A FAMILY OF PRECONDITIONED ITERATIVE SOLVERS FOR SPARSE LINEAR SYSTEMS

BY

ULRIKE MEIER YANG

Dipl., Ruhruniversität Bochum, 1983

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1995

Urbana, Illinois

A FAMILY OF PRECONDITIONED ITERATIVE SOLVERS
FOR SPARSE LINEAR SYSTEMS

Ulrike Meier Yang, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1995
Kyle A. Gallivan, Advisor

Based on the relationship between the family of Broyden methods and the EN method, a new family of iterative methods, the family of EN-like methods, is developed and analyzed. These methods are shown to be related to a variety of other known methods, which comprise the Broyden methods, GCR, GMRES, Newton's method for approximating the inverse, and a combination of a Galerkin step followed by a step of Richardson's method. Scaling-invariant versions and implementations of higher efficiency are developed, and their complexity is examined. The convergence of the new methods, as well as their restarted and truncated versions, are examined. Various convergence results are derived, which include termination within a finite number of steps and estimates for residuals and errors. The methods are also shown to be suitable in the context of inner/outer iteration schemes, and, for two of the methods, orthogonality preserving inner/outer iteration schemes are developed. Adaptive versions, which are a combination of truncated and restarted versions, and which automatically increase the size of the Krylov subspace, are included in the software package PARASPAR, which achieves robustness by reevaluating a parametrized preconditioner whenever poor convergence or instability is encountered. We present numerical experiments that demonstrate the efficiency of several members of this new family in comparison with other known methods, in the context of PARASPAR, and in the context of inner/outer iteration schemes. The experiments show that even though methods like CGS and BiCGSTAB may converge faster for many problems, EN-like methods are in general more robust, since, like GMRES, they have the option of increasing the size of the Krylov subspace. Additionally, they often require less memory than GMRES or ORTHOMIN. They are also very suitable as iterative solvers inside PARASPAR, since they evaluate, without additional cost, parameters that estimate the quality of the preconditioner. Some EN-like methods also show drastic divergence when applied to an ill-conditioned problem. Consequently, it is possible to quickly recognize when to revealuate the preconditioner. Finally, nonlinear EN-like methods are developed, and their convergence behavior is investigated.

To my husband, Ming Hong, and my son, Alexander.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support, encouragement and vision of Kyle Gallivan, my thesis advisor. I am very grateful for all the helpful discussions we have had over the past several years. I also thank the rest of my examination committee, Michael Heath, Faisal Saied and Stratis Gallopoulos for all their comments and suggestions, which have improved this thesis. Stratis also provided me with a variety of helpful references.

I am also grateful to Paul van Dooren for various inspiring discussions and helpful suggestions. I would like to thank Zahari Zlatev and Jerzy Wasniewski for the original PARASPAR-code, without which many of the experiments described in this thesis would not have been possible. I also thank Merle Levy for checking this thesis for grammatical and spelling errors.

I would also like to thank Ahmed Sameh, who originally invited me to come to the University of Illinois, for the support he has given me over the years.

I am grateful to my parents, Kurt and Ilselore Meier, who have instilled in me to persevere when times are tough, and my mother-in-law, Xuqing Zhu, who has saved me many worries by taking very good care of my son during his first year of life. Finally, my special thanks goes to my husband, Ming Hong, for all his encouragement and for always being there for me.

This work was supported in part by the National Science Foundation under Grant No. CCR-9120105.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

There still is a great need to find a robust efficient iterative solver for general sparse linear systems, $Ax = b$, where A is an $n \times n$-matrix. Such systems arise in many applications. A large number of efficient iterative methods [21] have been developed. Many of these methods often fail, however, but the more robust methods available tend to converge too slowly [28].

The most popular solvers are influenced by the conjugate gradient method (CG), which is a very effective method for solving symmetric positive definite sparse linear systems but is not directly applicable to nonsymmetric or indefinite linear systems. CG has two fundamental properties: it minimizes the residual vector along each iteration step, and the residual vectors satisfy a three-term recurrence. It is not possible to maintain both properties for a nonsymmetric linear system [16], [49]. There are several ways to deal with this situation.

One can maintain both properties by considering the normal equations instead of the original system. The new matrices $A^T A$ (CGNR) or AA^T (CGNE) are symmetric positive definite, and the CG algorithm can therefore be applied to them. This approach is considered in [14]. The disadvantage of using the normal equations is that the condition number, which plays a role in the convergence rate of CG, increases significantly.

One can maintain the minimization property by choosing the direction vector as a linear combination of the residual vector and k previous direction vectors. Such an approach is used, e.g., for the generalized conjugate residual method (GCR) [14] or the generalized minimal residual method (GMRES) [39]. The disadvantage of these methods is that they require substantial storage for the previous direction vectors. In order to reduce the storage, restarted or truncated versions of the above algorithms are used, which involve a fixed number k of previous direction vectors. In general, the convergence rate improves with increasing k. The possibility of increasing k gives these methods robustness compared to those considered next, but at the cost of requiring more storage.

One can also choose to maintain the three-term recurrence property, as is done, e.g., in the biconjugate gradient algorithm (BICG) [17], conjugate gradient squared (CGS) [41], and the stabilized version of CGS (BICGSTAB) [47]. This approach leads to a low memory requirement, but also to a lack of robustness. This can be mitigated by the use of a preconditioner, but the methods often fail completely [28].

Another approach is the quasi-minimal residual algorithm (QMR) [19], which is based on a quasi-minimization of the residual. The original QMR-algorithm involves the, possibly costly, multiplication of the transpose of the matrix with a vector. Freund [18] developed a transpose-free QMR-algorithm (TFQMR), which is related to the CGS-algorithm. Based on these algorithms and the ones mentioned above, a variety of other similar algorithms have been and are being developed that are combinations of the above ideas, such as QMRS [20] or QMRCGSTAB [7], which behave similarly to some of the methods considered above [45].

Two methods that have not received much attention for solving linear systems are Broyden's method [4] and the EN method [13], which was developed fairly recently. The family of Broyden methods has suffered from a bad reputation for solving linear systems, but recent efforts [8] have shown that different line search strategies lead to versions that are competitive with GMRES. The EN method can be related to GMRES [51]. A closer look at the EN method and Broyden methods shows that they are also related. Using this relationship, one can define a new family of methods, the EN-like methods, which includes the EN method as a special case.

In this thesis, we consider both families. Under certain assumptions, members of both families terminate after a finite number of steps and have superlinear convergence. As with other iterative methods such as GCR and GMRES, the full methods are too expensive, and restarted, truncated and adaptive versions must be considered. The computational complexity per iteration step of methods of the EN-like family is almost twice as high as that of the corresponding Broyden methods. While the full EN-like methods in many cases converge only about twice as fast, their restarted, truncated and adaptive versions often converge significantly more than twice as fast as the corresponding Broyden methods, GCR and GMRES, and are therefore more efficient. The possibility of increasing the subspace dimension gives the EN-like method a greater robustness than methods like CGS and BiCGSTAB. We will also see that they often require less memory than the Broyden methods, GCR and GMRES.

Methods in both families require, like most other iterative methods, a good preconditioner in order to be robust. There are different ways to precondition iterative methods. We will consider here two different types of preconditioners, the use of an inner iterative method as a preconditioner similar to GMRESR [48] or FGMRES [37] as well as an Incomplete LU factorization based on numerical dropping. For the former type of preconditioning we assess the new algorithms as inner, as well as outer, method. The latter preconditioner is taken from PARASPAR, a robust parallel software package based on Y12M [22], which has many other interesting features. The new family of methods appears to be very suitable for the strategy used in PARASPAR that gives it its robustness.

Finally, due to its relationship with Broyden's methods, it is possible to develop nonlinear EN-like methods.

