of convergence. In fact, one would be lead to believe the opposite in the case of matrix iterations: If the component of error has been taken out in dominant eigenspace for which the coefficients were determined, then one would hardly believe the same coefficients would be adequate for error in the sub-dominant eigenspaces. An example of this would be Aitken's method on a scalar sequence x_n of the form

$$x_n = x_\infty + cp^k. \tag{3.27}$$

A single extrapolation gives the fixed point x within a certain accuracy, while further extrapolations on the sequence of extrapolants would consist of inner products of numerical "noise", and likely to produce ridiculous numbers. For details of this situation, see Graves-Morris [17]. However, if the polynomial produced by the method is a good polynomial, but only removes part of the error, then we would expect reextrapolations to work.

We close this section with a formulation of the MPE method as a summability technique. We also show the close similarity between the projection methods of chapter 2, where a problem was completely on to a subspace, and the MPE method where we look for elements of minimal norm in a linear variety.

Let x_i be a given vector sequence and consider a summability method

$$\overline{x}_i = \sum_{j=1}^{i} \mu_{ij} x_j$$

where $\sum_{j=1}^{i} \mu_{ij} = 1$. Then the error in the sum is given by

$$\overline{\epsilon}_i = \sum_{j=1}^{i} \mu_{ij} \epsilon_j = p(G)\epsilon_1$$

where $p(\cdot)$ is a polynomial such that $p(1) = 1$. Let $r_i = -\Delta x_i$. Applying $(I - G)^{-1}$ to both sides gives the residual $\overline{r}_i$ of the summability technique:

$$\overline{r}_i = \sum_{j=1}^{i} \mu_{ij} r_j = p(G)r_0 = \alpha p_\alpha(G)r_0$$

where p_α has leading coefficient 1. The question now is the choice of the polynomial $p(G)$. The MPE method is formed by minimizing $p(G)r_0$ or equivalently $p_\alpha(G)r_0$ in ℓ_2. The equivalence of these statements follows form different but equivalent versions of the projection theorem [22]. In other words the residual $\overline{r}_i$ is in the linear variety $V = \{r| \ r = \sum_{i=1}^{i} \mu_{ij} r_j$ and $\sum_{j=1}^{i} \mu_{ij} = 1\}$. and therefore orthogonal to the subspace $M = V - r, \ \ r \in V$ implied by the linear variety V. Hence we finding x in span$\{x_1, x_2, \ldots, x_k\}$ such that $((I - G)x - k) \perp M$.

3.2 A Generalization of MPE

In the previous section we used the residual Δx_i of a matrix iteration for determination of an annihilating polynomial. This is the "residual" polynomial approach [6]. The presence of the so-called residual polynomial dominates Krylov subspace type methods such as the conjugate gradient method and its generalizations to be discussed.

Residual polynomial methods often seem to lose their effectiveness when the direction of the residual is not a good approximation to the direction of the error, which would happen in iterative schemes when the matrix is nonhermitian, nondefinite, or defective. This is because a small norm of $r_n = P_n(G)r_0$ does not imply a small norm of $\epsilon_n = P_n(G)\epsilon_0$. So it may be of benefit to consider vectors other than the residuals for extrapolation.

A generalization of MPE can be made by considering the solution of

$$\frac{r(f(G))}{s(g(G))}\delta(\epsilon_i) = \varrho(\bar{\epsilon}_i) \tag{3.28}$$

where r and s are unknown polynomials (for computational reasons); f, g are matrix functions and g is chosen so that the formal quantity $1/s(g(G))$ is easily determined, they may be projections of sorts; h is a matrix or vector function and $\delta(\epsilon_i)$ is a known quantity which we will call the *iteration generalized residual*; and ϱ, which we will call the *generalized extrapolation residual*, is a (usually small) vector or matrix quantity determined to make (3.28) exact after r and s are chosen. (Warning: ϱ corresponds to the "residual" of the MPE least squares problem (3.11) and is <u>not</u> equivalent to the iteration generalized residual we are considering).

Generally, an extrapolation is obtained via some algebraic relation between $\delta(\epsilon_i) = \delta(x_i - x_\infty)$ and ϱ. For example, in MPE, we let $\delta(\epsilon_i) = \Delta x_i$; $\varrho(\bar{\epsilon}_i = \bar{x}_{i+1} - \bar{x}_i$ where $\bar{x}_i$ is the ith MPE extrapolation, and $r(f(G))/s(g(G)) = p(G)$ where p is a polynomial of degree $k < i$. For systems $Ax - b$ where A is symmetric positive definite, a method based on the case $p_n(A)r_0 = r_n$ of (3.28) where p_n is chosen at the nth step to minimize the "energy" norm of the error, $E_m = r_m^T A^{-1} r_m = \epsilon_m A \epsilon_m$ leads to the conjugate gradient method [6].

With conjugate gradient methods, (3.28) can be solved with short recursions to produce the extrapolation because the polynomials are picked a'priori to be orthogonal polynomials over the region containing the spectrum of A. With MPE (3.28) leads to a system

$$\Lambda c = \Delta x_n + \bar{\Lambda} \tag{3.29}$$

which we solve by the method of least squares. When solving (3.29) we obtained a solution

$$c = \Lambda^{\dagger}(\Delta x_n + \bar{\Lambda}) \tag{3.30}$$

where $\dagger$ denotes the *pseudoinverse* of Λ. This is a part of the area of *generalized* inversion which we now elaborate on. Three suggestions for generalized residuals δ, in increasing level of generality are

1 $\delta(\epsilon_i) = -\Delta x_i = (I - G)x_i - k = (I - G)(\epsilon_i)$. (The familiar residual).

2 $\delta(\epsilon_i) = \pi_n \epsilon_i$ where π_n is a projection (orthogonal or oblique) onto some subspace. Here, as in approach (2) page 11, we solve a projected residual equation $\delta(\epsilon_i) = Q^H(I - G)QQ^H\epsilon_i$.

3 $\delta(\epsilon_i) = f(\epsilon_i)$ where $f : \mathcal{R}^n \rightarrow \mathcal{R}^\nu$ and $\nu < n$.

3.2.1 On Generalized Inversion

The theory of generalized inverses is well developed, as is evidenced by the 1,775 articles in Nashed [24]. One approach to generalized inversion is to characterize properties of matrix inverses which may be satisfied by singular or non-square matrices. Perhaps the nicest characterization for a generalized inverse, the *pseudoinverse* was summarized by Roger Penrose in four remarkably symmetric conditions. Penrose [27] showed that for every finite matrix of real or complex elements there is a unique matrix X satisfying the equations

C1. $AXA = A$,

C2. $XAX = X$

C3. $(AX)^H = AX$

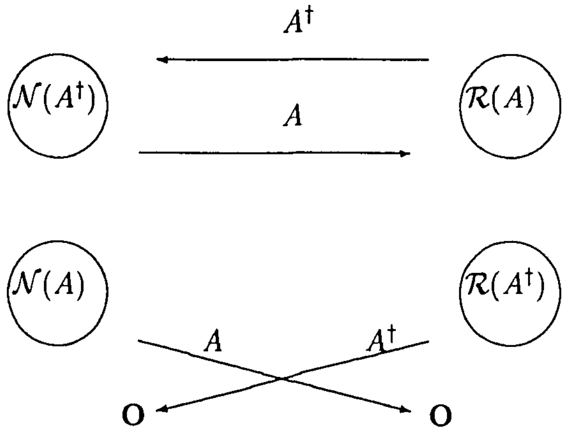
C4. $(XA)^H = XA$

where A^H denotes the conjugate transpose of A. This matrix X is commonly known as the Moore-Penrose generalized inverse (or pseudoinverse) and denoted by $A^\dagger$. These conditions amount to the requirement that $AA^\dagger$ and $A^\dagger A$ be orthogonal projections onto $\mathcal{R}(A)$ and $\mathcal{R}(A^H)$, respectively, and, consequently, is the solution to

$$\min_{X \in \mathcal{C}^{n \times m}} \|AX - I_m\|_F. \tag{3.1}$$

where "F" denotes the Frobenius norm. If A is nonsingular it can be shown that

Figure 3.2: The Pseudoinverse



$A^\dagger = A^{-1}$. In general, a matrix X which satisfy conditions $(i), (j), \ldots, (l)$ among $C1$,$C2$,$C3$,$C4$ will be called a $(i), (j), \ldots, (l)$-inverse and denoted by $A^{\{(i),(j),\ldots,(l)\}}$. The set of such matrices is denoted by $A\{(i), (j), \ldots, (l)\}$. When it is not necessary to specify the type of generalized inverse, we revert to $A^\ddagger$.

The following theorem ([5] p.103 ff.), is relevant to the MPE method, and other pseudoinversion techniques where the arguments are vectors, when an overdetermined system of the form

$$\|\Lambda c-\delta\|_2. \quad c,\delta\in C^m$$

is solved.

Theorem 11 *Let $\Lambda\in C^{n\times m}$, $\delta\in C^m$. Then $\|\Lambda c-\delta\|$ is smallest when $x=\Lambda^{\{1,3\}}\delta$ where $\Lambda^{\{1,3\}}\in A\{1,3\}$. Conversely, if $X\in C^{n\times m}$ has the property that, for all δ, $\|\delta-\Lambda c\|$ is smallest when $c=X\delta$, then $X\in A\{1,3\}$.*

Corollary 1 *A vector δ is a least-squares (l^2) solution of $\Lambda c=\delta$ if and only if*

$$\Lambda c=P_{R(A)}\delta=\Lambda\Lambda^{\{1,3\}}\delta.$$

Moreover, the general least-squares solution is

$$x=\Lambda^{\{1,3\}}\delta+(I_n-\Lambda^{\{1,3\}}\Lambda)y$$

where $\Lambda^{\{1,3\}}\in A\{1,3\}$ and arbitrary $y\in C^n$.

It can be shown that the least-squares solution of minimal l^2 norm is the Moore-Penrose generalized inverse. Furthermore, the generalized inverse is unique only when A has full column rank [5].

3.2.2 Weighted Generalized Inverses

Very often, extrapolations give an inordinate amount of weight to the most recent iterates produced [34]. To attempt to avoid this in generalized inverse extrapolations, we may choose to give different weights to different components of the the generalized residual in (3.28). A further generalization which encompasses this [5] is the minimization of a give positive definite quadratic forms in the generalized residuals, i.e. the minimization of

$$\|\delta\|=\delta^H W\delta \tag{3.32}$$

where W is a given positive definite matrix.

Let U be another positive definite matrix. If A is not of full column rank, the problem (3.32) does not have a unique solution for x, so generally we choose a solution which has minimal U- norm, i.e. among the x which solve (3.32), we pick the one such that

$$\|x\|_U^2 \equiv x^H U x. \tag{3.33}$$

is smallest. In summary, we have the problem

a. Minimize $\|\delta\|_W^2$ with the constraint that

b. $\|x\|_U^2$ is minimal.

Since every inner product $(\cdot,\cdot)$ in C^n can be represented as x^*Uy for some positive definite matrix U and every positive definite matrix has a square root[4], we will show that the problem of minimizing (3.32) and (3.33) is equivalent to a problem standard least squares problem. Given positive definite H, denote by $H^{1/2}$ the unique matrix K such that $K^2 = H$. $\left(H^{1/2}\right)^{-1}$ will be denoted by $H^{-1/2}$. Introducing the transformations

$$\tilde{A} = W^{1/2}AU^{-1/2}, \quad \tilde{x} = U^{1/2}x, \quad \tilde{b} = W^{1/2}b,$$

it can be shown [5] that

$$\|b - Ax\|_W = \|\tilde{A}\tilde{x} - \tilde{b}\|^2$$

and

$$\|x|_U = \|\tilde{x}\|_2.$$

The following result can be shown in a straightforward manner [5]:

Theorem 12 *Let $A \in C^{m\times n}$, $b \in C^m$, and let $W \in C^{m\times m}$ and $U \in C^{n\times n}$ be positive definite. Then, there is a unique matrix*

$$X = A^{(1,2)}_{(W,U)} \in A\{1,2\} \tag{3.34}$$

satisfying

$$(WAX)^H = WAX, \quad (UXA)^H = UXA. \tag{3.35}$$

Moreover, $\|b - Ax\|_W$ assumes its minimum value for $x = Xb$, and in the set of vectors x for which this minimum value is assumed, $x = Xb$ is the one for which $\|x\|_U$ is smallest.

If $Y \in C^{n\times m}$ has the property that, for all b, $x = Yb$ is the vector of C^m for which $\|x\|_U$ is smallest among those for which $\|b - Ax\|_W$ assumes its minimum value, then $Y = A^{(1,2)}_{(W,U)}$. As before, if the system is overdetermined, the solution is unique.

3.3 Pseudoinversion and Descent Methods

The MPE acceleration was evaluated by a derivative of the Gram-Schmidt process, the Modified Gram Schmidt process. This naturally leads to the question whether a class of processes related to the Gram-Schmidt process, so-called *conjugate direction methods* for the minimization of functionals, can be adapted for use as extrapolations. Here we prescribe a simple way of doing so. First we give a brief explanation of the methods. The methods will be presented, for clarity, in their normally given formulation as *direct solvers*. Their adaptation into *extrapolations* is accomplished quite easily, both theoretically and computationally. Roughly the idea is to replace the direct matrix A by the iterative matrix $I - G$ (in the iterative scheme $x_{i+1} = Gx_i + k$). With gradient methods, the matrix is treated indirectly through multiplication with a vector in a Krylov span, so the process is iterative. The same holds for $I - G$ since there is no difference between the spans $K = \{x_0, Gx_0, \ldots, G^\mu x_0\}$ and $K = \{x_0, (I-G)x_0, \ldots, (I-G)^\mu x_0\}$. Vectors determined by the gradient method will be new guesses to place in the iteration $x_{i+1} = Gx_i + k$. One way of changing direct method for $Ax = b$ to an extrapolation for an iteration $x_{i+1} = Gx_i + k$ is to make the substitution

$$x_n \longmapsto W^{-1}x_n, \quad A \longmapsto W^{-1}(I-G)W, \quad r_n \longmapsto W^{-1}s_n. \tag{3.36}$$

and then multiply the recurrence relation by W, the *preconditioner*. More generally, we make the substitution

$$x_n \longmapsto W^{\ddagger}x_n, \quad A \longmapsto W^{\ddagger}(I-G)W, \quad r_n \longmapsto W^{\ddagger}s_n. \tag{3.37}$$

where ‡ denotes a generalized inverse and then multiply the recurrence relation by W, the *pseudo-preconditioner.* It is usually a small matter to adapt existing codes for direct descent methods into extrapolations. With more work, more efficiency may by obtained by careful recombination of the terms; however, here we prefer to sacrifice some efficiency to keep the programs modular. Note the the substitution $A \longmapsto W^{\ddagger}(I-G)W$ in (3.37) includes orthogonal and oblique projections of $(I-G)$ onto subspaces specified by the generalized inverse. Often when using full inverses (3.26), the matrix W is chosen from a part of a matrix splitting of A [2].