1.2 Overview of the Thesis

In the rest of this chapter, we present definitions and notation used throughout this thesis. Chapter 2 decribes the EN method, the family of Broyden methods, the new family of EN-like methods and some of their basic properties. Scaling-invariant versions are developed, and the generalized conjugate residual method (GCR) is presented. Finally, the relationship of the methods to well-known methods such as GCR, Newton's method and others is explored. In Chapter 3, more efficient versions of the methods are presented, including restarted and truncated versions. Additionally, computational complexity with regard to operations and memory usage is discussed. Chapter 4 examines the convergence behavior of the families. It is shown that most of the methods terminate within a finite number of steps. The conditions under which local superlinear convergence can be expected are derived. Additionally, the convergence behavior of restarted and truncated methods is investigated. In Chapter 5, several types of preconditioning are considered. The use of inner iterative methods to approximate the inverse of the matrix is considered. We also discuss modifications of this approach that preserve the orthogonality of the inner methods' subspace to the subspace of the outer iteration. Additionally, the software package PARASPAR, which is used as a framework for our experiments and uses an incomplete LU-factorization with numerical dropping, is described. Adaptive versions of the methods are developed and described in the context of PARASPAR, including the choice of stopping and restarting criteria. In Chapter 6, we demonstrate through numerical experiments the effectiveness of the methods as well as the validity of many of the theoretical results presented in the previous chapters. Chapter 7 considers nonlinear versions of the new family in comparison to the corresponding nonlinear versions of the Broyden family. The one-dimensional case is considered separately. The convergence of the n-dimensional case is examined and q-superlinear and n-step q-quadratic convergence is shown for the nonlinear EN-like methods. Finally, Chapter 8 presents the conclusions of this research and outlines future work.

1.3 Notation and Definitions

Throughout this thesis, all scalars, vectors, and matrices are assumed to be complex unless explicitly stated otherwise; in Chapter 8 they are assumed to be real. $\mathcal{R}^n$ denotes the n-dimensional Euclidean space (where $\mathcal{R} := \mathcal{R}^1$), and $\mathcal{C}^n$ the n-dimensional unitary space (where $\mathcal{C} := \mathcal{C}^1$). Matrices are denoted by capital Roman letters and vectors by lower case Roman letters. Matrix transposition is indicated by the superscript T, and conjugate transposition by the superscript H. The norm, $\| \; \|$, refers to the vector 2-norm and induced matrix norm, and $\|A\|_F := \sqrt{trace(AA^H)}$ is the *Frobenius norm* of a matrix A, where the *trace* of a square matrix is the sum of its diagonal entries. $R(A)$, $N(A)$, and $rank(A)$ denote the *range*, *null space*, and *rank* of A, respectively. $\mathcal{V} \oplus \mathcal{W}$ denotes the *direct sum* of the vectorspaces $\mathcal{V}$ and $\mathcal{W}$ and $dim(\mathcal{V})$ the *dimension* of $\mathcal{V}$. The vector space spanned by the vectors $v_1, ..., v_n$ is denoted by $span\{v_1, ..., v_n\}$. The k-dimensional *Krylov subspace* is defined by $\mathcal{K}_k(A, x) := span\{x, Ax, ..., A^{k-1}x\}$.

For the convergence theory in Chapters 4 and 7, we need the following definitions (see also [11]):

Definition 1.1 Let $x \in \mathcal{C}^n$ (or $\mathcal{R}^n$), $x_k \in \mathcal{C}^n$ (or $\mathcal{R}^n$), $k = 0, 1, ...$

Then the sequence $\{x_k\} = \{x_0, x_1, ...\}$ *converges q-linearly* if there exists a constant $c \in [0, 1)$ and an integer $\hat{k} \geq 0$ such that for all $k \geq \hat{k}$

$$\|x_{k+1} - x\| \leq c\|x_k - x\|.$$

$\{x_k\}$ *converges q-superlinearly* if there exists an integer $\hat{k} \geq 0$ and a sequence $\{c_k\}$ that converges to 0 such that for all $k \geq \hat{k}$

$$\|x_{k+1} - x\| \leq c_k \|x_k - x\|.$$

$\{x_k\}$ *converges q-quadratically* if there exists a constant $c \geq 0$ and an integer $\hat{k} \geq 0$ such that for all $k \geq \hat{k}$

$$\|x_{k+1} - x\| \leq c \|x_k - x\|^2.$$

$\{x_k\}$ *converges n-step q-quadratically* if there exists a constant $c \geq 0$ and an integer $\hat{k} \geq 0$ such that for all $k \geq \hat{k}$

$$\|x_{k+n} - x\| \leq c \|x_k - x\|^2.$$

Definition 1.2 $C^1(D)$ denotes the space of all continuously differentiable functions $F : D \to \mathcal{R}^n$, where $D \subset \mathcal{R}^n$.

Definition 1.3 A function $F : D \to \mathcal{R}^n$, where $D \subset \mathcal{R}^n$, is *Lipschitz continuous* if it satisfies

$$\|F(x) - F(y)\| \leq \gamma \|x - y\|$$

where $\gamma > 0$, $x, y \in D$. This condition is denoted $F \in Lip_\gamma(D)$.

CHAPTER 2

DESCRIPTIONS AND BASIC CHARACTERISTICS OF METHODS

In this chapter, several methods based on rank-one updates and their basic characteristics are presented. We first describe the little-known EN method and the family of Broyden methods to which the EN method is related. Using this connection, we derive the new family of EN-like methods and present some of its basic properties. Next, we consider the scaling-dependence of both the Broyden and EN-like families. While scaling-dependence is, under certain conditions, a useful property (see Chapter 6), it can also be a problem; we develop scaling-invariant versions of the EN-like and Broyden methods. We also consider the generalized conjugate residual method (GCR), which can also be presented as a method based on rank-one updates. In Section 2.6, relationships of Broyden and EN-like methods to other methods are presented. Finally, Section 2.7 contains a summary of the methods.

2.1 The EN Method

The EN method was first proposed by Eirola and Nevanlinna [13]. The main idea is to improve an approximation to A^{-1}, H_k, via a rank-one update on each iteration of the method while simultaneously improving an approximation x_k to the solution of the linear system. The rank-one update, denoted $\tilde{u}_k v_k^H$, is chosen in such a way that the k-th approximation of the inverse is no worse than the $(k-1)$-th approximation. If we quantify the quality of the approximation by defining the error matrix

$$E_k := I - AH_k,$$

the constraints on the rank-one update can be achieved by first choosing

$$v_k = \frac{E_k^H A\tilde{u}_k}{\|A\tilde{u}_k\|^2},$$

and then considering $\tilde{u}_k$. If we define the residual $r_k := b - Ax_k$, the best choice is

$$\tilde{u}_k = A^{-1}E_k r_k,$$

which leads to $r_{k+1} = 0$. Of course, such a choice clearly begs the question of solving the system of linear equations. Therefore, Eirola and Nevanlinna used H_k, the best available approximation of A^{-1}, and set

$$\tilde{u}_k = H_k E_k r_k.$$

The resulting algorithm is:

Algorithm 1. EN method (original version)

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$

For $k = 0, 1, \ldots$:

$$\begin{aligned}
\tilde{u}_k &= H_k(I - AH_k)r_k \\
v_k &= \frac{(I - H_k^H A^H)A\tilde{u}_k}{\|A\tilde{u}_k\|^2} \\
H_{k+1} &= H_k + \tilde{u}_k v_k^H \\
E_{k+1} &= I - AH_{k+1} \\
x_{k+1} &= x_k + H_{k+1} r_k \\
r_{k+1} &= E_{k+1} r_k
\end{aligned}$$

end

This method has several interesting properties, some of which we consider later in the context of the more general class of EN-like methods. Theorem 2.1 summarizes a few of the basic characteristics of the EN method that have been noted in the literature (see [13] and [51]) but not proved and presented in a unified fashion.

Theorem 2.1 *Define the following vectors:*

$$
\begin{aligned}
u_k &:= \frac{\tilde{u}_k}{\|A\tilde{u}_k\|} \\
c_k &:= Au_k.
\end{aligned}
$$

Then, the EN method has the following properties

(i) $E_{k+1} = (I - c_k c_k^H)E_k$,

(ii) $c_i^H c_j \;=\; \delta_{ij}$ *(orthogonality),*

(iii) $u_i^H A^H A u_j \;=\; \delta_{ij}$ *($A^H A$-conjugacy),*

(iv) $c_i^H r_{j+1} \;=\; 0, \quad j \geq i,$

(v) *the individual singular values of E_k do not increase with increasing k.*

Proof: (i)

$$
\begin{aligned}
E_{k+1} &= I - AH_{k+1} \\
&= I - AH_k - \frac{A\tilde{u}_k \tilde{u}_k^H A^H E_k}{\|A\tilde{u}_k\|^2} \\
&= (I - c_k c_k^H)E_k.
\end{aligned}
$$

(ii) For $i = j$, $c_i^H c_i = 1$ according to its definition. If we define $q_i := AH_i r_i$, then we have

$$c_i = \frac{E_i q_i}{\|E_i q_i\|}.$$

It is easy to see that

$$E_{k+1}q_k = (I - c_k c_k^H)E_k q_k = 0$$

and using induction and (i), we get for $j > 0$

$$E_{k+j}q_k = 0.$$

We prove $c_i^H c_j = 0$ for $j > i$, using induction. Certainly,

$$
\begin{aligned}
c_0^H c_1 &= q_0^H E_0^H E_1 q_1 \\
&= q_0^H E_0^H (I - c_0 c_0^H) E_0 q_1 \\
&= q_0^H E_1^H E_0 q_1 \\
&= 0.
\end{aligned}
$$

We assume for $j > i$

$$c_i^H c_{j-1} = 0.$$

Since the multiplication of $I - vv^H$ and $I - ww^H$ is commutative, if $v^H w = 0$, we have

$$\begin{aligned}
c_i^H c_j &= q_i^H E_i^H E_j^H q_j \\
&= q_i^H E_i^H (I - c_{j-1}c_{j-1}^H)...(I - c_i c_i^H) E_i q_j \\
&= q_i^H E_j^H E_i q_j \\
&= 0.
\end{aligned}$$

(iii) follows from (ii) since $c_i = Au_i$.

(iv) Using (i), we have

$$\begin{aligned}
c_i^H r_{i+1} &= c_i^H E_{i+1} r_i \\
&= c_i^H (I - c_i c_i^H) E_i r_i \\
&= 0.
\end{aligned}$$

(v) can be proved using (i) and Theorem 5.9 in [42]. Since the singular values of E_k are the square roots of the eigenvalues of the Hermitian matrix $E_k^H E_k$, the following relationship is valid

$$\begin{aligned}
\sigma_i(E_{k+1}) &= \min_{dim(S)=i} \max_{x \in S \setminus \{0\}} \frac{\|E_{k+1}x\|^2}{\|x\|^2} \\
&= \min_{dim(S)=i} \max_{x \in S \setminus \{0\}} \frac{\|(I - c_k c_k^H)E_k x\|^2}{\|x\|^2} \\
&\leq \min_{dim(S)=i} \max_{x \in S \setminus \{0\}} \frac{\|E_k x\|^2}{\|x\|^2} \\
&= \sigma_i(E_k).
\end{aligned}$$

□

The results of this theorem are applied in later chapters. Properties (i) to (iv) are used in Chapter 3 to improve the efficiency of the implementation of the method. Properties (ii) and (iv) imply that the EN method terminates within n steps (see Chapter 4). Property (v) implies that the 2-norm and the Frobenius norm (or any other unitarily invariant norm [25]) of E_k do not increase with k, a fact that is used for the convergence theory in Chapter 4.

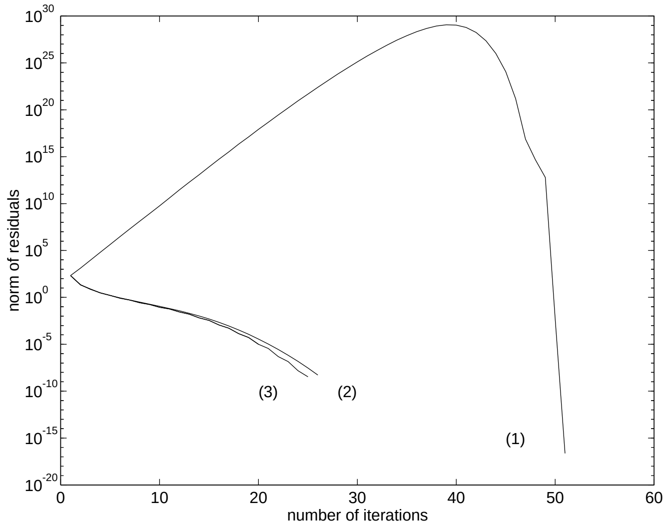

Figure 2.1 Convergence behavior of the EN method for varying H_0

To illustrate the convergence behavior of the methods presented in this chapter we use a test problem from [51]. The matrix is of the form SDS^{-1}, where D is a diagonal matrix with the diagonal vector $(1, 2, ..., 50)^T$, and

$$S = \begin{pmatrix} 1 & \beta & & \\ & \cdot & \cdot & \\ & & \cdot & \beta \\ & & & 1 \end{pmatrix}.$$

We chose $\beta = 0.9$ and the order of the matrix to be 50, which leads to $\lambda_{min} = 1$ and $\lambda_{max} = 50$.

The convergence behavior of the EN method for various $H_0 = cI$ is shown in Figure 2.1, where $c = 1$ (1), $c = 1/\lambda_{max}$ (2) and $c = 2/(\lambda_{max} + \lambda_{min})$ (3) (where λ_{min} and λ_{max} are the smallest and the largest eigenvalues of A).

The sensitivity of the convergence to changes in H_0 is an effect of the scaling-dependence of the EN method (which is also considered in [51]). We develop scaling-invariant versions of the EN method in Section 2.4.

2.2 The Family of Broyden Methods

An important class of methods based on rank-k updates is the class of quasi-Newton methods [10]. The purpose of quasi-Newton methods is to determine a zero of a function F or minimize a function G using approximations of the Jacobian of F, or the Hessian of G, or their inverses. There are several effective quasi-Newton methods, including the Davidon-Fletcher-Powell method and the BFGS method [10]. These methods assume symmetric (and often positive definite) matrices, and we do not consider them here, since our goal is to solve nonsymmetric linear systems. Instead, we focus on variants of Broyden's method, a quasi-Newton method that is suitable for solving nonsymmetric linear systems [4].

If F is defined by $F(x) = Ax - b$, then its Jacobian equals A, and Broyden's method, in its most general form, is given by

Algorithm 2. Broyden's method (with B_k approximating A)

Initialization:

x_0, B_0 arbitrary, $r_0 \; = \; b - Ax_0$

For $k = 0, 1, \ldots$:

$$\begin{aligned} Solve\ \ B_k p_k &= r_k \\ q_k &= A p_k \\ x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k q_k \\ B_{k+1} &= B_k + \frac{(q_k - B_k p_k) g_k^H}{g_k^H p_k} \end{aligned} \tag{2.1}$$

end

where g_k needs to be chosen in such a way, that $g_k^H p_k \neq 0$. Since this algorithm still requires solving the linear system $B_k p_k = q_k$, Broyden also introduced a similar algorithm that approximates A^{-1} and can be obtained by applying the Sherman-Morrison formula to (2.1).

Algorithm 3. Broyden's method (with H_k approximating A^{-1})

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$

For $k = 0, 1, \ldots$:

$$\begin{aligned} p_k &= H_k r_k \\ q_k &= A p_k \\ x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k q_k \\ H_{k+1} &= H_k + \frac{(p_k - H_k q_k) f_k^H}{f_k^H q_k} \end{aligned}$$

end

where f_k needs to be chosen so that $f_k^H q_k \neq 0$ and $g_k = B_k^H f_k$. We focus here on Algorithm 3, and in this thesis the term 'Broyden's method' refers to this algorithm when discussing solving linear systems. Algorithm 2 is referred to as 'Broyden's direct method'.

2.2.1 Choice of f_k

There are two undefined variables, f_k and α_k, whose choice is considered in this and the next section. Based on Algorithm 2, Broyden suggested using $f_k = H_k^H p_k$ [4]. The resulting algorithm is known in the literature as the 'good' Broyden method (GBM). It can be shown that, with this choice, H_{k+1} is the solution to the minimization problem

$$min \|H_k^{-1} - B\|_F,$$

(where $\|.\|_F$ denotes the Frobenius norm) on the set of all matrices B that fulfill the secant condition

$$Bp_k = q_k, \tag{2.2}$$

see [11].