3.3.1 Descent Methods for Extrapolation

The introductory parts of this section are adapted from the excellent book of Luenburger [22]. In descent methods an optimization problem is solved by iterating from a starting point x_0 in such a way as to decrease a "cost" functional from one step to the next. When the functional is positive definite, global convergence can be insured. As a general framework for the method, assume that we seek to minimize a functional f and that an initial point x_i is given. Iterations are constructed according to the equation

$$x_{n+1} = x_n + \alpha_n p_n \tag{3.38}$$

where α_n is a scalar and p_n is a search direction vector. The procedure for selecting the vector p_n varies from technique to technique, but, ideally, once it is chosen the scalar α_n is selected to minimize $f(x_n + \alpha p_n)$ (regarded as a function of α). Most often a direction of *descent* is chosen so that $f(x_n + \alpha p_n) < f(x_n)$ for small positive α. The scalar α_n is often taken as the smallest positive root of the equation

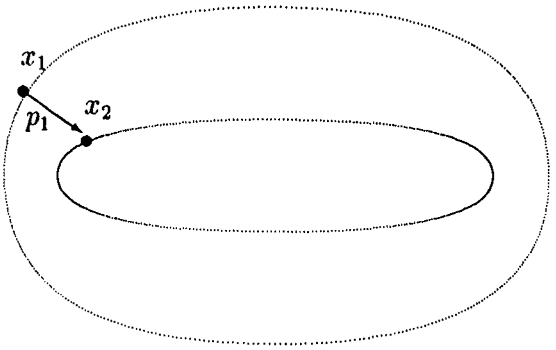
$$\frac{d}{d\alpha} f(x_n + \alpha p_n) = 0. \tag{3.39}$$

In practice it is rarely possible to find the minimizing α exactly. Instead, some iterative search or approximation is required. The essential point, however is that after an α_n is computed, we must verify that $f(x_n + \alpha_n p_n)$ is evaluated to verify that the objective has in fact decreased from $f(x_n)$; otherwise, a new α_n is chosen.

The descent process can be visualized in a Banach space X where the functional f is represented by its contours. Starting from a point x_1, one moves along the direction vector p_1 until reaching, as illustrated in figure 3.3, the first point where the line $x_1 + \alpha p_1$ is tangent to a contour of f. Alternatively, the method can be visualized, as shown in figure 3.4, in the space $R \times X$, the space containing the graph of f.

If f is bounded below, it is clear that the descent process defines a bounded decreasing sequence of functional values and that the sequence $f_n = f(x_n)$ tends toward a limit f_∞. The difficulties remaing are those of insuring that f_∞ is, in fact, the minimum of f, that the sequence of approximations x_n converges to a minimizing vector, and finally, and most difficult, that convergence is rapid enough to make the whole scheme practical. Practical schemes are then converted to practical extrapolations.

Figure 3.3: The Descent Process in X



3.0.1 The Steepest Descent Procedure

The widely used descent procedure for minimizing a functional f, the method of steepest descent, is applicable to functionals defined on a Hilbert space $\mathcal{H}$. In this method the direction vector p_n at a given point x_n is chosen to be the negative of the gradient of f at x_n. If $\mathcal{H}$ is not a Hilbert space, the method can be modified by selecting p_n at a given point x_n to be aligned with, or almost aligned with, the negative gradient.

Generally the method is used for the minimization of a quadratic functional

$$f(x) = (x, Qx) - 2(b, x)$$

where Q is a self-adjoint positive-definite operator on a Hilbert space $\mathcal{H}$. If the Rayleigh quotients

$$m = \inf_{x \neq 0} \frac{(x, Qx)}{(x, x)} \text{ and } M = \sup_{x \neq 0} \frac{(x, Qx)}{(x, x)}$$

are positive, finite numbers, finding a unique solution x_∞ to $Qx = b$ is equivalent to the minimization of f. If $r = b - Qx$, then it is easily shown that $2r$ is the negative gradient of f at the point x. Consequently, the method of steepest descent takes the form ([22]) $x_{n+1} = x_n + \alpha_n r_n$, where α_n is chosen to minimize $f(x_{n+1})$ and can be found to be $\alpha_n = (r_n, r_n)/(r_n, Qr_n)$.

Note that, according to Theorem 13 below and seen from figure 3.3, the rate of convergence depends on the eccentricity of the elliptical contours of f. For $m = M$, the contours are circular and convergence occurs in one step. The steepest descent procedure blindly plods in the direction of the residual–the direction of steepest descent–and then only changes direction when a lowest altitude is reached. So when there is a high degree of ellipticity, the procedure will likely bog down (cf. figure 3.4).

Figure 3.4: When Steepest Descent Fails

The following theorem gives the details [22].

Theorem 13 *For any $x_0 \in \mathcal{H}$, the sequence x_n defined by*

$$x_{n+1} = x_n + \frac{(r_n, r_n)}{(r_n, Qr_n)} r_n \tag{3.40}$$

converges to the unique solution x_∞ of $Qx = b$. The rate of convergence satisfies

$$\|\epsilon_n\| \leq \frac{1}{m}\epsilon_0 \left(\frac{1-c}{1+c}\right)^{2n} \tag{3.41}$$

where $c = m/M$.

3.3.3 Conjugate Direction and Conjugate Gradient Methods

Conjugate Direction Methods are a notion from the theory of *Fourier series* in an inner product space where the inner product is defined by $< \cdot, G\cdot >$ where G is symmetric and positive definite. For clarity, the procedures are considered for a linear system $Ax = b$ in a Hilbert space $\mathcal{H}$.

Consider a quadratic functional

$$f(x) = (x, Ax) - 2(x, b),$$

where A is a self adjoint linear operator on a Hilbert space $\mathcal{H}$ which satisfies

$$(x, Ax) \leq M(x, x) \tag{3.42}$$

$$(x, Ax) \geq m(x, x) \tag{3.43}$$

where $m, M > 0$ are squares of the smallest and largest singular values of A [14]. Under these conditions the unique vector x_∞ minimizing f is the unique solution of the equation $Ax = b$.

The problem is equivalent to minimizing

$$\|x - x_\infty\|_A^2.$$

This can be accomplished by minimizing $\|x - x_\infty\|_A^2$ *sequentially* over a span of linearly independent vectors $\{p_1, p_2, \ldots\}$. Suppose the sequence members $p_1, p_2, \ldots$ are orthogonal with respect to the inner product $(\cdot, Q\cdot)$. It is of benefit to do this because of a nice three term relationship that exits among the p_i's (see Theorem ?). Since $(\cdot,\cdot)$ is an inner product, the error is decreased most at the nth step by picking $x_n = x$ to be the Fourier series expansion of x_∞. If the sequence is complete, the process converges to x_∞. Even though the quantity x_0 is not known, we can compute the appropiate inner products for the expansion because $(p_i, Ax_\infty) = (p_i, b)$. More generally, a direction *A-orthogonal* to the error is orthogonal to the residual because $(p_i, Q\epsilon_i) = (p_i, r_i)$. A algorithm which accomplishes this orthogonalization, the *Method of Conjugate Directions*, is given in figure 3.5.

Figure 3.5: The Method of Conjugate Directions

Let $\{p_i\}$ be a sequence in $\mathcal{H}$ such that $(p_i, Ap_j) = 0, i \neq j$. Then for any $x_1 \in \mathcal{H}$, the sequence generated by the recursion

1. $x_{n=1} = x_n + \alpha_n p_n$
2. $\alpha_n = (p_n, r_n)/(p_n, Ap_n)$
3. $r_n = b - Ax_n$

Because of a nice three term recurrence relation for A-orthogonal vectors in $\mathcal{H}$ which span the Krylov subspace $\{x_1, Ax_1, \ldots, A^\nu x\}$, there is no need to resort to a Gram-Schmidt type process to generate the vectors p_i. The A-orthogonal sequence is generated by the algorithm shown in figure 3.6 on the next page. The Hermitian property of B is needed in only one small part of the proof for the three term recurrence (see [11] or [22]), which is unfortunate; for otherwise, we would have a three term recurrence for orthogonalizing all Krylov subspaces.

Consider again the conjugate direction method. If the Q-orthogonal sequence of directions is generated from the Krylov subspace $\mathcal{K} = \{r_1, Ar_1, \ldots, A^\nu r_1\}$ by

Figure 3.6: Three Term Recurrence for A-Orthogonal Vectors

Denote the inner product $(\cdot, A\cdot)$ by $[\cdot,\cdot]$ and consider for some self adjoint operator B the Krylov subspace $\mathcal{K} = \{x_1, Bx_1, \ldots, B^\nu x\}$. Then $\mathcal{K}$ is spanned by a A-orthogonal sequence $\{p_1, p_2, \ldots, p_\nu\}$ generated by the following algorithm:

1. $p_1 = x$
2. $p_2 = Bp_1 - \frac{[p_1, Bp_1]}{[p_1, p_1]} p_1$
3. $p_{n+1} = Bp_n - \frac{[p_n, Bp_n]}{[p_n, p_n]} p_n - \frac{[p_{n-1}, Bp_n]}{[p_{n-1}, p_{n-1}]} p_{n-1} \quad (n \geq 2)$

the three term recurrence in fig. 3.6, the conjugate gradient method of figure 3.7 on page 70 is obtained. For many "practical" variations of this method the interested reader might consult Hestenes [20]. It is shown in [14] that

$$\|\epsilon_n\| \leq \frac{4}{m}\epsilon_0 \left(\frac{1-\sqrt{c}}{1+\sqrt{c}}\right)^{2n}$$

where $c = m/M$. This is improved convergence, but there is more computational complexity in trade.

3.4 A More Sophisticated Gradient Extrapolation

Much effort has been devoted to extending the conjugate gradient (CG) and conjugate direction (CD) methods to nondefinite quadratic and nonquadratic functionals. The number is extensive, so a reader wishing to see a semblance of the full spectrum should cite the book of Hestenes [20].

Heuristically, the need for these extensions comes when the the matrix separating the error from the residual, namely $(I - G)$ in $(I - G)\epsilon_i = r_i$, has a wide spectrum or is quite nondefinite. Traditionally, the approach has been to precondition $(I - G)$, which is done in equation (3.36). Here instead we adopt the approach that the sequence is being generated by nondefinite symmetic matrix iteration. This is not a bad "local" approach because the set of symmetric matrices is dense in the Hilbert space of matrices.

3.4.1 Planar Conjugate Gradient Acceleration

This modification of the CG-algorithm enables us to obtain a critical point x_∞ of a quadratic function $F(x) = (1/2)x^H Ax - b^H x + c$ whose Hessian A is nonsingular. A is possible indefinite. When A is definite, x_∞ is an extreme point; when indefinite, a saddle point. Either way, x_∞ is a critical point and a solution to

$$F'(x) = Ax - b = 0. \tag{3.44}$$

Before proceeding, it should be mentioned that the point x_∞ is also the minimum point of the quadratic function

$$\hat{F}(x) = \frac{1}{2}|F'(x)|^2 = \frac{1}{2}|h - Ax|^2. \tag{3.45}$$

and the CG-method could be applied directly to this functional since it is quadratic.

However, when CG is applied directly to F, the algorithm may fail when A is indefinite. In CG successive critical points $x_0, x_1, \ldots$ of F were obtained on mutually conjugate lines $x = x_k + \alpha p_k$ (k=0,1,...). If in the kth step a situation in which $r - k = -F'(x_k) \neq 0$ and $d_k = p_k^H A p_k = 0$, then F has no critical point on the line $x = x_k + \alpha p_k$ and the algorithm terminates prematurely. We see how this may be caused in the case of indefiniteness. In this event we proceed by finding the critical point of F on the two dimensional plane $x = x_k + \alpha_k + \beta A p_k$. So we modify the CG-algorithm incorporating this account of the indefinite case. Details on the justification of this algorithm are given in [20]. A suggested formulation from [20] is given in figure 3.8 on page 71. A matrix form of the algorithm is given in [20].

3.5 A Generalized Inverse Extrapolation Based on Newton's Method

A generalized inverse can be used to obtain a modifications of the Newton Method for extrapolations. *Newton's method* for solving a system of m equations in n variables

$$\begin{aligned} f_1(x_1,\ldots,x_n) &= 0 && (3.46)\\ f_2(x_1,\ldots,x_n) &= 0 && (3.47)\\ &\vdots && (3.48)\\ f_m(x_1,\ldots,x_n) &= 0 && (3.49)\end{aligned}$$

or

$$f(x) = 0$$

is given by, for the case $m = n$,

$$x_{i+1} = x_i - f'(x_i)^{-1}f(x_i) \quad (k = 0,1,\ldots), \tag{3.50}$$

where

$$f'(x_i) = \left[\frac{\partial f_i}{\partial x_j}(x_k)\right]. \tag{3.51}$$

We not allow the possibility that f is a projection or a derivative of the residual and, hence, nonsingularity may not be assumed.

A natural extension is then to inquire whether a generalized inverse may be applied in (3.50), so that we have the iteration

$$x_{k+1} = x_k - (f'(x_k))^{\dagger}f(x_k) \tag{3.52}$$

which converges to x_∞, where ‡ denotes pseudoinversion, which converges to x_∞. Note that this is a nonlinear generalization of (2.42). The following theorem shows that we have convergence, but not necessarily to x_∞. As usual, $\|\cdot\|$ is a given (but arbitrary) vector norm in C^n, and a matrix norm in $C^{m\times n}$ consistent with it. For a given point $x_0 \in C^n$ and a positive scalar r, the *open ball* of radius r about x_0 is denoted by

$$B(x_0, r) = \{x \in C^n : \|x - x_0\| < r\}.$$

The *closed ball*, $\{x \in C^n : \|x - x_0\| < r\}$, is denoted by $\overline{B(x_0, r)}$.

Theorem 14 *Let the following be given:*

$$x_0 \in C^n, \quad r > 0,$$

$$f : B(x_0, r) \to C^m \quad \textit{a function}$$

$$\epsilon > 0, \quad \delta > 0, \quad \eta > 0 \quad \textit{positive scalars,}$$

and for any $x \in \overline{B(x_0, r)}$ let

$$A_x \in C^{m\times n}, \quad T_x \in C^{n\times m} \tag{3.53}$$

be matrices such that for all $u, v \in B(x_0, r)$:

$$\|f(u) - f(v) - A_v(u - v)\| \leq \epsilon\|u - v\|$$

$$\textit{for all } u, v \in B(x_0, r), \tag{3.54}$$

$$T_u A_u T_u = T_u, \quad \textit{i.e. } T_u \in A_u\{2\} \tag{3.55}$$

$$\|(T_u - T_v)f(v)\| \leq \eta\|u - v\|, \tag{3.56}$$

$$\epsilon\|T\| + \eta \leq \delta < 1, \tag{3.57}$$

$$\|T\|\|f(x_0)\| < (1 - \delta)r. \tag{3.58}$$

Then the sequence

$$x_{i+1} = x_i - T_{x_i} f(x_k) \tag{3.59}$$

converges to a point

$$x_\infty \in \overline{B(x_0, r)} \tag{3.60}$$

which solves

$$T_{x_\infty} f(x) = 0. \tag{3.61}$$

Note that (3.59), considered in the form $\Delta x_i = -T_{x_i} f(x_k)$ is a particular case of (3.28). There is a great deal of leeway in the choice of f which will be explored experimentally in chapter 6. Perhaps the nicest aspect of the Newton extrapolation (3.52) is that it is an ideal form in which to use the highly perfected singular value decomposition (SVD) for the computation of the appropriate Moore-Penrose pseudoinverses.

Figure 3.7: The Method of Conjugate Gradients

1. Select x_0 and compute

$$r_0 = b - Ax_1 = -F(x_0)$$

Iterative Steps. Having obtained x_k,r_k, and p_k by the formulas

2.

$$\alpha_k = \frac{c_k}{d_k}, \quad d_k = p_k^H A p_k, \quad c_k = p_k^H r_k \text{ or } c_k = |r_k|^2,$$

3.

$$x_{n+1} = x_n + \alpha_n p_n, \quad r_{k+1} = r_k - \alpha_k A p_k,$$

4.

$$b_k = -\frac{p_k^H A r_{k+1}}{d_k} \text{ or } b_k = \frac{|r_{k+1}|^2}{c_k},$$

5.

$$p_{k+1} = r_{k+1} + b_k p_k.$$

6. *Termination.* Terminate at the mth step if $r_{m+1} = 0$ or guesses have degrading residuals. Set $x_{m+1} = x_0$ and restart if necessary minimum point has not been reached.

Figure 3.8: Planar CG-Algorithm

1. *Initial step.* Select x_0, set $\epsilon = 1/2$, and compute

$$r_0 = -F(x_0), \;\; p_0 = r_0, \;\; q_1 = Ap_1.$$

2. *Iterative steps.* Having obtained x_k, r_k, p_k, q_k, compute

$$Ap_k, \;\; Aq_k, \;\; d_k = p_k^H Ap_k, \;\; \delta_k = p_k^H Aq_k, \;\; e_k = q_k^H Aq_k, \qquad (1)$$

$$\Delta_k = d_k = d_k e_k - \delta_k^2, \;\; c_k = p_k^H r_k.$$

3. If $|\Delta_k| \leq \epsilon \delta_k^2$, do this step, otherwise go to equation 2 in step 6.

$$a_k = \frac{c_k}{d_k}, \;\; x_{k+1} = x_k + a_k p_k, \;\; r_{k+1} = r_k - a_k Ap_k.$$

4. If $r_{k+1} = 0$ terminate, else compute

$$p_{k+1} = r_{k+1} + b_k p_k, \;\; b_k = -\frac{p_k^* A r_{k+1}}{d_k} = \frac{|r_{k+1}|^2}{c_k},$$

$$q_{k+1} = Ap_{k+1} + \beta_k p_k, \;\; \beta_k = -\frac{p_k^* AAp_{k+1}}{d_k} = -\frac{q_k^* Ap_{k+1}}{d_k}.$$

5. Increase the index k by 1 and to to equation 1 in step 2.

6.

$$\hat{c}_k = \frac{c_k e_k - \delta_k q_k^H r_k}{\Delta_k}, \;\; \hat{d}_k = \frac{d_k q_k^H r_k - \delta_k c_k}{\Delta_k}, \qquad (2)$$

$$x_{k+2} = x_k + \hat{c}_k p_k + \hat{d}_k q_k, \;\; r_{k+2} = r_k - \hat{c}_k Ap_k - \hat{d}_k Aq_k.$$

7. If $r_{k+2} = 0$ terminate, else compute

$$p_{k+2} = r_{k+2} + \left(\frac{\hat{b}_k}{\Delta_k}\right)(d_k q_k - \delta_k p_k), \;\; \hat{b}_k = -q_k^H A r_{k+2},$$

$$q_{k+2} = Ap_{k+2} + \left(\frac{\hat{\beta}_k}{\Delta_k}\right)(d_k q_k - \delta_k p_k), \;\; \hat{\beta}_k = -q_k^H AAp_{k+2},$$

8. Increase the index k by 2 and go to equation 1 step 2.

9. In matrix iterations, the algorithm should work in order(G) iterations unless significant roundoff errors occur. Successive estimates x_{n+1}, x_{n+2} should be checked for diverging behavior.

Chapter 4

The Annihilation and Suppression of Spectral Terms

4.1 The Classical Shanks-Schmidt Approach and Generalizations.

Suppose that a sequence of complex numbers x_n has a dependence on a complex number x_∞ in the following form:

$$x_n = x_\infty + \sum_{r=1}^{\infty} c_r \lambda_r^n \tag{4.1}$$

where $\lambda_r (\neq 1)$ and c_r are complex numbers. Suppose further that $|\lambda_{k+1}| < |\lambda_k|$ for some k and $|\lambda_r| \ll 1$ for all $r > k > 0$. For large enough n consider the truncation

$$x_n \approx x_\infty + \sum_{r=1}^{k} c_r \lambda_r^n. \tag{4.2}$$

Form a linear system of $k + 1$ successive iterates

$$\begin{aligned} x_n &\approx x_\infty + \sum_{r=1}^{k} c_r \lambda_r^n \\ x_{n+1} &\approx x_\infty + \sum_{r=1}^{k} c_r \lambda_r^{n+1} \\ &\vdots \\ x_{n+k} &\approx x_\infty + \sum_{r=1}^{k} c_r \lambda_r^{n+k}. \end{aligned} \tag{4.3}$$

In matrix form the system appears as $\Lambda c \approx s$ where

$$\Lambda = \begin{bmatrix} 1 & \lambda_1^n & \lambda_2^n & \cdots & \lambda_k^n \\ 1 & \lambda_1^{n+1} & \lambda_2^{n+1} & \cdots & \lambda_k^{n+1} \\ & & \vdots & & \\ 1 & \lambda_1^{n+k} & \lambda_2^{n+k} & \cdots & \lambda_k^{n+k} \end{bmatrix},$$

$$c = \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} \quad \text{and} \quad s = \begin{bmatrix} x_n \\ x_{n+1} \\ \vdots \\ x_{n+k} \end{bmatrix} \tag{4.4}$$

Use Cramer's rule to solve for the first unknown in s, so that

$$x_\infty \approx \frac{\det \begin{bmatrix} x_n & \lambda_1^n & \cdots & \lambda_k^n \\ x_{n+1} & \lambda_1^{n+1} & \cdots & \lambda_k^{n+1} \\ & \vdots & & \\ x_{n+k} & \lambda_1^{n+k} & \cdots & \lambda_k^{n+k} \end{bmatrix}}{\det \begin{bmatrix} 1 & \lambda_1^n & \cdots & \lambda_k^n \\ 1 & \lambda_1^{n+1} & \cdots & \lambda_k^{n+1} \\ & \vdots & & \\ 1 & \lambda_1^{n+k} & \cdots & \lambda_k^{n+k} \end{bmatrix}}. \tag{4.5}$$

If the expression (4.3) were exact, the extrapolation (4.5) would be exact. the accuracy of the approximation in (4.3) depends on the sensitivity of the linear system $\Lambda c = s$ to a perturbation of the terms of an amount ϵf where $\epsilon > 0$ and $f \epsilon C^n$. It is well known (see Golub and VanLoan for example) that the solutions to $\Lambda c_E = s + \epsilon f$ and $\Lambda c = s$ have a relative error bound given by

$$\frac{\|c_E - c\|}{\|c\|} \leq \epsilon K(\Lambda) \frac{\|f\|}{\|s\|} \tag{4.6}$$

where $\|\cdot\|$ is any vector norm and $K(\Lambda) = \|A\| \cdot \|A^{-1}\|$ is the condition number of A in the given norm. So the efficacy of the transformation is subject to the size of terms neglected (the perturbation ϵf), the conditioning of Λ, and the nature of the segment of the sequence s.

Suppose for now that the sequence, conditioning and perturbation are such that (4.5) will given an extrapolation of a desired accuracy. The calculation of the determinants in (4.5) is now the only obstacle left in the way of the extrapolation.

The determinant in the denominator of (4.5) can be evaluated in $O(k^2)$ flops since

$$\det \begin{vmatrix} 1 & \lambda_1^n & \cdots & \lambda_k^n \\ 1 & \lambda_1^{n+1} & \cdots & \lambda_k^{n+1} \\ \vdots & & & \\ 1 & \lambda_1^{n+k} & & \lambda_k^{n+k} \end{vmatrix} = \prod \lambda_i^n \det \begin{vmatrix} 1 & 1 & 1 & & 1 \\ 1 & \lambda_1 & \lambda_2 & \cdots & \lambda_k \\ & & \vdots & & \\ 1 & \lambda_1^k & \lambda_2^k & & \lambda_n^k \end{vmatrix}$$

which is effectively the Vandermonde determinant of polynomial interpolation that can be evaluated in $O(k^2)$ flops [28]. For the numerator such efficiency is not obvious; however, the evaluation of (4.5) can sometimes be made efficient with the use of determinental identities (cf. section 4.1.1).

The expression on the right hand side in (4.5) defines a potentially useful sequence transformation even if the extrapolation is not exact. Denote the right hand side of (4.5) by $S_{n,k}(x)$ i.e.

$$S_{n,k}(x) = \frac{\det \begin{vmatrix} x_n & \lambda_1^n & \cdots & \lambda_k^n \\ x_{n+1} & \lambda_1^{n+1} & & \lambda_k^{n+1} \\ & \vdots & & \\ x_{n+k} & \lambda_1^{n+k} & \cdots & \lambda_k^{n+k} \end{vmatrix}}{\det \begin{vmatrix} 1 & \lambda_1^n & \cdots & \lambda_k^n \\ 1 & \lambda_1^{n+1} & \cdots & \lambda_k^{n+1} \\ & \vdots & & \\ 1 & \lambda_1^{n+k} & \cdots & \lambda_k^{n+k} \end{vmatrix}} \qquad (4.7)$$

where $x = x_1, x_2, \ldots$. Note that the length and starting point of the segment $x_n, x_{n+1}, \ldots, x_{n+k}$ of the sequence x is implied by n and k in the transform x.

Let

$$D_{n,k}(x) = \det \begin{vmatrix} x_n & \lambda_1^n & \cdots & \lambda_k^n \\ x_{n+1} & \lambda_1^{n+1} & \cdots & \lambda_k^n + 1 \\ & \vdots & & \\ x_{n+k} & \lambda_1^{n+k} & \cdots & \lambda_k^{n+k} \end{vmatrix} \tag{4.8}$$

Denote the sequence 1, 1, 1, 1, ... by e. From (4.7) it follows that

$$S_{n,k}(x) = \frac{D_{n,k}(x)}{D_{n,k}(e)} \tag{4.9}$$

We will also need an auxiliary matrix defined by equation (4.13).

Except for determinants of very low order, the right hand side of (4.7) is usually evaluated in a recursive manner that has the effect of minimizing storage and computations. A method of forming the recursion will now be described. The idea is to use certain minors of the determinants in (4.7), which describe sequence transforms in themselves, in forming the recursion. Here we will use *Sylvester's determinental identity* which arises out of the Gaußian elimination process. These are not the only possible identities, for example certain Schweinsian identities might do [1], but they will be sufficient for our purposes.

4.1.1 Sylvester's Identity

Submatrices. Let $A \in \mathcal{M}_{m,n}(C)$. For index sets $\alpha \subseteq \{1, 2, \ldots, m\}$ and $\beta \subseteq \{1, 2, \ldots, n\}$, the submatrix that consists of the rows of A indexed by α and the columns indexed by β will be denoted $A(\alpha, \beta)$. If $\alpha = \beta$, we use the notation $A(\alpha) = A(\alpha, \alpha)$.

Let $\alpha \subseteq \{1, 2, \ldots, n\}$ be a fixed index set and let $B = [b_{ij}] \in \mathcal{M}_{n-k}(C)$ be defined by

$$b_{ij} = \det\ A(\alpha \cup \{i\}, \alpha\{j\})$$

where k is the cardinality of α, $i, j \in \{1, 2, \ldots, n\}$ are indices *not* contained in α, and $A \in \mathcal{M}_n(C)$. Sylvester's determinantal identity is

$$\det B = [\det A(\alpha)]^{n-k-1} \det A. \tag{4.10}$$

A popular special case of (4.10) results from letting α be an index set of order $n-2$, say $\alpha = \{2,3,\ldots,n-1\}$. We then have

$$b_{11}b_{22} - b_{21}b_{12} = \det A(\alpha)\det A. \tag{4.11}$$

We generally want $\det A$, so we use

$$\det A = \frac{b_{11}b_{22} - b_{21}b_{12}}{\det A(\alpha)}. \tag{4.12}$$

This particular identity has the special important feature of using contiguous elements in the matrix A in forming minors. These minors will turn out to be "lower" order transforms of particular segments. We now give a simplified proof of the *scalar B.H. Protocall* apparently discovered independently by Brezinski and Håive. [34]

To develop a recursion for (4.7) we need a further auxillary determinant. In order to show the generality of this recursion, let $f_r(n) = \lambda_r^n$ and let f_{kr}^n be defined by

$$f_{kr}^n = \frac{\det\begin{bmatrix} f_r(n) & f_r(n+1) & \ldots & f_r(n+k) \\ f_1(n) & f_1(n+1) & \ldots & f_1(n+k) \\ \vdots & \vdots & & \vdots \\ f_k(n) & f_k(n+1) & \ldots & f_k(n+k) \end{bmatrix}}{\det\begin{bmatrix} 1 & 1 & \ldots & 1 \\ f_1(n) & f_1(n+1) & \ldots & f_1(n+k) \\ \vdots & \vdots & & \vdots \\ f_k(n) & f_k(n+1) & \ldots & f_k(n+k) \end{bmatrix}} \tag{4.13}$$

which is effectively $S_{n,k}(f_r(n), f_r(n+1), \ldots, f_r(n+k), \ldots)$ Now given a fraction r/s let $\mathcal{N}(r/s) = r$. Apply Sylvester's identity to the numerator and denominator of (4.7).

$$S_{n,k}(x) = \frac{D_{n,k-1}(x)\mathcal{N}(f_{k-1,k}^{n+1}) - D_{n+1,k-1}(x)\mathcal{N}(f_{k-1,k}^{n})}{D_{n,k-1}(e) - \mathcal{N}(f_{k-1,k}^{n+1}) - D_{n+1,k-1}(e)\mathcal{N}(f_{k-1,k}^{n})} \tag{4.14}$$

Divide the numerator and denominator of the above fraction by $D_{n,k-1}(e)D_{n+1,k-1}(e)$ to obtain

$$\begin{aligned} S_{n,k}(x) &= \frac{S_{n,k-1}(x)f^{n+1}_{k-1,k} - S_{k-1,n+1}(x)f^{n}_{k-1,k}}{f^{n+1}_{k-1,k} - f^{n}_{k-1,k}}, \qquad (4.15) \\ \text{where } n,k &> 0 \\ & \qquad (4.16) \end{aligned}$$

with initial condition

$$S_{n,0}(x) = x_n \qquad (4.17)$$

Identically, the following recursion may be formed for the quantities $f^n_{k,r}$:

$$\begin{aligned} f^n_{k,r} &= \frac{f^n_{k-1,r}f^{n+1}_{k-1,k} - f^{n+1}_{k-1,i}f^{n}_{k-1,k}}{f^{n+1}_{k-1,k} - f^{n}_{k-1,k}}, \qquad (4.18) \\ f^n_{0,r} &= \lambda^n_r, \\ r &\geq 1, \\ 0 \leq k &\leq r-2. \end{aligned}$$

The recursion (4.15) and (4.16) is directly generalizable to extrapolation on sequences of the form

$$x_n = x_\infty + \sum_{r=1}^{\infty} c_r f_r(n) \qquad (4.19)$$

where $\{f_r(n)\}$ is now any asymptotic scale [34], not just an exponential and the c_r are vectors. Wimp's choice in [34] of an asymptotic scale seems the proper generalization of (4.1).