By a similar argument, minimizing $\|H_k - H\|_F$ where H is an element of the set of all matrices that fulfill the form of the secant condition

$$Hq_k = p_k, \tag{2.3}$$

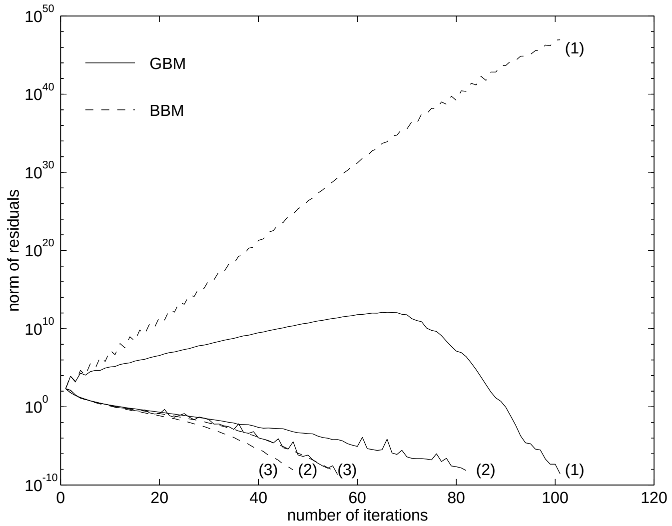

Figure 2.2 Convergence behavior of GBM and BBM with $\alpha_k = 1$

one obtains the choice $f_k = q_k$. In the literature the resulting method is known as the 'bad' Broyden method (BBM). Obviously from the name, this variant often does not perform as well as GBM.

Figure 2.2 shows the convergence behavior of GBM and BBM with $\alpha_k = 1$ for our test problem. Case (1) clearly shows how Broyden's 'bad' method got its name, since for this case the method diverges, even though in exact arithmetic it must terminate within $2n$ steps (see Chapter 4). For the cases (2) and (3), however, it converges faster than GBM.

These are the best known choices for f_k. There are, however, other interesting choices. For the special case of a Hermitian matrix A, a Hermitian update for H_k is needed, which yields the choice $f_k = p_k - H_k q_k$. The interesting aspect of this method is that, if A and H_0 are Hermitian, then H_k, the approximation of the inverse of A, is constructed in such a way that $H_{k+1} q_i = p_i, i \leq k$, for $k+1$ points $q_i = Ap_i, i = 0, \ldots, k$. Consequently, this method terminates within at most n steps (see also [27]), since the algorithm constructs H_k, a better approximation to A^{-1}, on the k-th iteration, until finally $H_n = A^{-1}$, if $f_i^H q_i \neq 0, i = 0, ..., n-1$. Unfortunately, this is not the case if A is nonsymmetric, and finite termination within n steps is no longer guaranteed. Interestingly, if we apply this method, which we call the Hermitian Broyden method (HBM), to our nonsymmetric test problem (see Figure 2.3), it converges within

68 iterations and consequently performs better than GBM (which takes 100 iterations) or BBM (which diverges).

In order to get the same effect as HBM for the nonsymmetric case and preserve termination within n steps, one needs to constrain f_k to be orthogonal to q_i, $i = 0, ..., k$.

Lemma 2.1 *If A is nonsingular, $p_0, \ldots, p_k$ are given linearly independent vectors, and $q_i = Ap_i$, $i = 0, \ldots, k$, with*

$$H_{i+1} = H_i + \frac{(p_i - H_i q_i) f_i^H}{f_i^H q_i}, \tag{2.4}$$

$$f_k^H q_i = 0, \qquad i < k, \qquad f_i^H q_i \neq 0, \tag{2.5}$$

then

$$p_i = H_{k+1} q_i, \qquad i \leq k, \tag{2.6}$$

and termination is guaranteed in n steps.

Proof: We prove the above by induction. First consider the case $i = k$:

$$\begin{aligned} H_{k+1} q_k &= H_k q_k + \frac{(p_k - H_k q_k) f_k^H q_k}{f_k^H q_k} \\ &= H_k q_k + p_k - H_k q_k \\ &= p_k \end{aligned}$$

Now assume the assumption is true for k. For $i < k$, we have

$$\begin{aligned} H_{k+1} q_i &= H_k q_i + \frac{(p_k - H_k q_k) f_k^H q_i}{f_k^H q_k} \\ &= p_i. \end{aligned}$$

□

Our previous choices for f_k do not fulfill the condition of Lemma 2.1, and therefore, in general, the method does not terminate within n steps.

One obvious choice for f_k which fixes the above problem, but also increases the amount of work per iteration significantly, is

$$f_k = z_k - \sum_{i=0}^{k-1} \tilde{q}_i^H z_k \tilde{q}_i, \tag{2.7}$$

where $z_k \neq 0$ is some arbitrary vector in $\mathcal{R}^n$, and the vectors $\tilde{q}_0, \ldots, \tilde{q}_{k-1}$ are an orthonormal basis of the space spanned by $q_i, i = 0, \ldots, k-1$. The best choice for z_k is q_k, since the resulting error matrix $E_{k+1} = I - AH_{k+1}$ can be determined through the product of E_k and a projection matrix (see proof of Lemma 4.11), and $\|E_k\|$ does not increase with k.

Note, that the choice

$$f_k = H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i), \tag{2.8}$$

where the vectors $\tilde{p}_0, \ldots, \tilde{p}_{k-1}$ are an orthonormal basis of the space spanned by $p_i, i = 0, \ldots, k-1$, also leads to termination within n steps, which can be shown by applying the above argument to Broyden's direct method.

Both methods were developed independently by Gay and Schnabel [23] who call them Broyden's method with projected updates. We call the first method with projected updates 'PBM', and the second one, which was inspired by Broyden's direct method 'dual PBM'. See Figure 2.3 for their convergence behavior when applied to our test problem.

There is another choice of f_k that leads to a version of Broyden's method that is equivalent to the generalized conjugate residual method (GCR) [12] or GMRES with the initial vector $\tilde{x}_0 = x_0 + H_0 r_0$ and consequently terminates within n steps [51]. For this method we choose $f_k = (I - AH_k)^H(I - AH_k)q_k$. Its convergence behavior is similar to that of GMRES or GCR, which is considered in further detail later. We refer to this method as 'GCR-BM'. See Figure 2.3 for its convergence behavior when applied to our test problem.

Just as GBM is the dual method to BBM and there is a dual method for PBM, it is possible to define a 'dual GCR-BM' with $f_k = H_k^H(I - A^{-1}H_k^{-1})^H(I - A^{-1}H_k^{-1})p_k$. This method, due to the occurrence of A^{-1}, is of only theoretical interest, but is included in the convergence investigations in Chapter 4.

In the experiments in Chapter 6, we focus on GBM and BBM, since the symmetric update, despite its unexpected performance on our test problem, is not generally useful for nonsymmetric systems; and Broyden's method with projected updates is related to GCR and GMRES (see Section 2.6), which have lower computational complexity.

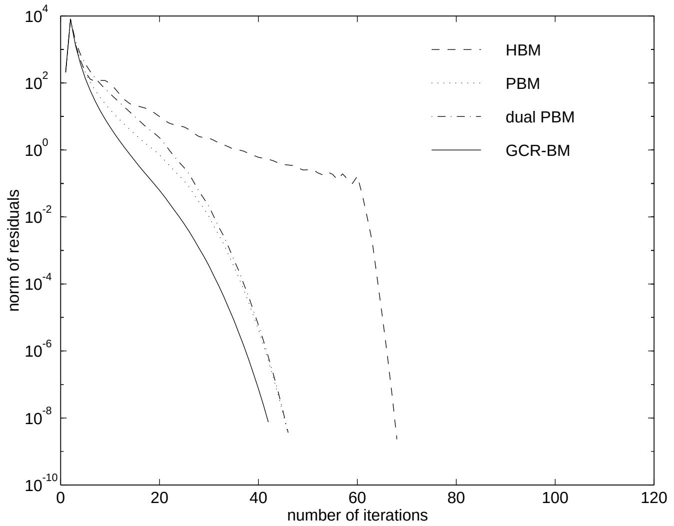

Figure 2.3 Broyden's method with various f_k's, $\alpha_k = 1$, $H_0 = I$

2.2.2 Choice of α_k

Now consider, α_k, the second undetermined parameter. The most obvious choice is $\alpha_k = 1$. One can show that for this case Broyden's method terminates within at most $2n$ steps [24] (see also Chapter 4). However, the convergence behavior of GBM and BBM for case (1) of our test problem in Figure 2.2 shows that this is not always a desirable choice despite its finite termination property.