The appearance of the recurrence (4.15) should not be misconstrued as as a highly efficient means of evaluating $S_{n,k}$. Its chief advantage is using previous iterates, so that convergence of in between iterates may be checked in the process of proceeding from S_{n_0,k_0} to $S_{n,k}$ where $n > n_0$ and $k > k_0$. In fact, it is not emphasized enough in scalar sequence transforms that the extrapolation is a single component solution to a linear system of the form

$$\begin{bmatrix} 1 & f_1(n) & \dots & f_k(n) \\ 1 & f_1(n+1) & \dots & f_k(n+1) \\ & \vdots & & \\ 1 & f_1(n+k) & \dots & f_k(n+k) \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n+1} \\ \vdots \\ x_{n+k} \end{bmatrix}. \tag{4.20}$$

So any evaluation of the transform should not take many more operations that the forward elimination process of Gaußian Elimination, which costs $(k+1)^3/3$ flops for (4.19) above. Also, various symmetries of the systems and subsystems in (4.19) may be exploited with the many elimination processes that exist in the public domain, for example Toeplitz system solvers and Cholesky decompositions, to produce efficient implementations of (4.7).

The original error structure $\Sigma_{i=0}^{k} c_i \lambda_r^n$ has some distinguishing characteristics that should be shown. Consider an interpolation polynomial $p_k(z) = \sum_{i=0}^{k} a_i z^i$ of degree k which has the complex numbers $\lambda_1, \lambda_2, ...\lambda_k$ as roots and normalized so that $p_k(1) = 1$. Denote by E the forward shift operator $Ex_n = x_{n+1}$ and notice that $\epsilon_{n+\ell+1} = \sum_{r=0}^{\ell} \Delta x_{n+r} + \epsilon_r$ when $\ell > r$. Apply $p_k(E)$ to both sides of $\epsilon_n = \sum_{i=0}^{k} c_i \lambda_i^n$ to obtain

$$p_k(E)\epsilon_n = \sum_{\ell=0}^{k} a_\ell \epsilon_{n+\ell} = \sum_{\ell=0}^{k} a_\ell \left(\sum_{r=0}^{\ell} \Delta x_{n+r} - \epsilon_n \right) = 0. \tag{4.21}$$

Hence, upon changing the order of summation and redefining the coefficients, we have the following error structure

$$x_n = x_\infty + \sum_{r=0}^{k} d_r \Delta x_{n+r} \tag{4.22}$$

Moreover, p_k annihilates ϵ_r for $r > n$ as can be seen by a suitable redefining of the coefficients c_r in for values of $n > n_0$. Hence the system (4.3) is equivalent to

$$x_m = x_\infty + \sum_{r=0}^{k} d_r \Delta x_{m+r} \qquad n \leq m \leq n+k+1$$

which has solution

$$x_\infty = \frac{\det\begin{vmatrix} x_n & \Delta x_n & \cdots & \Delta x_{n+k} \\ x_{n+1} & \Delta x_{n+1} & \cdots & \Delta x_{n+k+1} \\ & \vdots & & \\ x_{n+k+1} & \Delta x_{n+k+1} & \cdots & \Delta x_{n-2k+1} \end{vmatrix}}{\det\begin{vmatrix} 1 & \Delta x_n & \cdots & \Delta x_{n+k} \\ 1 & \Delta x_{n+1} & \cdots & \Delta x_{n+k+1} \\ \vdots & & & \\ 1 & \Delta x_{n+k+1} & \cdots & \Delta x_{n+2k+1} \end{vmatrix}} \tag{4.23}$$

This is the classical <u>Shank's</u> <u>transform</u> (see Shanks). It has the characteristic that it spectral terms $\lambda_1, \ldots, \lambda_n$ are a simple byproduct of the iteration and need not be postulated.

There are other recursions that evaluate the transform (4.22) expressed in the ϵ- algorithm of Wynn [36]. The procedure is roughly the same as before–determinental identities lead to a recursion. The relevant identities, however, are an obscure two of the Schweinsian type and the procedure is somewhat indirect [34]. The algorithm has the following appearance:

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n-1)} + \frac{1}{\epsilon_l^{(n+1)} - \epsilon_k^{(n)}} \qquad \begin{matrix} n > 0 \\ k \geq 0' \end{matrix} \tag{4.24}$$

with initial conditions

$$\begin{aligned} \epsilon_{-1}^{(n)} &= 0 \\ \epsilon_0^{(n)} &= x_n \qquad n \geq 1. \end{aligned} \tag{4.25}$$

A result of Wynn [36] states that

$$\epsilon_{2k}^{(n)} = S_{n,k}(x) \tag{4.26}$$

and

$$\epsilon_{2k+1}^{(n)} = S_{n,k}(\Delta x)$$

where $\Delta x = \{\Delta x_1, \Delta x_2, \cdots .\}$

The ϵ-algorithm is but one of many ways to evaluate the transform (4.21). However, there is understandably no recursion as simple to evaluate the the much more general B.H. protocall. This is to be expected since the linear systems produced do not, in general, consist of nice matrices with Toeplitz submatrices. In fact, the use of Sylvester's identity is convenient, but probably not as efficient or stable as a good Gaußian elimination routine which takes advantage of peculiarities of the system or some large subsystem formed to determine the extrapolation.

Consider a system, such as (4.19), formed in the course of an extrapolation:

$$\begin{bmatrix} 1 & \\ \vdots & \mathbf{M}_k \\ 1 & \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n+1} \\ \vdots \\ x_{n+k} \end{bmatrix} \tag{4.27}$$

$\mathbf{M}_k$ is a $(k+1) \times k$ matrix. By permuting the first column to the last, we can solve for x_∞ with the forward stage of elimination with one back substitution. The question here is the structure of $\mathbf{M}_k$. For example, if it is symmetric, a Cholesky method could be used; if Hankel, a Trench or Durbin algorithm would be appropriate [14]. If it is desired to compute an a higher order transform by enlarging the system with an additional row, the new system may be solved efficiently using the pivots from the previous lower order system. So there is usually at most an addition of $\mathcal{O}((k+1)^2)$ operations. If (4.27) has a subsystem consisting of a particular matrices used in previous extrapolations, efficient extrapolations may be implemented with a *Sherman-Woodbury-Morrison formula* [14]. This is discussed in section 4.3.

It is important that the transforms be numerically stable; that is, not unduly sensitive to the natural numerical errors in the sequence x that result from floating point arithmetic. The epsilon algorithm has been shown to be stable for schemes with error structures of the form

$$\epsilon_n = \sum_{i=1}^{\infty} c_i \lambda_i^n$$

where $\frac{1}{2} > |\lambda_1| > |\lambda_2| \cdots$ [34].

4.2 Eigenvalue Problems, Analytic Continuation, and Rational Approximation.

In this section we investigate the use of *rational approximations* for forming approximations. A nice motivation for the use of a certain types of rational approximations, the *Padé-type approximations*, is found in the problem of finding an analytic continuation to a type of power series evaluated at ∞. To be more specific, let $x_n(z) = \sum_{i=0}^{n} \frac{c_i}{z_{i+1}}$ be a power series converging on some ball $\mathcal{B}$ about ∞ to $x(z) = \sum_{i=0}^{\infty} \frac{c_i}{z^{i+1}}$ where $c_i = vA^i w$ for some matrix A and fixed vectors v and w and none of the eigenvalues of A are in $\mathcal{B}$. Then on $\mathcal{B}$ we have that

$$\begin{aligned} x(z) &= \left(v \sum_{i=0}^{\infty} \frac{A^i}{z^{i+1}} \right) w \\ &= v \frac{1}{z} \frac{1}{1 - \frac{A}{z}} w \\ &= v(z - A)^{-1} w. \end{aligned}$$

so that the problem of finding the poles of $x(z)$ from $x_n(z)$ is equivalent to finding the eigenvalues of A from the coefficients, or "moments", c_i, $i \leq n$. More importantly from the extrapolation standpoint, we consider a scalar sequence element x_n to be the partial sum $x_n(1)$ and look for an analytic continuation $x_\infty = x(1)$ to $x(z)$ at $z = \alpha$ if $\Delta x_i(\alpha)$ is chosen to be c_i/α^{i+1}. Generally, α is picked as 1. We expect, roughly, that if there are eigenvalues of A near α, that some sort of analytic continuation will be necessary to produce a more accurate approximation to the limit point of the sequence x_n than the series.

Now consider the matrix iteration $x_{i+1} = Ax_i + w = \sum_{k=0}^{i} A^k w$ where $x_0 = 0$ and the spectrum of the square matrix A is within or on the unit ball $\mathcal{B}(1,0)$. Then a termwise extrapolation can be formed from analytic continuations to $e_l^T, x_{i+1} = \sum_{k=0}^{i} e_l^T A^k w$ for $e_l = (0,0,\ldots,1,0,\ldots,0)$ where the 0 is in the lth spot. Here we will discuss the use of rational approximations as analytic continuations.

4.2.1 Rational and Polynomial Approximation

Polynomial approximations are generally used for *interpolation*, but are generally poor choices for *extrapolation* because of their growth outside a sufficiently large interval. In terms of uses for transformations then, it might be suspected then that unless a nonmonotone transformation of the indices is made, polynomial approximations are poor choices for monotone sequences. However, when viewing the sequence transform problem as an error minimization problem over a compact region of the plane (which happens with some nonmonotone and specific monotonic sequences), polynomial approximations often produce very effective transforms [18].

In contrast to polynomial approximation, rational approximations of functions on $[-\infty,\infty]$ are useful for extrapolation outside a given closed interval $[a,b]$ of consideration, especially when the function under consideration has finite asymptotic behaviour. Rational approximation is a natural choice for sequence transformations if one considers a segment $x_n, x_{n+1}, \cdots x_{n+k}$ of a sequence x_n as being values of a function where the index i is an ordinate value corresponding to abscissi x_i and the limit point x as the value of the function at infinity. Extrapolations may also be formed by considering the sequence as a sequence of partial sums of a power series and using rational approximations as *analytic continuations* to the partial sums.

Pade' approximants are formed as a result of a special summability technique for the partial sums of a power series, this approach is due to Wimp [34]. Let

$x_n(z)$ be the partial sums of a power series analytic at 0 converging to $x(z)$ within an appropiate radius and

$$\begin{aligned} x(z) &= \sum_{k=0}^{\infty} a_k z^k && (4.28) \\ x_n(z) &= \sum_{k=0}^{n} a_k z^k. && (4.29) \end{aligned}$$

Let γ be complex and σ_{nk} be an infinite lower triangular array of numbers. Define

$$\begin{aligned} A_n(z,\gamma) &= \sum_{k=0}^{n} \gamma^{-k} \sigma_{nk} x_k(z), && (4.30) \\ B_n(z,\gamma) &= \sum_{k=0}^{n} \gamma^{-k} \sigma_{nk}, && (4.31) \\ E_n(z,\gamma) &= \sum_{k=0}^{n} \gamma^{-k} \sigma_{nk} e_k(z) && (4.32) \end{aligned}$$

where $e_k(z) = x_n(z) - x(z)$. We have then that

$$x(z) \quad = \quad \bar{x}_n(z,\gamma) - \bar{e}_n(z,\gamma) \qquad (4.33)$$

where

$$\begin{aligned} \bar{x}_n(z,\gamma) &= \frac{A_n(z,\gamma)}{B_n(\gamma)} && (4.34) \\ \bar{r}_n(z,\gamma) &= \frac{E_n(z,\gamma)}{B_n(\gamma)}. && (4.35) \end{aligned}$$

For γ fixed, $\bar{x_n}$ is just a weighted mean of $x_0, \cdots, x_n$ with weights

$$\mu_{nk} = \frac{\gamma \sigma_{nk}}{\sum_{k=0}^{n} \gamma^{-k} \sigma_{nk}}. \qquad (4.36)$$

Let $\gamma = z$ and $\bar{x}_n(z,z) = \bar{x}(z)$, from (4.31), (4.34) and (4.36) we have

$$\begin{aligned} A_n(z,z) &= A_n(z), && (4.37) \\ \bar{r}(z,z) &= \bar{z}, && (4.38) \\ \bar{s}(z) &= \frac{A_n(z)}{B_n(z)} = \frac{z^n A_n(z)}{z^n B_n(z)} && (4.39) \end{aligned}$$

and because the latter is a ratio of two polynomials, it is a rational approximation. To show how good the rational approximation is, consider

$$D_n(z) = [z^n B_n(z)]\, x(z) - [z^n A_n(z)]\,. \tag{4.40}$$

Careful consideration of the sums on the left and right reveals that the rational approximation agrees with the power series through $n+1$ terms, i.e. $D_n(z) = \mathcal{O}(z^n)$. There is obviously a great deal of leeway in the choice of the weights. In fact, they can be chosen to agree up to $\mathcal{O}(z^{2n+1})$ in which case the approximation is a *diagonal Padé approximant* (see section 4.2.2). In general, the rational approximant (4.40) is known as a *Padé-type Approximant.*

The next concern is the choice of the weights σ_{ij}. We will let these weights be chosen automatically through the use of *Padé Approximation*, which is in essense the formulation of rational approximations of given degree in the numerators and denominators which, when expanded in their *own* Taylor series, agree with the given Taylor series to as many terms as possible. Traditionally with matrix iterations, choices of the weights have been based on the roots or coefficients of polynomials *orthogonal* over a given region containing the spectrum of the iterative matrix. We choose here *not* to have the advantage of spectral estimates.