Broyden suggests in [4] choosing α_k so that $\|r_{k+1}\| < \|r_k\|$. He also states that this choice of α_k can lead to worse results than choosing an α_k that does not necessarily fulfill $\|r_{k+1}\| < \|r_k\|$, e.g., $\alpha_k = 1$. Our experiments confirm this statement. Figure 2.4 shows the convergence behavior of GBM with $\alpha_k = 1$ and a damped version where

$$\alpha_k = \frac{q_k^H r_k}{\|q_k\|^2} \tag{2.9}$$

is chosen to minimize $\|r_{k+1}\|$. For case (1), the damped version converges faster while the unmodified version performs better for the other two cases, with case (3) yielding the best results.

Deuflhard et al. [8] showed that the best α_k for a method depends on the choice of f_k and proposed

$$\alpha_k = \frac{f_k^H r_k}{f_k^H q_k}. \tag{2.10}$$

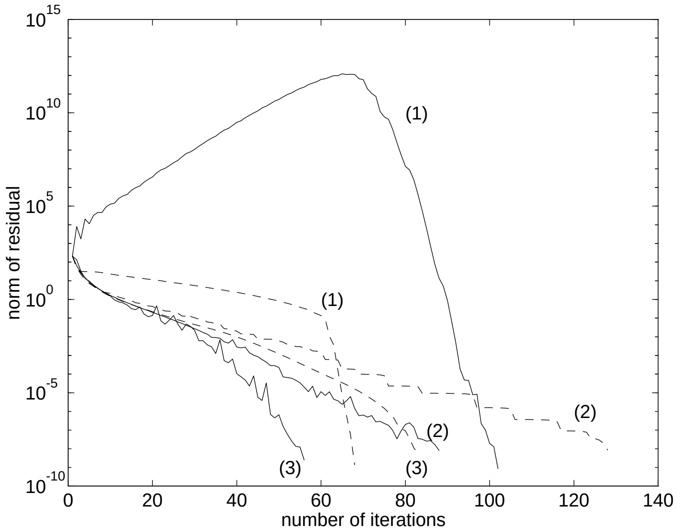

Figure 2.4 Convergence behavior of GBM for $\alpha_k = 1$ (solid curves) and $\alpha_k = q_k^H r_k / q_k^H q_k$ (dashed curves)

Figure 2.5 confirms their result. Their experiments showed that this choice produces Broyden methods that are competitive with GMRES.

2.3 The Family of EN-like Methods

We have seen that the EN-like method and the family of Broyden methods have several apparent similarities. They are both based on rank-one updates. Both are scaling-dependent. The methods also show similar convergence behavior (compare Figures 2.1 and 2.2) on our simple test problem. In this section, we investigate this relationship more formally, and using it, define a new family of EN-like methods.

If we use the definitions of p_k and q_k in Algorithm 3 and define $E_k = I - AH_k$, we can rewrite the evaluation of H_{k+1} as

$$H_{k+1} = H_k + \frac{H_k E_k r_k f_k^H}{f_k^H A H_k r_k}. \tag{2.11}$$

Setting $f_k = E_k^H E_k A H_k r_k$, we obtain

$$H_{k+1} = H_k + \frac{H_k E_k r_k r_k^H E_k^H H_k^H A^H E_k}{||E_k A H_k r_k||^2} \tag{2.12}$$

which looks just like H_{k+1} as evaluated in the EN method. Consequently, it is possible to derive a family of methods with a general f_k, just as can be done for Broyden's method.

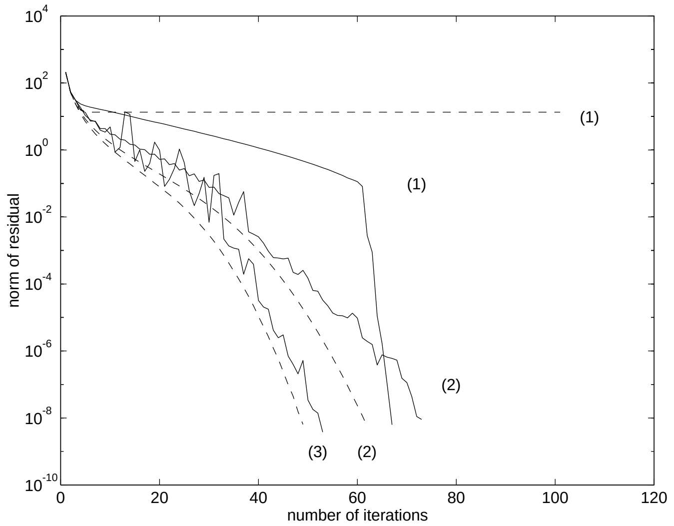

Figure 2.5 GBM (solid curves) and BBM (dashed curves) with $\alpha_k = f_k^H r_k / f_k^H q_k$

The new EN-like method is given by

Algorithm 4. EN-like method

Initialization:

x_0, H_0 arbitrary, $r_0 \;=\; b - Ax_0$, $E_0 \;=\; I - AH_0$,

For $k = 0, 1, \ldots$:

$$
\begin{aligned}
H_{k+1} &= H_k + \frac{H_k E_k r_k f_k^H}{f_k^H A H_k r_k} \\
E_{k+1} &= I - AH_{k+1} \\
x_{k+1} &= x_k + H_{k+1} r_k \\
r_{k+1} &= E_{k+1} r_k
\end{aligned}
$$

end

where f_k needs to be chosen so that $f_k^H A H_k r_k \neq 0$.

Algorithm 4 can also be written in a form that more closely resembles Broyden's method as follows:

Algorithm 5. EN-like method

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$

For $k = 0, 1, \ldots$:

$$
\begin{aligned}
p_k &= H_k r_k \\
q_k &= A p_k \\
H_{k+1} &= H_k + \frac{(p_k - H_k q_k) f_k^H}{f_k^H q_k} \\
\tilde{p}_k &= H_{k+1} r_k \\
\tilde{q}_k &= A p_k \\
x_{k+1} &= x_k + \tilde{p}_k \\
r_{k+1} &= r_k - \tilde{q}_k
\end{aligned}
$$

end

where f_k needs to be chosen so that $f_k^H q_k \neq 0$.

Note that in this form it becomes apparent that Broyden's method is to the EN-like method as the Jacobi method is to the Gauss-Seidel method. For Broyden's method, the direction vector is evaluated using H_k, while for the EN-like method H_{k+1}, the new approximation to A^{-1}, is used, which can lead to faster convergence at the cost of increased complexity per iteration.

Some preliminary experiments using our simple test problem show that the EN-like methods converge about twice as fast as the corresponding Broyden method. Figure 2.6 shows the convergence behavior of GBM and the corresponding EN-like method with $f_k = H_k^H p_k$, which we call 'GEN', and Figure 2.7 shows the results for BBM and the corresponding EN-like method with $f_k = q_k$, which we call 'BEN'. We also see that the disturbing increase in the norm of the residual that occurs in case (1) for Broyden's method is significantly increased for the EN-like method.

Other choices for f_k considered in Chapter 4 are $f_k = H_k^H(I - A^{-1}H_k^{-1})^H(I - A^{-1}H_k^{-1})p_k$, which yields a dual method for EN (denoted by 'dual EN'), $f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$, which yields an EN-like method based on projected updates (denoted by 'PEN'), and $f_k = H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$, which yields a dual method for PEN (denoted by 'dual PEN').