4.2.2 Padé Approximation

Let f be a formal power series at $x = 0$ and

$$f(z) = \sum_{i=0}^{m+n} c_k x^k + \mathcal{O}(x^{m+n+1}). \tag{4.41}$$

We look for a rational function p/q, $p \in \mathcal{P}_m$, $q \in \mathcal{P}_n$ such that

$$f(z) - \frac{p(z)}{q(z)} = \mathcal{O}(z^{m+n+1}). \tag{4.42}$$

or, if $q \neq 0$

$$q(x)f(x) - p(x) = \mathcal{O}(z^{m+n+1}). \tag{4.43}$$

Putting $p(z) = \sum_{i=0}^{m} a_i z^i$ and $q(z) = \sum_{i=0}^{n} b_i z^i$, one obtains from (4.43):

$$\left(\sum_{k=0}^{n+m} c_k x^k\right)\left(\sum_{j=0}^{n} b_j x^j\right) = \sum_{i=0}^{n} a_i x^i + O(x^{m+n+1}). \tag{4.44}$$

This leads to the linear system

$$\begin{aligned} \sum_{j=0}^{n} c_{m-n+i+j} b_{n-j} &= 0, \qquad i = 1,2,\ldots,n, && (4.45)\\ \sum_{j=0}^{\min\{i,n\}} c_{i-j} b_j - a_i &= 0, \qquad i = 0,1,\ldots,m, && (4.46) \end{aligned}$$

with the understanding that $c_k = 0$ for $k < 0$. This system of homogeneous equations *always* has a nontrivial solution where $q \neq 0$. When solving directly, the b_j's from the denominator can be determined from (4.45), and the a_i's determined subsequently from (4.46). Sometimes, a Padé approximant is *degenerate*; that is, $q(0) = 0$ or degree$(p) \leq m - 1$ or degree$(q) \leq n - 1$. Details of this situation may be found in [17]; we will assume that no such degeneracies occur from extrapolation sequences.

4.2.3 On the Evaluation of Padé Approximants

We are assuming here that the existence and degeneracy of approximants are not in question. This enables us to use two recursive methods for pointwise evaluation of Padé approximants. These methods are the quotient difference algorithm (Q.D. algorithm) and the ϵ-algorithm, which has already been introduced in a different context of evaluating the Shanks- Schmidt transform (equations 4.24 and 4.25).

Q.D. Algorithm. This algorithm develops a continued-fraction representation for a sequence of Padé approximants. It requires that the elements of the corresponding continued fraction be finite and nonzero. The form of the continued fraction related to the original power series is given on the next page by

$$\sum_{i=0}^{\infty} c_i z^i = \cfrac{c_0}{1 - \cfrac{q_1^0 z}{1 - \cfrac{e_1^0 z}{1 - \cfrac{q_2^0 z}{1 - \cfrac{e_2^0 z}{1 - \ldots}}}}}, \tag{4.47}$$

where

$$\begin{aligned} e_0^J &= 0, \qquad J = 1,2,3,\ldots, \\ q_1^J &= \frac{c_{J+1}}{c_J} \qquad J = 0,1,2,\ldots \end{aligned} \tag{4.48}$$

and the elements of (4.47) satisfy, for $J \geq 0$ and $M = 1,2,3,\ldots,$

$$e_M^J \, q_{M+1}^J = e_M^{J+1} \, q_M^{J+1} \tag{4.49}$$

$$q_M^J + e_M^J = e_{M-1}^{J+1} + q_M^{J+1}. \tag{4.50}$$

The elements are normally exhibited in a Q.D. table as shown in figure 4.1 on the next page. The implementation of the Q.D. algorithm is a little more complicated that the ϵ-algorithm because of the appearance of two relations (4.49) and (4.50). Usually successive convergents are desired in extrapolation techniques, so generally a forward recurrence or a particular summation formula is used to evaluate the continued fraction once the coefficients are determined ([17] p.114 ff.).

Considering $f(z) = \sum_{i=0}^{\infty} c_i z^i$ as a meromorphic function, it can be shown [17] that the "q" columns converge to the reciprocals of the poles of $f(z)$, provided that the poles are distinct. Similarly the "e" rows converge to the reciprocals of the zeros of $f(z)$, provided they have distinct moduli. Also, recalling from section 4.2 the equivalence of finding the poles of $f(z)$ from power series coefficients c_i

Figure 4.1: The Quotient Difference Algorithm

The two left columns are specified by (4.48), and the remaining elements are determined by (4.49) and (4.50). The elements along *any* diagonal are the elements of the continued fraction (4.47)

$$
\begin{array}{ccccccc}
 & q_1^0 & & & & & \\
e_0^1 & & e_1^0 & & & & \\
 & q_1^1 & & q_2^0 & & & \\
e_0^2 & & e_1^1 & & e_2^0 & & \\
 & q_1^2 & & q_2^1 & & q_3^0 & \\
e_0^3 & & e_1^2 & & e_2^1 & & \ddots \\
 & q_1^3 & & q_2^2 & & \vdots & \\
e_0^4 & & e_1^3 & & \vdots & & \\
 & q_1^4 & & \vdots & & & \\
e_0^5 & & \vdots & & & & \\
\vdots & & & & & &
\end{array}
$$

where $c_i = vA^i w$ for two vectors $v, w$and the eigenvalue problem of a matrix A , we see that "q" columns may be used in an eigenvalue extrapolation of the form (2.19). These facts may be used in forming an extrapolation by using the poles as eigenvalues in an eigenvalue extrapolation.

4.2.4 On Acceleration Methods for Vector Sequences

Many problems of physics and engineering are solved with vector iterations. This especially applies to problems of nonlinear ordinary and partial differential equations. The computer codes that arise in these applications are often large and complex. One who uses the code for a particular application may find that it converges slowly or diverges. Appropriate modifiction of the code may be a task for an expert and completely impractical for a user who needs the answers quickly and chose the code as a matter of convenience. In this situation a vector sequence transformation may be useful. They are usually quite easy to incorporate. All that is needed in general is some extra storage, a call to a subroutine and the vector acceleration subroutine itself.

As mentioned before, the simplest vector sequence transformation is to apply a scalar sequence transformation componentwise to the vector. Despite their simplicity, these methods may be quite effective. However, certain singularity problems in the scalar sequence transform may be magnified with application in the vector case because of the large number of times it must be used. As an example, consider the ϵ-algorithm (4.23). The recursion may be represented as

$$\begin{tikzcd}[row sep=small, column sep=small]
x_0 \arrow[dr] & & & \\
0 \arrow[r] & \epsilon_1^{(0)} \arrow[dr] & & \\
x_1 \arrow[ur, no head] \arrow[rr] \arrow[dr] & & \epsilon_2^{(0)} & \\
0 \arrow[r] & \epsilon_1^{(1)} \arrow[ur, no head] & & \epsilon_3^{(0)} \\
x_2 \arrow[ur, no head] & & \epsilon_2^{(1)} & \vdots \\
0 & \epsilon_1^{(2)} & \vdots & \\
x_3 & \vdots & & \\
\vdots & \vdots & &
\end{tikzcd} \tag{4.51}$$

From (4.23) it is seen that numerical overflows will result if $|\epsilon_k^{(n+1)} - \epsilon_k^{(n)}|$ becomes sufficiently small for some n and from (4.27) that the scheme may be unstable if the Shank's transform $S_{n,k}(\cdot)$ extrapolates (correctly) the sequence Δx_i before extrapolating x_i. This may render the ϵ-algorithm unsuitable for a large number of componentwise extrapolations.

In response to this, two notable generalization of scalar sequence transforms suitable for vector use will now be discussed. One is the direct "vector" generalization of the scalar transform

$$E_{n,k}(s) = \frac{\det \begin{vmatrix} x_n & f_1(n) & \cdots & f_k(n) \\ x_{n+1} & f_1(n+1) & \cdots & f_k(n+1) \\ \vdots & & & \\ x_{n+k} & f_1(n+k) & \cdots & f_k(n+k) \end{vmatrix}}{\det \begin{vmatrix} 1 & f_1(n) & \cdots & f_k(n) \\ 1 & f_1(n+1) & \cdots & f_k(n+1) \\ \vdots & & & \\ 1 & f_1(n+k) & \cdots & f_k(n+k) \end{vmatrix}} \tag{4.52}$$

discussed earlier. The other is a generalization taken directly from a recursion relation, such as the algorithm of Brezinski and Håvie or the ϵ-algorithm of

Wynn.

The idea of the latter can be expressed through example. In the recursion (4.23), the quantities $\epsilon_k^{(n)}$ are formally replaced by vector quantities. The move from scalar addition to vector addition is naturally defined, but a problem exists as to the meaning of

$$\frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}}. \tag{4.53}$$

There are many possibilities for a vector "inverse". Wynn [37] was the first to try Samuelson's inverse of a vector v, which is

$$v^+ = \frac{\overline{v}}{v \cdot v} \tag{4.54}$$

One recognizes this as the Moore Penrose generalized inverse of the "matrix" v, that is, the unique vector of minimal ℓ_2 norm such that $v \cdot v^+ = v^+ \cdot v = 1$. However, some remarkable exactness results given in the following theorem for the Samuelson's inverse version were discovered and later proven by McLeod [23] and Graves-Morris [17].

Theorem 15 (Graves-Morris [17]) *Suppose that a vector sequence x_n converging to x_∞ satisfies a relation of the form*

$$\sum_{i=0} c_r x_{n+r} = x_\infty, \tag{4.55}$$

where $\sum_{i=1}^{k} c_r = 1$ and k is a positive integer. Then the vector ϵ-algorithm

$$\begin{aligned} \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + \left(\epsilon_k^{(n+1)} - \epsilon_k^{(n)}\right)^+, \\ \epsilon_{-1}^{(n)} &= 0, \\ \epsilon_0^{(n)} &= x_n. \end{aligned} \tag{4.56}$$

is exact in the 2k-th column.

The vector epsilon algorithm is exact for matrix iterations $x_{1+1} = Ax_i + b$, where A is square, since they satisfy a relationship of the form (4.47). In this case k is likely to be the order of the matrix A, unless we are lucky enough to guess a starting vector purely in an invariant subspace. Theorem 1 is of little practical value where computation to the $2k$th column, k = order A, would amount to a very expensive numerical method. However, it will be shown in the numerical results of chapter 5 that the vector ϵ-algorithm is a good convergence acceleration technique in variety of situations, especially if, when taking dot products, the sparsity of the vectors is taken in consideration. There is certainly a vector analog of the recursions (4.15) through (4.17) or any other scalar recursion relationship that involves a rational function of iterates. The problem, as can be deduced from the extensive papers of Graves-Morris [17] and McLeod [23] is proving anything about them.

A natural generalization of (4.43) is due to Brezinski and Håive [19]. Suppose that $\mathcal{B}$ is a nontrivial real or complex Banach space with norm $\|\cdot\|$ with dual $\mathcal{B}'$. Pick a convergent sequence of functionals $\phi_1, \phi_2, \cdots$ in $\mathcal{B}'$, where $\ker(\phi_r) \neq \mathcal{B}$, for $r = 1, \infty$. Suppose that sequence $x_1, x_2, \cdots$ in $\mathcal{B}$ has the following representation

$$x_n = x_\infty + \sum_{r=0}^{\infty} c_r f_r(n) \tag{4.57}$$

where the c_r are scalars and for each r, the "spectral" sequence $f_r(1), f_r(2), \ldots$ is convergent in $\mathcal{B}$. Assume that $\|f_r\| = \mathcal{O}(1)$ as $r \to \infty$ and, without loss of generality, that $f_j \neq f_k$ for $j \neq k$. The generalization is given by

$$E_k(x_n) = \frac{\begin{vmatrix} x_n & 0 & f_1(n) & \cdots & f_m(n) \\ \phi_n(x_n) & 1 & \phi_n(f_1(n)) & \cdots & \phi_n(f_k(n)) \\ & & \vdots & & \\ \phi_{n+k}(x_{n+k}) & 1 & \phi_{n+k}(f_n(n)) & & \phi_{n+k}(f_k(n)) \end{vmatrix}}{\begin{vmatrix} 1 & \phi_n(f_n(n)) & \cdots & \phi_n(f_k(n)) \\ 1 & \phi_{n+1}(f_n(n)) & \cdots & \phi_n(f_k(n)) \\ & \vdots & & \\ 1 & \phi_{n+k}(f_1(n+k)) & \cdots & \phi_{n+k}(f_k(n+k)) \end{vmatrix}} \tag{4.58}$$

where the expansion is done formally along the first row to make the determinant in the numerator well defined and the denominator is assumed non-zero.

If the Banach space $\mathcal{B}$ is actually $\mathcal{C}$ and the sequence of functionals ϕ is $\{1,1,\cdots,1,\cdots\}$, then (4.50) is identical to (4.44) [34]. The sequence transform (4.50) is the most general one to date for which nice regularity (convergence of sequence implies convergence of transform) results have been proven for different choices of the functional sequence ϕ and which reduces in the scalar case to the "well known" sequence transforms. The essential tool in the regularity results is Sylvester's identity for reducing the determinants involved into similar ones of lower order.

Some regularity results in the theory of the general B.H. protocall (4.50) will now be given. Assume that $\lim_{k\to\infty} \phi_n = \phi \neq 0$ and that the kernal of ϕ is not the whole space B. Consider a path P through the table of transforms. Graphically, the path is represented by a line through a table of transforms. For example,

$$\begin{array}{cccc} E_1(x_1) & E_2(x_1) & E_3(x_1) & E_4(x_1) \\ E_1(x_2) & E_2(x_2) & E_3(x_2) & E_4(x_2) \\ E_1(x_3) & E_2(x_3) & E_3(x_3) & E_4(x_3) \\ E_1(x_4) & E_2(x_4) & E_3(x_4) & E_4(x_4) \\ E_1(x_5) & E_2(x_5) & E_3(x_5) & E_4(x_5) \end{array} \tag{4.59}$$

To state the results we introduce the definitions:

$$\begin{aligned} \mathcal{K} &= \{x \in \mathcal{B} | \phi(x) = 0\} \\ d(b, \mathcal{K}) &= \inf \lim_{t \epsilon \mathcal{K}} \|b - t\| \\ S_n^k &= \phi_n(E_k(x_n)) \\ S &= \phi(x_\infty) \\ \text{and } \epsilon_n^k &= E_k(x_n) - x_\infty. \end{aligned}$$

Consider the two cases

i. $\phi_1, \phi_2, \ldots$ a constant sequence, P an arbitrary path.

ii. $\phi_1, \phi_2, \ldots$ a general sequence and P an arbitrary path through (4.51) with $n \to \infty$.