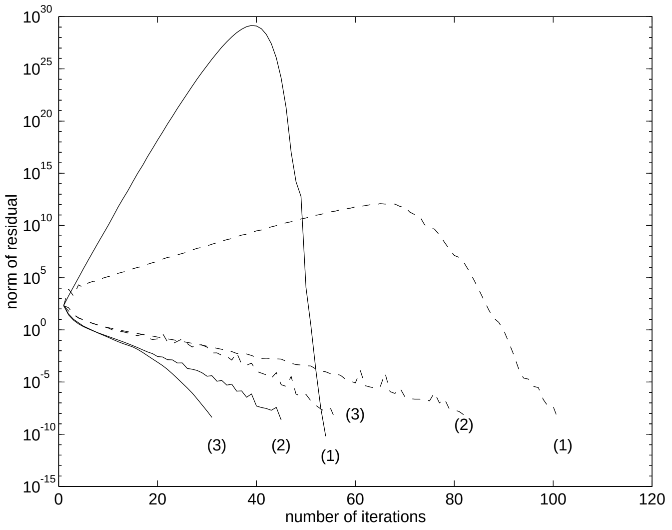

Figure 2.6 GBM (dashed curves) vs. GEN (solid curves)

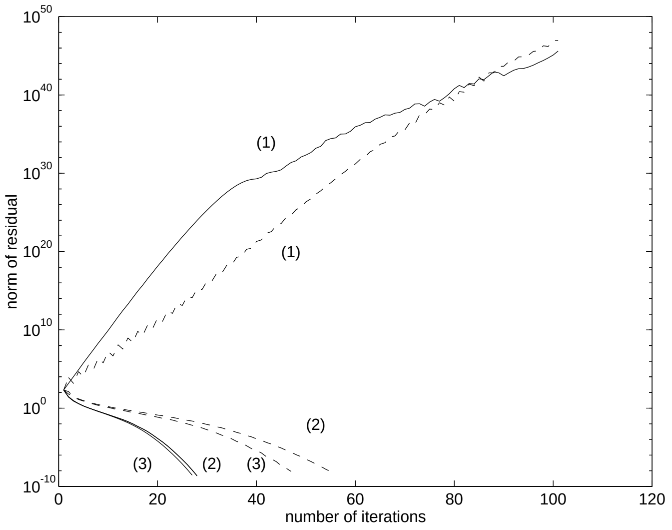

Figure 2.7 BBM (dashed curves) vs. BEN (solid curves)

We have already noted the similarity of two steps of Broyden to one step of the corresponding EN-like method. If we initialize H_k, x_k, and f_k to identical values for both methods, the effect of a single step of an EN-like method compared with two steps of Broyden's method can be formalized in the following lemma.

Lemma 2.2 *If $H_k^B = H_k^E$, $x_k^B = x_k^E$ and $f_k^B = f_k^E$, the following identities hold (where the omission of E or B denotes that the corresponding vectors or matrices are equal, e.g., r_k stands for r_k^B and r_k^E, which are equal):*

$$
\begin{aligned}
H_{k+1}^B &= H_{k+1}^E && (2.13)\\
\tilde{p}_k &= p_k^B + p_{k+1}^B && (2.14)\\
x_{k+2}^B &= x_{k+1}^E && (2.15)\\
r_{k+2}^B &= r_{k+1}^E && (2.16)\\
H_{k+2}^B &= H_{k+1} + \frac{H_{k+1}E_{k+1}E_k r_k (f_{k+1}^B)^H}{(f_{k+1}^B)^H A H_{k+1} E_k r_k} && (2.17)\\
H_{k+2}^E &= H_{k+1} + \frac{H_{k+1}E_{k+1}^2 r_k (f_{k+1}^E)^H}{(f_{k+1}^E)^H A H_{k+1} E_{k+1} r_k} && (2.18)
\end{aligned}
$$

Even though it appears here that two steps of a Broyden method lead to the same solution x_{k+2}^B as one step of the corresponding EN-like method, this is not the case, since $H_{k+2}^B \neq H_{k+1}^E$, and further iteration steps of corresponding methods build different solution spaces that are not necessarily related. Unfortunately, further investigation of the spaces using this approach does not lead to any significant insight. It is possible, however, to show a more useful relationship between one step of an EN-like method and Broyden's method.

Lemma 2.3 *One iteration step of an EN-like method can be decomposed in the following way:*

$$
\begin{aligned}
\tilde{x}_{k+1} &= x_k + H_k r_k && (2.19)\\
\tilde{r}_{k+1} &= r_k - A H_k r_k && (2.20)\\
x_{k+1} &= \tilde{x}_{k+1} + \alpha_k H_k \tilde{r}_{k+1} && (2.21)
\end{aligned}
$$

where

$$\alpha_k = \frac{f_k^H r_k}{f_k^H q_k}.$$

Proof: The proof is straightforward. □

Lemma 2.3 shows that one step of an EN-like method can be considered as an iteration step of the corresponding Broyden's method with $\alpha_k = 1$ followed by an iteration step of Broyden's method using the optimal line search principle of [8] without updating the approximation to A^{-1}.

2.4 Scaling-Invariance

As noted earlier, it is possible for scaling of the linear system to affect an iterative solver's performance significantly. This can be made more formal by considering the concept of scaling-invariance.

Definition 2.1 An iterative method is called *scaling-invariant* if $x_k = \tilde{x}_k$, where x_k is the k-th iterate generated by applying the iterative solver to $Ax = b$, and $\tilde{x}_k$ is the k-th iterate generated by applying the iterative solver to $\rho Ax = \rho b$ for any $\rho \neq 0$.

In general, neither Broyden's method nor the EN-like method is scaling-invariant. This is also indicated by the experiments, where we chose $H_0 = cI$, which is equivalent to applying the solvers to the linear systems $(1/c)Ax = (1/c)b$ with various choices for c.

One way to remedy this problem is to premultiply H_0 by a scaling parameter in the initialization phase of the algorithm. We denote the methods that are generated this way by a prefix 's' (e.g. sEN for the scaling-invariant version of the EN method).

Theorem 2.2 *For a Broyden or EN-like method define $H_0 = \gamma M$, where*

$$\gamma = \frac{(Az)^H z}{\|Az\|^2} \quad \textit{or} \quad \gamma = \frac{(AMz)^H z}{\|AMz\|^2}, \tag{2.22}$$

z is an arbitrary vector with $Az \neq 0$, and M is a nonsingular matrix. If $\alpha_k = \tilde{\alpha}_k$ for Broyden's method, and $\tilde{f}_k = g(\rho) f_k$ for a function $g : \mathcal{C} \to \mathcal{C}$, (where $\tilde{\alpha}_k$ and $\tilde{f}_k$ are generated by applying the method to the scaled system $\rho Ax = \rho b$), then the method is scaling-invariant.

Proof: Clearly, $\tilde{\gamma} = \frac{1}{\rho}\gamma$.

We prove by induction that

$$\begin{aligned} \tilde{p}_k &= p_k, \\ \tilde{q}_k &= \rho q_k, \end{aligned}$$

$$
\begin{aligned}
\tilde{s}_k &= s_k, \\
\tilde{t}_k &= \rho t_k, \\
\tilde{r}_k &= \rho r_k, \\
\tilde{H}_k &= \frac{1}{\rho} H_k, \\
\tilde{x}_k &= x_k.
\end{aligned}
$$

Certainly, these assertions are true for $k = 0$.

Now,

$$
\begin{aligned}
\tilde{p}_k &= \tilde{H}_k \tilde{r}_k = p_k \\
\tilde{q}_k &= \rho A \tilde{p}_k = \rho q_k \\
\tilde{r}_{k+1} &= \tilde{r}_k - \tilde{\alpha}_k \tilde{q}_k = \rho r_{k+1} \\
\tilde{H}_{k+1} &= \tilde{H}_k + \frac{(\tilde{p}_k - \tilde{H}_k \tilde{q}_k)\tilde{f}_k^H}{\tilde{f}_k^H \tilde{q}_k} = \frac{1}{\rho} H_k \\
\tilde{s}_k &= \tilde{H}_{k+1} \tilde{r}_k = s_k \\
\tilde{t}_k &= \rho A \tilde{p}_k = \rho t_k \\
\tilde{x}_{k+1} &= \tilde{x}_k + \tilde{\alpha}_k \tilde{p}_k = x_{k+1}.
\end{aligned}
$$

□

Even though an arbitrary z theoretically guarantees scaling-invariance, its choice is an important practical matter. An unfortunate choice of z might make no difference or even degrade the convergence of the method, whereas a well-chosen z might lead to an improvement in the number of iterations. An example of this can be found in Figure 2.8, where for the dotted curve z is the vector with all components $z_i = 1$, for the solid curve $z = r_0 = b$, and for the dashed curve z is an eigenvector v_1 corresponding to the eigenvalue 1. For the last case $\gamma = 1$, and consequently we see the same behavior as for the scaling-dependent version of the method.

Vuik and van der Vorst [51] suggested another scaling-invariant version of the EN method, which we denote by 'SEN'. While examining the error matrix

$$E_{k+1} = (I - c_k c_k^H)...(I - c_0 c_0^H)E_0, \tag{2.23}$$

they realized that the scaling-dependence was caused by the factor $I - AH_0$, and they suggested introducing a scaling parameter γ_k, leading to the product $(I - c_k c_k^H)...(I - c_0 c_0^H)(I - \gamma_k A H_0)$.

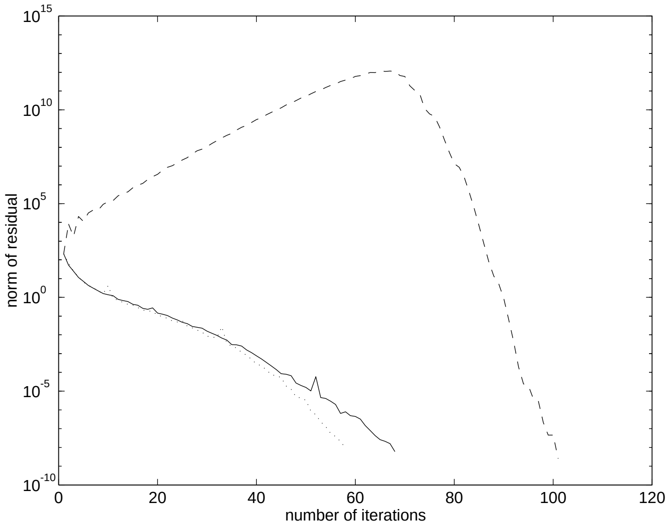

Figure 2.8 sGBM with $z = b$ (solid curve), $z = (1, ..., 1)^T$ (dotted curve) and $z = v_1$ (dashed curve)

The scaling parameter γ_k is determined by minimizing the vector $(I - \gamma_k AH_0)r_k$, which leads to

$$\gamma_k = \frac{(AH_0r_k)^H r_k}{||AH_0r_k||^2}. \tag{2.24}$$

The new method is slightly more expensive and requires two more inner products per iteration (which can, however, be performed simultaneously), but converges significantly better for our test problem (see Figure 2.9). We will see later that this version is not always preferable to the original EN method.

Using the same approach, it is possible to suggest other scaling-invariant versions for Broyden and EN-like methods, which are denoted by a prefix 'Si'. Since

$$E_{k+1} = E_0(I - q_0v_0^H)...(I - q_kv_k^H), \tag{2.25}$$

with $v_k := f_k/(f_k^H q_k)$, the introduction of a scaling parameter γ_k would lead to the product $(I - \gamma_k AH_0)(I - q_0v_0^H)...(I - q_kv_k^H)$. Thus, γ_k could be chosen to minimize the residual, i.e.,

$$\gamma_k = \frac{(AH_0y_k)^H y_k}{||AH_0y_k||^2}, \tag{2.26}$$

with

$$y_k = (I - q_0v_0^H)...(I - q_kv_k^H)r_k. \tag{2.27}$$

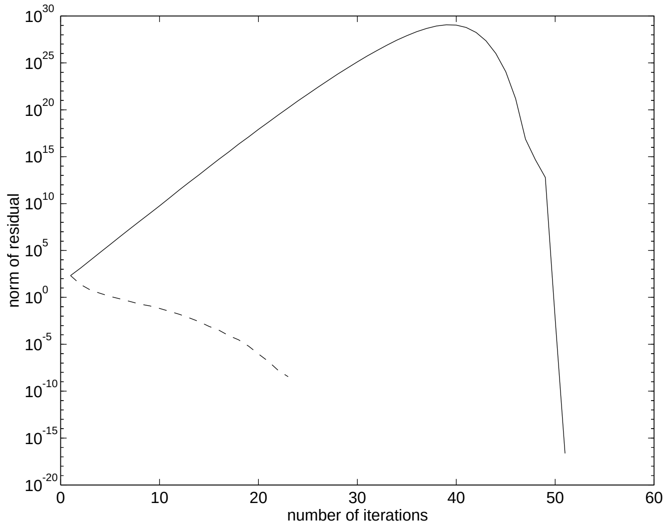

Figure 2.9 EN (solid curve) vs. SEN (dashed curve)

This version is of particular interest for BBM and BEN, since in this case the projections $I - q_i v_i^H$ can be bounded by 1, and the residuals of the new methods do not increase. For our test problem, the scaling-invariant versions of BBM and BEN converge faster than the original methods, and converge, albeit slowly, even for $H_0 = I$ (the case where the original methods failed). Unfortunately, these methods are of more theoretical than practical interest, since the incorporation of such a parameter requires a large number of additional operations per iteration. The computation of y_k requires $k+1$ additional daxpys $(u + \beta v)$ and inner products $(u^H v)$, and the computation of the approximate solution x_k requires an additional $k+1$ daxpys and preconditioning step. For these reasons, these versions are not investigated further in this thesis.

The problem of scaling-invariance has been considered in the context of quasi-Newton methods for nonlinear systems. Since Newton's method is scaling-invariant, some authors were interested in developing scaling-invariant quasi-Newton methods [1] [34]. The definition they use is more general than the one above. Note that none of the scaling-invariant algorithms above is scaling-invariant according to the new definition below.

Definition 2.2 An iterative method is called *scaling-invariant (according to Paloschi and Perkins)* if, for any nonsingular matrices S_x and S_f, $x_k = S_x \tilde{x}_k$, where x_k is the k-th iterate generated by applying the iterative solver to $f(x) = 0$ using as initial start vector x_0, and $\tilde{x}_k$ is the k-th iterate generated by applying the iterative solver to $S_f f(x) = 0$ using as initial start vector $S_x x_0$.

In [34], Paloschi and Perkins show circumstances under which Broyden's method is scaling-invariant according to their definition. The same result is true for the EN-like method and can be shown accordingly.

Theorem 2.3 *Broyden's method is scaling-invariant (according to Paloschi and Perkins), if and only if*

$$\tilde{H}_k = S_x H_k S_f^{-1}. \tag{2.28}$$

They also suggest a few choices for f_k that guarantee scaling-invariance of Broyden's method and seem to give good results for some badly scaled nonlinear systems. Our experiments using these f_k for linear systems, however, showed worse results than the other methods under investigation, and we therefore did not pursue these methods any further.

In summary, we have introduced here the scaling-invariant 's'- and 'Si'-versions of the families of Broyden and EN-like methods, as well as briefly described SEN. The additional cost involved to achieve scaling-invariance is neglible for the 's'-versions, only an additional $2n$ floating point operations (flops). However an additional $4n$ flops per iteration for SEN and at least $6(k+1)n$ flops during the k-th iteration step are required for the 'Si'-versions. We have also mentioned versions that are scaling-invariant using a more general definition due to Paloschi and Perkins, which is not satisfied by the other versions. In the later chapters, we consider only the 's'-versions and SEN, since the other scaling-invariant versions are either too costly or inferior to the SEN and the 's'-methods.

2.5 The Generalized Conjugate Residual Algorithm (GCR)

Another well-known algorithm that is related in several ways to the methods considered in this thesis and plays a major role in the following sections is the generalized conjugate residual algorithm (GCR) [12] [14].

The algorithm is given below:

Algorithm 6. GCR

Initialization:

x_0, H_0 arbitrary, $r_0 \;=\; b - Ax_0$

For $k = 0, 1, \ldots$:

$$\begin{aligned}
\alpha_i &= \frac{\tilde{c}_i^H A H_0 r_k}{\rho_i}\,, \qquad , i = 0, ..., k-1 \\
\tilde{u}_k &= H_0 r_k - \sum_{i=0}^{k-1} \alpha_i \tilde{u}_i \\
\tilde{c}_k &= A H_0 r_k - \sum_{i=0}^{k-1} \alpha_i \tilde{c}_i \\
\rho_k &= \|\tilde{c}_k\|^2 \\
\delta_k &= \frac{\tilde{c}_k^H r_k}{\rho_k} \\
x_{k+1} &= x_k + \delta_k \tilde{u}_k \\
r_{k+1} &= r_k - \delta_k \tilde{c}_k
\end{aligned}$$

end

As mentioned by Eirola and Nevanlinna [13] and proved by Vuik and van der Vorst [51], this algorithm can also be obtained by replacing the evaluation of $\tilde{u}_k$ in the EN method with

$$\tilde{u}_k \;=\; H_k r_k. \tag{2.29}$$

Consequently, GCR can also be considered as a method based on rank-one updates. In fact, Theorem 1.1 is also valid for GCR using the same proof and yields the following properties. (Properties (ii) through (iv) can also be found in [14].)