Theorem 16 *In cases i. and ii. above let $d(r_n^{(k)}, \mathcal{K})/\|r_n^{(k)}\| \geq \delta > 0$ on P where $n + k$ is sufficiently large, then $E_k(x_n)$ converges to x_∞ on P if an only if $S_n^k(x)$ converges to S along the path P.*

For the convergent sequence $x_n \to x_\infty$, Brezinski [8] has shown that the boundedness away from 1 of $\dfrac{\phi_{n+1}(E_k(f_{k+1}(n+1)))}{\phi_n(E_k(f_{k+1}(n)))}$ for paths where $1 \leq k \leq K$, K a fixed constant, is enough to insure convergence of $E_n(x_k)$ to x_∞ on paths within $0 \leq k \leq K + 1$.

The behaviour of $D_k(x_n)$ for unbounded k is apparently unexplored. Results for classes of sequences for which $\frac{\|E_k(x_n) - x_\infty\|}{\|x_n - x_\infty\|} = O(1)$, i.e. the transform is accelerative, are also unknown. There are some results for acceleration in a seminorm $(\phi(\cdot))$ for case i. For details see [34] chapter 10.

4.3 On the Formation of Extrapolations

It is not emphasized in the literature that many scalar extrapolation methods are single component solutions to a system of the form

$$\begin{bmatrix} 1 & \\ \vdots & \mathrm{M} \\ 1 & \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n+1} \\ \vdots \\ x_{n+k} \end{bmatrix} \tag{4.60}$$

M is a $(k+1) \times k$ matrix. There seems to be a notion that because determinental methods determine x_∞ without determining $c_1, c_2, \ldots, c_k$, that they are somehow more efficient. However, flop counts reveal that this is not so and, in fact, in many situations Gaußian elimination is preferable ([34] p.198). Determination of good parameters $c_1, c_2, \ldots, c_r$ is all important because, in a sense, they describe the number of degrees of freedom that the sequence terms have on spectral terms. Accordingly we propose that "vector" extrapolations involve the "solution" of a formal system

$$\begin{bmatrix} I & f_1(n) & \ldots & f_r(n) \\ \mathrm{V} & & \mathrm{M}_r & \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} x_n \\ \mathrm{W} \end{bmatrix} \tag{4.61}$$

The MPE method is of this form with $f_i(n) = x_n$, $\mathbf{M}_r$ given by

$$\begin{bmatrix} (\overline{\Delta x_1}, \Delta x_1) & \dots & (\overline{\Delta x_1}, \Delta x_1) \\ \vdots & & \vdots \\ (\overline{\Delta x_r}, \Delta x_1) & \dots & (\overline{\Delta x_r}, \Delta x_r) \end{bmatrix},$$

$\mathbf{V}$ is the $n \times r$ zero matrix, and $\mathbf{W}$ is

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

We will consider the simplification

$$\begin{bmatrix} I & f_1(n) & \dots & f_r(n) \\ \phi_1 & & & \\ \vdots & & \mathbf{M}_r & \\ \phi_r & & & \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} x_n \\ \phi_1 \cdot x_{n+1} \\ \vdots \\ \phi_r \cdot x_{n+k} \end{bmatrix} \qquad (4.62)$$

where the first row is vector or scalar quantities; the unknowns and the right hand side are scalars; I is the $n \times n$ identity matrix; and $\mathbf{M}_r$ is a $(r+1) \times r$ matrix somehow related to the first row such as a sequence of functional evaluations of the vector spectral terms $f_r(n)$; and the column $[\phi_1, \phi_2, \dots, \phi_r]$ and $[\psi_1, \psi_2, \dots, \psi_r]$ are given functionals. In (4.62) the multiplication $\phi_k \cdot x_\infty$ $1 \leq k \leq r$ may be specified or determined from $\phi_k \cdot x_n = \phi_k \cdot x_\infty + \sum_{i=0}^{r} c_i \phi_k \cdot f_i(n)$ if the functional is linear and there is confidence in the representation. A typical choice for the quantities ϕ_k and $f_k(n)$ $1 \leq k \leq r$ is a set of vectors $\phi_k = [\phi_k^1, \phi_k^2, \dots, \phi_k^\ell]$ and $f_k(n) = [f_k^1(n), f_k^2(n), \dots, f_k^\ell(n)]^H$ in $\mathcal{R}^\ell$, $\ell = \dim(x_\infty)$. In this case the formal system is a large matrix of order $(n+r) \times (n+r)$, but quite sparse, so advantage can be taken with a proper sparse solver. Also in the essential problem now is the evaluation of the transform implied in (4.61), the matrix $\mathbf{M}_r$ is obtained from a matrix $\mathbf{M}_{r-1}$ by appending a row and a column. When this happens $\mathbf{M}_{r+1}^{-1}$ is easily obtained from $\mathbf{M}_r^{-1}$ with use of a *Sherman-Woodbury-Morrison* type formula [14]. We can "precondition" the system (??) by left multiplication by the matrix

$$\begin{bmatrix} I_n & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \mathbf{M}_{r+1}^{-1} & \\ 0 & & & \end{bmatrix} \tag{4.63}$$

to obtain the system

$$\begin{bmatrix} I_n & \mathrm{f} \\ \mathbf{M}_{r+1}^{-1}\phi & I_{r+1} \end{bmatrix} \begin{bmatrix} x_\infty \\ c_1 \\ \vdots \\ c_{r+1} \end{bmatrix} = \left[\begin{bmatrix} I_n & 0 \\ \mathbf{M}_{r+1}^{-1}\phi & I_{r+1} \end{bmatrix} \begin{bmatrix} x_n \\ \phi_1 x_{n+1} \\ \vdots \\ \phi_r x_{n+r+1} \end{bmatrix} \right] \tag{4.64}$$

where $\mathbf{M}$ is a $(r+2) \times (r+1)$ matrix, I_ℓ is the $\ell \times \ell$ identity matrix, $\phi = [\phi_1, \phi_2, \dots, \phi_{r+1}]^T$ and $\mathrm{f} = [f_1, f_2, \dots f_{r+1}]$. This system is quite sparse and can be solved efficiently. Possble appropriate sparse solvers are the Conjugate Gradient Methods of chapter 3 and, again, the Sherman-Woodbury-Morrison type formulas. The algorithm will be shown through an example in chapter 5.

There are two key steps in the implementation of this extrapolation.

- A good choice of representing functions f must be chosen. This leads to the overdetermined system $x_n = x_\infty + \sum_{i=0}^{r} c_i f_i(n)$ of which $[x_\infty, c_1, \dots, c_{r+1}]$ is a solution.

- The functionals ϕ_k must be chosen to select a good solution from the set of solutions to the overdetermined system. There is an all important question that will, unfortunately, not be answered here. Intuitively, if one is very confident with the adequacy of the representation

$$x_n = x_\infty + \sum_{i=0}^{r} c_i f_i(n), \tag{4.65}$$

then it is probably best to determine $\phi_k \cdot x_\infty$ explicitly from (4.65). If one is not confident, then perhaps it is best to go to the other extreme with a representation (4.61) where $\mathbf{V}$ is zero.

Making the system (4.61) square was done for clarity. It should be noted that it is possible to make (4.61) overdetermined by picking more ϕ terms than f terms. There are many sparse solvers, such as the planar conjugate gradient technique, which efficiently solve such systems. It may also be desired to allow the first row quantities be scalar terms and the column of unknowns $[x_\infty, c_1, \ldots, c_r]$ be vectors. This leads to extrapolations similar to Henrici's [7].

Chapter 5

Numerical Experiments with Selected Extrapolations

Introduction. Here we investigate the various extrapolations formed in chapters 2,3 and 4 on a model problem. The main thrust of the investigation will be on extrapolation to the limit of a vector sequence produced by two well known iterative methods for linear systems. The first of these methods is the successive overrelaxation method (S.O.R.) and the second is the Chebychev semiiterative method (C.S.I.). These methods have parameters which, when modified, cause the eigenelements of the associated iterative matrix to vary over a wide range. There are "optimal" values— values which minimize the spectral radius of the iterative matrix— for the parameters which are known for certain test problems, but not for general iterative matrices. Consequently, the parameters are often not the best possible. It will be shown that in this situation that the extrapolations are economical and simple methods for improving convergence.

In the following section we give a brief introduction to formulation of classes of iterative methods including S.O.R. and C.S.I..

5.1 Preconditioning and the Formulation of Iterative Methods

Consider the numerical solution of a large sparse linear algebraic system

$$Ax = b, \quad x, b \in \mathcal{R}^n \tag{5.1}$$

where A is nonsingular. Very often in applications A is sparse with a band or skyline structure in which there are only a few nonzero entries. Direct methods based upon a factorization $A = LU$, where L and U are upper and lower triangular factors respectively, tend to produce costly fill-in within the band or skyline structure. Furthermore, in direct solution methods round-off and data errors tend to increase faster than the condition numbers (see for instance [3]). In defense of direct methods, this can be partially alleviated by the method of iterative improvement at additional cost (see Rice [28]).

Iterative methods do not suffer from fill-in and, with effective preconditioning or acceleration, algorithms of almost optimal order may be derived. Furthermore, preconditioning and acceleration techniques are often extendible to *nonlinear* iterative numerical methods just as many of the classic linear iterative methods are.

Let C be an approximation of A which is relatively inexpensive to invert. C is often called a preconditioning matrix. A basic iterative method in *defect correction* form is given by

$$C\delta_i = r_i, \quad x_{i+1} = x_i + \delta_i \tag{5.2}$$

where $r_i = b - Ax_i$ is the residual (or defect) at the ith step. A good starting value is $x_0 = C^{-1}b$.

Consider the splitting

$$A = C - R \tag{5.3}$$

where R is known as the defect matrix. Then (5.2) takes the form

$$Cx_{i+1} = Rx_i + b \tag{5.4}$$

which converges if $\rho(C^{-1}R) < 1$, where $\rho(\cdot)$ is the spectral radius. There is extensive theory devoted to regular splittings ($C^{-1} > 0$ and $R > 0$ elementwise) which may be applied at this point. Included in this is the well known S.O.R. (successive overrelaxation method).

The number of iterations to reach a relative error, $\|x - x_n\|/\|x - x_0\| \leq \epsilon$ is given by

$$k \approx \frac{ln(1/\epsilon)}{ln(1/\rho_0)} \tag{5.5}$$

where $\rho_0 = \|C^{-1}R\|$. As noted be Axellson [2], in the case of common discretizations of second order elliptic partial differential equations where C is chosen to be the (block) diagonal part of the discretization matrix, one obtains $\rho_0 = 1/(1+\varsigma)$ for positive ς independent of h. Hence $k = O(h^{-2})$ which is unacceptably large. Three possible ways to improve the efficiency of iterative methods are

1. "accelerating" the iterative method,

2. picking a good preconditioner, and

3. a combination of (1.) and (2.).

Semiiterative Methods. Here we review the formulation of semiiterative methods. For more details see [6] or [32]. Consider first the iterative method modified from (5.2)

$$x_{i+1} = x_i - \tau_i C^{-1} r_i, \quad r_i = Ax_i - b \tag{5.6}$$

for the solution of $Ax = b$. This is the defect-correction form of the iterative method

$$x_{i+1} = (I - \tau_i C^{-1} A)x_i + C^{-1}b \quad i = 0, 1, \ldots \tag{5.7}$$

where $\{\tau_i\}$ is a sequence of iteration parameters. If $\tau_i = \tau, \ \ i = 0, 1, \ldots$, we talk about a stationary iterative method, otherwise the method is nonstationary or *semi-iterative.* Let $e_i = x_\infty - x_i$, the iteration error. Then it follows from (2.1) that $e_{i+1} = (I - \tau_i C^{-1}A)e_i, \ \ i = 0, 1, \ldots$. So $e_i = P_i(c^{-1}A)e_0$ and $r_i = AP_i(C^{-1}A)A^{-1}r_0 = P_i(AC^{-1})r_0$. Here $P_i(\lambda) = \Pi_{\ell=0}^{m}(1 - \tau_\ell \lambda)$ is an $(m+1)$th degree polynomial having zeros at $1/\tau_i$ and satisfying $P_i(0) = 1$.

We want to choose parameters $\{\tau_i\}$ so that $\|e_i\|$ is minimized. However, in general these parameters depend on e_0 which is not known. Instead we choose to minimize $\|e_i\|/\|e_0\|$ for all choices of e_0 by minimizing $\|P_m(C^{-1}A)\|$ in some way.

If the eigenvalues of $C^{-1}A$ are real and positive and positive lower a and upper b bound are known of the spectrum, then we may choose to minimize $\max_{a \le \lambda \le b} |P_m(\lambda)|$ for all P_m such that $P_m \in \mathcal{P}_m$ and $P_m(0) = 1$. The solution to this problem is well known,

$$P_m(\lambda) = \frac{T_m\left(\dfrac{b+a-2\lambda}{b-a}\right)}{T_m\left(\dfrac{b+a}{b-a}\right)},$$

where $T_m(z) = \cos(m \arccos z)$ are the Chebychev polynomials of the first kind. The corresponding values of τ_i satisfy

$$\frac{1}{\tau_i} = \frac{b-a}{2}\cos\theta_i + \frac{b+a}{2}, \ \ \theta_i = \frac{2i-1}{2m}\pi, \ \ i = 1, 2, \ldots, m,$$

which are the zeros of the polynomial. This method is known as the *Chebychev (one step) method.* It can be shown that if $m \geq \frac{1}{2}(b/a)^{1/2}\ln(2/\varepsilon)$ then $\|\epsilon_m\|/\|\epsilon_0\| \leq \varepsilon$ [14].

Disadvantaged to this method are that on needs accurate estimates of a and b for fast convergence and a tendency toward instability if the parameters τ_i are not taken in a certain order [13]. The latter problem may be alleviated by employing the three term recursion for Chebychev polynomials of the first kind over an interval,

$$x_{i+1} = x\alpha_i x_i + (1-\alpha_i)x_{i-1} - \beta_I C^{-1} r_i, \quad i = 1,2,\ldots,$$

where $x_1 = x_0 - \frac{1}{2}\beta_0 C^{-1} r_0$, $\beta_0 = 4/(a+b)$,

$$\alpha_i = \frac{a+b}{2}\beta_i, \quad \beta_i^{-1} = \frac{a+b}{2} - \left(\frac{b-a}{4}\right)^2 \beta_{i-1}, \quad i = 1,2,\ldots.$$

This is the *Chebychev Semiiterative Method.* We do not have to determine the number of steps beforehand and the method is numerically stable for reasonable close estimates [14]. When the eigenvalues are complex with positive real parts and we know an ellipse containing them, the parameters may be chosen similarly. See Young [18] for details. The intention here is to employ various extrapolation techniques and compare their effectiveness with (slightly) inaccurate estimates of the spectral bounds a and b.

5.1.1 On the Formation of Classical Relaxation Methods

Here we formulate classical relaxation methods by the producing rational approximations to the exponential function. This approach has been considered in Varga [32]. The exponential function arises as a solution to a system of linear ordinary differential equations.

Suppose that a solution to the linear system $Ax = b$ is derived. Consider instead formulating the problem as

$$C\frac{dx}{dt} = b - Ax = r(x) \tag{5.8}$$

where C is a preconditioner to A. Consider a general solution to (5.8) with initial condition $x(0) = x_0$: The solution is given by

$$x(t) = A^{-1}b + \exp(-tC^{-1}A) \cdot \{x_0 - A^{-1}b\}. \tag{5.9}$$

Here $\exp(A) = I+A+A^2+A^3+\ldots$ [14]. Assume that $\exp(tQ) > 0$ (elementwise postive) for all $t > 0$. Such a matrix Q is known as an *essentially positive matrix.* The following theorems from [32] are necessary to understand the methods we shall derive.

The first theorem is closely related to the Perron-Frobenius Theorem.

Theorem 17 *[32] Let Q be an essentially positive matrix. Then Q has a real eigenvalue $\varsigma(Q)$ such that*

1. *To $\varsigma(Q)$ there corresponds an eigenvector $x > 0$.*

2. *If α is any other eigenvalue of Q, then $\Re(\alpha) < \varsigma(Q)$.*

3. *$\varsigma(Q)$ increases when any element of Q increases.*

[32] The following theorem, which follows from theorem 5.1 [32], shows that $\varsigma(Q)$ dictates the asymptotic behavior of $\|\exp(tQ)\|$ for large t when Q is essentially positive.

Theorem 18 *Let Q be an $n \times n$ essentially positive matrix. If $\varsigma(Q)$ is the eigenvalue of theorem 5.1, then*

$$\|\exp(tQ)\| \sim K\exp(t\varsigma(Q)), \qquad t \to +\infty,$$

where K is a positive constant independent of t.

Consider now the nonhomogeneous ordinary matrix differential equation

$$\frac{dv(t)}{dt} = Qv(t) + r \tag{5.10}$$

where v(0) is a specified initial condition and r is independent of t. If Q is nonsingular, the unique solution of (5.10) satisfying the initial condition is

$$v(t) = -Q^{-1}r + \exp(tQ) \cdot \{v(0) + Q^{-1}r\}, \qquad t \geq 0. \tag{5.11}$$

The proof of the following theorem follows from theorem 5.2. It describes the behavior of solutions to (5.10).

Theorem 19 *[32] Let the $n \times n$ matrix Q of (5.10) be essentially positive and nonsingular. If $\varsigma(Q) > 0$, then for certain initial vectors $v(0)$, the solution $v(t)$ of (5.10) satisfies*

$$\lim_{t \to +\infty} \|v(t)\| = +\infty \tag{5.12}$$

If $\varsigma(Q) < 0$, then the solution vector $v(t)$ is uniformly bounded in norm for all $t \geq 0$, and satisfies

$$\lim_{t \to +\infty} v(t) = -Q^{-1}r. \tag{5.13}$$

Let $Q = C^{-1}A$, it can be shown that if C has a positive diagonal and A is an *irreducible, M-matrix* [32], we have that $\varsigma(Q) < 0$ and (5.13) is satisfied.

Replace t by $t + \Delta t$ in (5.9), then

$$\begin{aligned} x(t + \Delta t) &= A^{-1}b + \exp(-\Delta t C^{-1}A)[x_0 - A^{-1}b] \\ &= A^{-1}b + \exp(-\Delta t C^{-1}A)\epsilon_0 \end{aligned} \tag{5.14}$$

for any $\Delta t > 0$ large or small. Iterative methods can be constructed from (5.10) by considering polynomial, rational, or general matrix function approximations to $\exp(-\Delta t C^{-1}A)$. The approximation is often to the scalar function $\exp(-\gamma)$ upon which the matrix argument $-C^{-1}A\Delta t$ is substituted. Globally accurate approximations over as large a region as possible are needed so that a large time step or large norm preconditioners may be used to diminish $\exp(-\Delta t C^{-1}A)\epsilon_0$. This produces an efficient iterative method. Of course there is always the trade-off that comes with the additional complexity that comes with better approximations.

The familiar point and block Jacobi methods can be constructed from (5.10) by letting C be a diagonal or block diagonal of $A, \Delta t = 1$, and approximating e^{-t} by its first order McClaurin expansion about 0; that is $e^{-t} \approx 1 - t$. Let $t = n\Delta t$.

Using a first order matrix approximation

$$\exp(-C^{-1}A\Delta t) \approx (I - \Delta tL)^{-1}\{\Delta tU + (1 - \Delta t)I\}$$

where

$$\begin{aligned} A &= D - (C_L + C_U) \\ L &= D^{-1}C_L \\ U &= D^{-1}C_U \end{aligned}$$

produces the iteration

$$x((n+1)\Delta t) = (I - \Delta tL)^{-1} \quad \{\Delta tU + (1 - \Delta t)I\}x(n\Delta t) \tag{5.15}$$

$$+\Delta t(I - \Delta tL)^{-1}D^{-1}b \tag{5.16}$$

This is the ***successive overrelaxation method*** (S.O.R.) when C_L and C_U are respectively the lower and upper traingular portions of A. The exponential derivation is due to Varga [32].

With rational matrix approximations, care must be taken to ensure that the inverse exist and is easily found numerically. Direct matrix approximations of

$$\exp(-\Delta tC^{-1}\sum A_i)$$

where $\sum A_i = A$ from a class of iterative methods including S.O.R. and Jacobi.

In problems where effective time accuracy is desired, it is important to consider the accuracy of the approximation, the time step, and the conditioning of the component matrices $C^{-1}A_i$. Details of the accuracy in terms of both the Schur decomposition and the Jordan canonical form of the matrix exponential argument are given in Golub and VanLoan [14].

Now consider the exponential approach in its generality. Let $A = \sum_{\ell=1}^{n} A_\ell$. Then from (5.10) it is seen that

$$x(t + \Delta t) = A^{-1}b + \prod_{\ell=1}^{n} \exp(-C^{-1}A_\ell\Delta t)[x_0 - A^{-1}b]. \tag{5.17}$$

Let $\Delta t = \sum_{q=1}^{r} \Delta t_i$ be a large time step. then

$$x(t+\Delta t) = A^{-1}b + \prod_{\ell=1}^{n} \prod_{q=1}^{r} \exp(-C^{-1}A_\ell \Delta t_q)[x_0 - A^{-1}b] \tag{5.18}$$

or

$$x(t+\Delta t) = \left[I - \prod_{\ell=1}^{n} \prod_{q=1}^{r} \exp(-C^{-1}A_\ell)\Delta t_q \right] A^{-1}b + \prod_{\ell=1}^{n} \prod_{q=1}^{r} \exp(-C^{-1}A_\ell \Delta t_q) x_0 \tag{5.19}$$

As before, an iterative method is constructed by considering rational approximations $(\frac{P}{Q})_{\ell q}(\lambda)$ to $\exp(\lambda)$ where (associated with $\exp(-C^{-1}A_\ell \Delta t_q)$ over a region containing the spectrum of $-C^{-1}A_\ell \Delta t_q$ or by considering direct rational matrix operations

$$\frac{P(-C^{-1}A_\ell \Delta t_q)}{Q(-C^{-1}A_\ell \Delta t_q)}.$$

In practice it is important for P and Q to be constructed such that

$$\frac{P(-C^{-1}A_\ell \Delta t_q)}{Q(-C^{-1}A_\ell \Delta t_q)} A^{-1}b$$

is easily calculated. Since it is A that is inverted, A^{-1} should not appear in the expression.

Let $T(\Delta t) = \prod \prod (\frac{P}{Q})_{eg}(-CA_e \Delta t_q)$ and consider the approximation in (5.19):

$$x(t+\Delta t) = A^{-1}b + T(\Delta t)[x_0 - A^{-1}b] \tag{5.20}$$

Considering this as an iteration and taking m time steps gives

$$x_m = x(t+m\Delta t) = A^{-1}b + T(\Delta t)^m[x_0 - A^{-1}b]. \tag{5.21}$$

It can be seen that if the time steps Δt_ℓ and the splitting is chosen so that the spectral radius $\rho(T(\Delta t)) << 1$, then the method will be quickly convergent.

In summary, the justification for the exponential formulation is (1) in the ease by which classical methods are formed from it, including the classical relaxation method; and (2) the incorporation of readily available accurate rational approximations to e^z; and (3) the splitting property $e^{a+b} = e^a e^b$ which enables us to form iterative methods from preconditioned "parts" of A, as in (5.17).

5.2 More on the S.O.R. Method

The successive overrelaxation method is given by (5.15). We let $\Delta t = \omega$ to formulate the method in defect correction form as

$$x_i = x_{i+1} - \omega(C_L + D)^{-1} \cdot r_i \tag{5.22}$$

where $r_i = b - Ax_i$. For certain physical reasons, the time step is known as the *overrelaxation* parameter. The following theorems are the results of extensive work by Young [18]:

- The method is convergent only for $0 < \omega < 2$. If $0 < \omega < 1$, we speak of underrelaxation.

- The method is faster than the Gauß-Seidel method for a large class of matrices arising from finite differencing and finite element techniques.

- If ρ_J is the spectral radius of the Jacobi iteration and the matrix has property "Y" [6], then the optimal choice of relaxation parameter is given by

$$\omega = \frac{2}{1 + \sqrt{1 - \rho_J^2}}. \tag{5.23}$$

- With optimal choice of parameter ω, the spectral radius for S.O.R. is given by

$$\rho_{\text{S.O.R.}} = \left(\frac{\rho_J}{1 + \sqrt{1 - \rho_J^2}} \right)^2 . \tag{5.24}$$

In two numerical examples, we will assume here that there is no knowledge of the spectrum of the Jacobi matrix of the iteration, so that we do not know the optimal overrelaxation parameter.

We now turn our attention to a boundary value problem exemplified by an *elliptic* partial differential equation. The commonly used *model equation* in the

Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x,y) \tag{5.25}$$

where the source term ρ is given. If the source term is equal to zero, the equation is *Laplace's* equation. Consider a uniform grid spacing of points

$$x_j = x_0 + j\Delta \qquad j = 0, 1, \ldots, J,$$

and

$$y_l = y_0 + l\Delta \qquad l = 0, 1, \ldots, L$$

where Δ is constant. The resulting finite difference approximation using $f(x_j, y_l) = f_{j,l}$ for any function f and

$$\frac{\partial^2 u}{\partial x^2}\Big|_{(x,y)} \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i,j}}{\Delta^2},$$

and similar for y, can be written in the form

$$u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l} = \Delta^2 \cdot \phi_{j,l}. \tag{5.26}$$

(5.26) defines a matrix equation $Ax = b$ where the structure of the matrix is determined by the way the two-dimensional grid is fit into the one dimensional vector of unknowns x. The finite difference equation defines a unique set of eigenvalues and eigenvectors which are the same modulo a similarity transformation of permutation matrices [6].

A general second-order elliptic equation in x and y approximated by finite differences over a square has the form

$$a_{j,l}u_{j+1,l} + b_{j,l}u_{j-1,l} + c_{j,l}u_{j,l+1} + d_{j,l}u_{j,l-1} + e_{j,l}u_{j,l} = f_{j,l}. \tag{5.27}$$

where the model equation has $a = b = c = d = 1$, $e = -4$. The finite difference form of the Jacobi method is obtained by updating the term $u_{j,l}$ in terms of the others:

$$u_{j,l}^{n+1} = \frac{f_{j,l} - a_{j,l}u_{j+1,l}^{n} - b_{j,l}u_{j-1,l}^{n} - c_{j,l}u_{j,l+1}^{n} - d_{j,l}u_{j,l-1}^{n}}{e_{j,l}}. \tag{5.28}$$

Gauß-Seidel results from updating iterates as soon as they become available in the given ordering of unknowns. If the ordering is done along rows, we have

$$u^{n+1}_{j,l} = \frac{a_{j,l}u^{n}_{j+1,l} + b_{j,l}u^{n+1}_{j-1,l} + c_{j,l}u^{n}_{j,l+1} + d_{j,l}u^{n+1}_{j,l-1} - f_{j,l}}{e_{j,l}}. \tag{5.29}$$

S.O.R. is obtained as a linear extrapolation between a Jacobi iterate and a previous iterate:

$$u^{\mathrm{new}}_{j,l} = \omega u^{*}_{j,l} + (1-\omega)u^{\mathrm{old}}_{j,l}. \tag{5.30}$$

where

$$u^{*}_{j,l} = \frac{f_{j,l} - a_{j,l}u^{n}_{j+1,l} - b_{j,l}u^{n}_{j-1,l} - c_{j,l}u^{n}_{j,l+1} - d_{j,l}u^{n}_{j,l-1}}{e_{j,l}}. \tag{5.31}$$

The residual has the form

$$r_{j,l} = f_{j,l} - a_{j,l}u_{j+1,l} - b_{j,l}u_{j-1,l} - c_{j,l}u_{j,l+1} - d_{j,l}u_{j,l-1} - e_{j,l}u_{j,l}. \tag{5.32}$$

so that the S.O.R. algorithm is given by

$$u^{\mathrm{new}}_{j,l} = u^{\mathrm{old}} + \omega\frac{rj,l}{e_{j,l}}. \tag{5.33}$$

In the examples in the following two sections we fix the size of u at 11×11, set $a = b = c = d = 1$, $e = -4$ and $f_{6,6} = 1$ and $f = 0$ elsewhere.

5.3 Extrapolations on the S.O.R. Method

The iterative matrix G in the S.O.R. method is in general nonsymmetric so that

$$x_{i+1} = Gx_i + k \tag{5.34}$$

is an iterative means of solving the nonsymmetric system $(I - G)x = k$. We shall attempt to accelerate the convergence with extrapolation methods that do not assume symmetry. These methods include the M.P.E. extrapolation of section (3.1), an eigenvalue extrapolation from a projected equation (section 2.2) where the eigenvalue solver is for nonsymmetric matrices, and the ϵ-algorithm in its vector version. We start off with the MPE method which, in the numerical examples we have tried and in agreement with Sidi, Smith, and Ford [30] is one of the best. In comparisons with the projected residual and projected eigenvalue methods, we shall show that MPE is the numerically wise way to extend Lusternik's transformation (2.9) for the matrix iterations considered. Accordingly, we devote a large portion of this chapter to this method in trying to determine how to effectively use it.

5.3.1 MPE on the S.O.R. Method

We will now extrapolate on S.O.R. on the model equation with $\omega = 1.0$ and $\omega = 1.2$. The first situation is commonly known as the Gauß-Seidel iteration. In Gauß-Seidel the eigenvalues of the matrix G are real and within the interval $[0,1]$. In fact, the closure of the spectrum nearly covers the whole interval, making Gauß-Seidel a very slowly converging iteration [18]. However, the advantage in this iteration is having a real spectrum which insures the presence of low dimension subspaces approximate to dominant invariant subspaces on which to project a problem. We employ the following notation for the MPE method:

$$\mathrm{MPE}^{a:b}_{x:y:z}. \tag{5.35}$$

Here a is the number of iterates used in the iteration. b means to use iterates $b \cdot i, \quad i = 1, 2, \ldots$, for example, b=2 means to use every other iterate. x is the starting index of the vector iterates (ordered from top to bottom) to use in the extrapolation, y is the finishing index, and z means to use every $z \cdot i, \quad i = 1, 2, \ldots$, indice from x to y. For example, let $x = [x_1, x_2, \cdots, x_{110}]^T$. $x_{10:2:90}$ is the vector $\bar{x} = [x_{10}, x_{12}, \ldots, x_{90}]$. As a final note, we have found that it is generally most effective to employ our extrapolations for a certain length of time to produce an extrapolation x^*, and then *restart* the iteration with $x_0 = x^*$. In the following tables an occurrence where a restart was not employed will be indicated by a rectangle around the number in the table (which is the ℓ_2 norm of the residual). For example, if on the 10th iteration the residual is 0.0123, the entry on the table will appear as $\boxed{0.0123}$ if no replacement was made after the residual of the extrapolation was determined. It will appear as 0.0123 in no replacement was made. Generally, boxes will not appear in the extrapolation columns. After an extrapolation is make, new coefficients are computed for the next extrapolation.

The following page has results of the Gauß-Seidel iteration:

Gauß-Seidel Iteration (S.O.R. with $\omega = 1.0$)

(entries are ℓ_2 norms of the residual)

Iteration	Gauß-Seidel	$\text{MPE}^{5:1}_{20:100:1}$	$\text{MPE}^{10:1}_{30:50:1}$	$\text{MPE}^{20:1}_{20:100:1}$
1	2.0000			
6	$\boxed{0.3474}$	$\boxed{0.2348}$		
11	0.2074	0.1157	0.1434	
17	0.1255	0.0255		
21	0.0760	0.0123	0.1363	0.0713
26	0.0460	0.0035		
31	0.0279		0.0231	
36	0.0169			
41	0.0102			0.0152
46	0.0061			

We now give results for the S.O.R. iteration (5.33) with $\omega = 1.2$. Note here that because of the ineffectiveness of the extrapolations used for the last two columns above we use different MPE iterations (see comments on the next page).

S.O.R. with $\omega = 1.2$

(entries are ℓ_2 norms of the residual)

Iteration	S.O.R. $\omega = 1.2$	$\mathrm{MPE}^{5:1}_{20:100:1}$	$\mathrm{MPE}^{5:1}_{20:100:5}$	$\mathrm{MPE}^{10:1}_{20:100:4}$
1	2000.0			
6	0.3389	0.2869	1.8923	
11	0.1555	0.1189	0.0326	0.1692
16	0.0713	0.0496	0.0135	
21	0.0327	0.0208	0.0056	0.0463
26	0.0150	0.0092		
31	0.0069			
36	0.0034			

Comments. In actual applications the residual calculation *for the extrapolation*, which is one of the more expensive parts of the iteration, would not be used as often as in the table where it is needed for purposes of illustration. For linear iterations such as above, the evidence seems to indicate that it is important to avoid rank deficiency, which is what occurred in columns two and three of the first table above and is seemingly the cause of the failure of the extrapolations. $MPE^{5:1}_{20:100:5}$ accomplished this better than the other versions by keeping the number of vectors in the extrapolation small and the size of the vectors small. As as rough rule for keeping the systems small to avoid rank deficiency is to use about $\sqrt{n}/2$ iterates, where n is the size of the vectors involved in the iteration.

5.3.2 S.O.R. and the ϵ-algorithm

We will now illustrate the dramatic effect the vector ϵ-algorithm has upon the sequences produced from the two variants of S.O.R. used in MPE above. We remind the reader that the meaningful analytic continuations are contained in the even numbered columns of the ϵ-array (4.56). We employ the algorithm in two ways: the first is the diagonal sequence ϵ_k^0 which is generally best to produce analytic continuations of nonrational functions, but expensive memorywise since we must save the whole bottom diagonal of the array; and the second is proceeding down the diagonal until a certain even column is reached, in this case column 5, and then proceeding down the column. The second is more economical storage wise, but slower.

ϵ-algorithm for the Gauß-Seidel Process

(entries are ℓ_2 norms of the residual)

Iteration	Gauß-Seidel	ϵ (diagonal)	ϵ (column 5)
4	0.4510	0.4670	0.4670
6	0.3474	0.4490	0.4490
8	0.2809	0.0719	0.1123
10	0.2294	0.0015	0.0141
12	0.1876	$1.502 \cdot 10^{-5}$	$2.0624 \cdot 10^{-4}$

Now for S.O.R. with some overrelaxation:

ϵ-algorithm for S.O.R. $\omega = 1.2$

(entries are ℓ_2 norms of the residual)

Iteration	S.O.R. $\omega = 1.2$	ϵ (diagonal)	ϵ (column 5)
4	0.4683	0.8154	0.8154
6	0.3389	0.5482	0.5482
8	0.2481	0.0400	0.0722
10	0.1817	$1.5020 \cdot 10^{-5}$	$1.2150 \cdot 10^{-4}$

Remarks. The ϵ-algorithm shows superior convergence to the MPE method of the previous subsection. But at what cost? Note that each use of MPE extrapolation using 5 iterates had a cost of about 2500 flops for the inner products and system inversion. Each extrapolation has a cost of about 700 flops. The cost of the extrapolation is small when is considered that we spend 2600 flops per iteration. The ϵ-algorithm costs about 500 flops in moving one column to the right. There is also a great deal of subscripting and movement of storage not taken into the flop count. Roughly, a move up the diagonal of a 5-column ϵ-algorithm costs the same as an MPE extrapolation. But this cost is incurred for *every iteration* as opposed to every 5 for the MPE method. However, with this version of S.O.R., the convenience and exceptional convergence make the ϵ-algorithm worth it.

We note that the iterates in this example had components that were monotone decreasing to the limit. The epsilon algorithm does not seem to do so well for a highly oscillatory sequence produced by the Chebychev iteration (see section 5.5). Finally we note that none of the extrapolations produced here are likely to work for an optimally overrelaxed scheme. Heuristically, the reason can be seen by consideration of iterative matrix G and the error ε_i at the ith step. The extrapolations produce semiiterative methods which have error of the form

$$\varepsilon_i = p_{i-1}(G)\varepsilon_0$$

where $p_i(G)$ is a polynomial of degree i. The question is the best choice of the polynomials $p_i(z)$. The spectrum of the optimally overrelaxed matrix in the model problem is on a circle centered about zero. So the ideal polynomials will have minimal norm (we choose ℓ_2) over the circle. However, the polynomials are exactly $p_i(z) = z^i$ for $i = 1, 2, \ldots$[12]. This is equivalent to no semiiterative modification of the original iteration $x_{i=1} = Gx_i + k$.

5.4 Experiments on S.O.R. With Other Extrapolations

In this section we consider the formulation of an extrapolation from a projected problem. In particular we use a projected *eigenvalue* problem to form the coefficients of the extrapolation. This is extrapolation of figure 2.1 page 27. We chose not to use some of the specialized projection methods, such as the Lanzcos method for representing the projection of the matrix in tridiagonal form, because of the moderate size of the iteration. Such methods, which save a full Krylov span from a beginning vector, are more suited for larger and sparser iterations [10].

For the model problem we solved the eigenproblem (2.36) with Q taken as the orthonormalization of the matrix $[\Delta x_1, \Delta x_2, \ldots, \Delta x_5]$. Two very simple extrapolations, one using the 10th and 11th iterations and one using the 15th and 16th iterations, with only the dominant eigenvalue were performed. The overhead involved iteration on the columns of Q, five vector inner products, and the solution of a 5×5 eigenproblem. The following convergence history was produced:

Selected Eigenvalue Extrapolation

(entries are ℓ_2 norms of the residual)

Iteration	Gauß-Seidel	Eigenvalue Extrapolation
6	0.3474	
11	0.2074	0.0090
16	0.1255	$6.5571 \cdot 10^{-4}$
21	0.0760	
26	0.0460	

5.5 Some Extrapolations on the Chebychev Method

Here we show that the Chebychev iteration may be improved when the endpoints chosen are not accurate values for the optimum endpoints. The extrapolations we choose are the most successful ones from the S.O.R. iteration, namely the vector ϵ-algorithm and the various successful implementations of the MPE method. Here the endpoint parameters were given the values $a = 0.0900$ and $b = 7.9000$. These values are upper and lower bounds on the spectrum and are close to the optimal values $a = 0.1620$ and $b = 7.8380$. These are much better estimates than can be obtained for the Gershgorin estimates [14]. Obtaining such good bounds for a general matrix without extensive analysis is unlikely.

Chebychev iteration is a semiiterative method in itself so that we no longer have a simple matrix iteration with which the extrapolations may be formulated. We instead *assume* there is some matrix iteration which approximates the iteration and employ the extrapolations which are, of course, formed from the vector iterates.

Chebychev Iteration with inaccurate parameters

(entries are ℓ_2 norms of the residual)

Iteration	Chebychev	$\text{MPE}^{5:1}_{20:100:1}$	ϵ (diagonal)	ϵ (column 5)
1	1.4142		1.5888	1.5888
6	0.1031	0.2226	10.2945	10.2945
11	0.6106	0.5837	0.9527	0.3578
16	0.0507	0.1039	0.1440	0.1052
21	0.2158	0.0245	0.0661	0.0724
26	0.0260		0.2399	0.2521

Contrary to the widely viewed notion that the ϵ-algorithm is the method of choice, we see that it failed to accelerate the convergence of this iteration. The method also becomes unreasonably expensive, especially down the diagonal, with the extensive movement into different storage elements and the cost of about 500 flops per step in moving up a diagonal. The $\text{MPE}^{5:1}_{20:100:1}$ method produced an effective extrapolation using iterates 21 thorugh 26 resulting in a savings of 5 iterations. Keep in mind that to implement a step of this method we solved a least squares problem costing only about 2500 flops. A single iteration of this extremely cheap method costs over 2600 flops. For full matrices the cost will rise to well over 10,000 flops per iteration. The cost per extrapolation, of course, is the same as for the sparse matrix in the model problem, so there is even greater benefit to be gained from less sparse iterations.

Bibliography

[1] A.C. Aitken. *Determinants and Matrices.* Oliver & Boyd, London, 1956.

[2] P. Axellson. A survey of preconditioning methods for linear systems. *BIT*, 35:166-187, 1985.

[3] P. Axellson and V.A. Barker. *Finite Element Solutions of Boundary Value Problems. Theory and Computation.* Academic Press, New York, 1984.

[4] H. Behnke et al. . *Fundamentals of Mathematics: Vol. II: Geometry.* MIT Press, Cambridge, MA, 1986.

[5] T.N.E. Benisrael and A. Greville. *Generalized Inverses: Theory and Applications.* Wiley, N.Y. 1974.

[6] G. Birkhoff and R. E. Lynch. *Numerical Solution of Elliptic Problems.* SIAM, Philadelphia, Pa., 1984.

[7] C. Brezinski. Convergence acceleration methods: the past decade. *Jour. of Comp. and Applied Math.,* 12 & 13: 19-36, 1985.

[8] C. Brezinski. *Padé-type Approximation and General Orthogonal Polynomials.* Birkhäuser, Basel, 1980.

[9] S. Cabay and L.W. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Jour. of Numer. Anal.*, 13: 734-751, 1976.

[10] F. Chatelin. *Spectral Approximation of Linear Operators.* Academic Press, N.Y., 1983.

[11] P. Davis. *Interpolation and Approximation.* Ginn(Blaisdell), Waltham, Massachusetts, 1963.

[12] C. Deboor, ed.. *Proceedings of Symposia in Applied Mathematics.* Amercian Mathematical Society, Providence, RI, 1986.

[13] D.K. Fadeev and V.N. Fadeeva. *Computational Methods of Linear Algebra.* W.H. Freeman & Co., San Francisco, 1963.

[14] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, Baltimore, Md., 1983.

[15] G.H. Golub and J.H. Wilkinson. Ill-conditioning and the computation of the Jordan canonical form. *SIAM Review*, 18:578-619, 1976.

[16] W. Gragg and L. Reichel. On the Application of Orthogonals Polynomials to the iterative solution of linear systems of equations with indefinite or non-Hermitian matrices. *Linear Algebra and Its Applications.* 88/89:349-371, 1987.

[17] P. R. Graves-Morris. Vector Valued Rational Interpolants I. Num. Math. 42, 331-348.

[18] L. Hageman and D. Young. *Applied Iterative Methods.* Academic Press, N.Y., 1981.

[19] T. Håive. Generalized Neville type extrapolation schemes. *BIT.* 19:204-213, 1979.

[20] M. R. Hestenes. *Conjugate Direction Methods in Optimization.* Springer-Verlag, N.Y., 1986.

[21] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge Univ. Press, N.Y., 1985.

[22] D.G. Luenberger. *Optimization by Vector Space Methods.* Wiley, New York, 1969.

[23] J.B. McLeod. A note on the ϵ-algorithm. *Computing,* 7: 17-24, 1971.

[24] M.Z. Nashed. *Generalized Inverses and Applications.* Academic Press, New York, 1976.

[25] B. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[26] B.N. Parlett and W.G. Poole. A geometric theory for the QR, and LU and power iterations. *SIAM J. Num. Anal.*, 10: 389-412, 1973.

[27] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51: 406-413, 1955.

[28] J.R. Rice. *Numercial Methods, Software, and Analysis.* McGraw Hill, New York, 1983.

[29] Y. Saad. On the rates of convergence of the Lanzcos and block- Lanzcos methods. *SIAM Jour. Numer. Anal.*, 23: 178-198, 1986.

[30] A. Sidi, W. F. Ford, and D. A. Smith. Acceleration of convergence of vector sequences. *SIAM Jour. of Numer. Anal.,* 23:178- 198, 1986.

[31] G.W. Stewart. Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. *Num. Math.*, 25:123-136, 1976.

[32] R.S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, New Jersey, 1962.

[33] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, New York, 1965.

[34] J. Wimp. *Sequence Transforms and Their Applications.* Academic Press, N.Y., 1981.

[35] Jet Wimp. Derviative-free iteration processes. *SIAM Jour. of Num. Anal.*, 7:329-334, 1970.

[36] P. Wynn. On a device for computing the $e_m(S_n)$ transformation. *MTAC*, 10: 91-96, 1956.

[37] P. Wynn. Acceleration techniques for iterated vector and matrix problems. *Math. Comp.*, 16: 301-322, 1962.