Corollary 2.1 *Define the following vectors:*

$$\begin{aligned}
u_k &:= \frac{\tilde{u}_k}{\|A\tilde{u}_k\|} \\
c_k &:= A u_k.
\end{aligned}$$

Then, GCR has the following properties

(i) $E_{k+1} = (I - c_k c_k^H) E_k$,

(ii) $c_i^H c_j \;=\; \delta_{ij}$ *(orthogonality),*

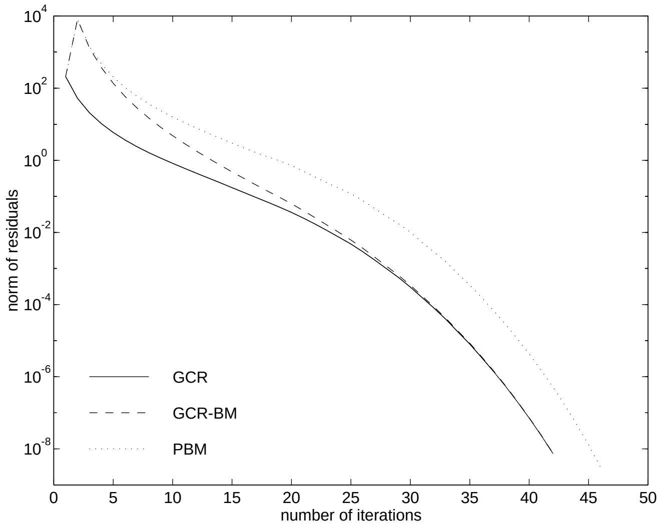

Figure 2.10 GCR and two related Broyden methods with $\alpha_k = 1$, $H_0 = I$

(iii) $u_i^H A^H A u_j \;=\; \delta_{ij}$ *($A^H A$-conjugacy),*

(iv) $c_i^H r_{j+1} \;=\; 0, \quad j \geq i,$

(v) *the individual singular values of E_k do not increase with increasing k.*

It is also easily verified that GCR is scaling-invariant.

2.6 Relationships between Methods

In the previous sections, we have indicated that the Broyden and EN-like methods are related to various other known methods. These relationships are interesting, since they help in gaining a better understanding of the new methods, and they can be used to derive theoretical results for the new methods, e.g. convergence results. The relationships also indicate potential avenues of investigation to improve the new methods. In this section, we show the relationships of PBM to GCR and of the EN-like method to a combination of a simple form of Galerkin's method and Richardson's method. Finally, we consider the relationship of the EN-like methods to Newton's method for approximating the inverse of a matrix.

2.6.1 GCR and Broyden's Method

As mentioned in [51] and in Section 2.2, GCR-BM (Broyden's method with $f_k = E_k^H E_k q_k$) is equivalent to GCR using the initial vector $x_0 + H_0 r_0$. This is equivalent to applying one step of Richardson's method to the linear system $H_0 A x = H_0 b$ with the initial vector x_0 and applying GCR to the resulting vector. This can be proved very easily using the formulation of GCR that is derived from the EN method. Note that even though GCR is scaling-invariant, GCR-BM is not, since Richardson's method is not. Figure 2.10 shows the influence of the initial Richardson step in GCR-BM for $H_0 = I$. If we choose $H_0 = (1/\lambda_{max})I$, the curve for GCR-BM coincides with that of GCR.

The second version that is related to GCR is PBM. Since, like GCR, it terminates within n steps, generates orthogonal vectors, and shows a similar convergence behavior as GCR in our testproblem (see Figure 2.10), it is reasonable to suspect such a relationship. The following lemma relates the f_k generated in PBM to the Krylov subspace generated by GCR.

Lemma 2.4 *For PBM, $f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$ is orthogonal to the k-th Krylov subspace $\mathcal{K}^k(AH_0, AH_0 r_0)$.*

Proof: By definition,

$$f_k \perp span\{q_0, ..., q_{k-1}\}. \tag{2.30}$$

If we can show that

$$q_j \in span\{AH_0 r_0, ..., (AH_0)^{j+1} r_0\}, \tag{2.31}$$

the above assertion follows. We show this statement by induction. It certainly is true for $k = 0$. Assume it is true for k, then

$$\begin{aligned}
q_{k+1} &= AH_{k+1} r_{k+1} \\
&= (AH_k + \frac{(q_k - AH_k q_k) f_k^H}{f_k^H q_k})(r_k - q_k) \\
&= \frac{f_k^H E_k r_k}{f_k^H q_k} E_k q_k \\
&= \frac{f_k^H E_k r_k}{f_k^H q_k} E_0 (I - \tilde{q}_0 \tilde{q}_0^H)...(I - \tilde{q}_{k-1} \tilde{q}_{k-1}^H) q_k \\
&= \frac{f_k^H E_k r_k}{f_k^H q_k} (I - AH_0)(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i),
\end{aligned}$$

which proves the assertion. □

The vectors $\tilde{q}_0, ..., \tilde{q}_{k-1}$ form an orthonormal basis for $\mathcal{K}^k(AH_0, AH_0r_0)$, as can easily be verified using induction. Defining

$$\tilde{r}_k = r_k - \sum_{i=0}^{k-1} \tilde{q}_i^H r_k \tilde{q}_i, \tag{2.32}$$

for which $\tilde{r}_{k+1} = \tilde{r}_k - \tilde{q}_k$, one obtains a sequence of 'residuals' with the property

$$\tilde{r}_k \perp \mathcal{K}^k(AH_0, AH_0r_0). \tag{2.33}$$

Consequently, applying the projectors $I - \tilde{c}_i\tilde{c}_i^H$, $i = 0, ..., k-1$, to the residuals of PBM, generates the residuals of GCR applied to $AH_0(H_0^{-1}x) = b$. It follows that $\|\tilde{r}_k\| \leq \|r_k\|$. Like other Broyden methods, PBM is not scaling-invariant. Figure 2.10 shows the convergence behavior of PBM for $H_0 = I$ for which the norm of the residual of PBM is always larger that that of GCR. If we choose $H_0 = (1/\lambda_{max})I$, the curve for PBM coincides with that of GCR.

Since GCR converges at least as fast as GCR-BM and PBM and can be written in a form with fewer operations per iteration step than GCR-BM and PBM, we do not include GCR-BM and PBM in our experiments in Chapter 6.

2.6.2 Richardson's Method and the Galerkin Method

Another interesting relationship of Broyden's method with Richardson's method and the Galerkin method is pointed out in [8].

Theorem 2.4 *For $\alpha_k = 1$, two steps of Broyden's method are equivalent to one Galerkin step*

$$\tilde{x}_{k+1} = x_k + \frac{f_k^H r_k}{f_k^H q_k} p_k \tag{2.34}$$

followed by one step of Richardson's method of the first order applied to the preconditioned linear system $H_kAx = H_kb$

$$x_{k+1} = \tilde{x}_{k+1} + H_k(b - Ax_{k+1}). \tag{2.35}$$

Proof: See [8]. □

Due to the relationship between Broyden's method and the EN-like method given in Lemma 2.3, a similar theorem can be derived for the EN-like method.

Theorem 2.5 *One step of the EN-like method consists of one Galerkin step followed by one step of Richardson's method.*

Proof: The update

$$x_{k+1} = x_k + p_k + \frac{f_k^H r_k}{f_k^H q_k}(p_k - H_k q_k)$$

can be decomposed into the Galerkin step

$$\tilde{x}_{k+1} = x_k + \frac{f_k^H r_k}{f_k^H q_k} p_k,$$

with the residual

$$\tilde{r}_{k+1} = r_k - \frac{f_k^H r_k}{f_k^H q_k} q_k,$$

(this is a Galerkin step, since $f_k^H \tilde{r}_{k+1} = 0$) and one step of Richardson's method applied to the preconditioned system $H_k A x = H_k b$ with the relaxation parameter $\omega = 1$:

$$x_{k+1} = \tilde{x}_{k+1} + H_k \tilde{r}_{k+1}.$$

□

This relationship is of interest since it possibly provides an opportunity to accelerate the EN-like method. Since the relaxation parameter for Richardson's method here equals 1, this part of the iteration can cause an increase in $\|r_{k+1}\|$, instead of a decrease. The introduction of a relaxation parameter $\omega_k \neq 1$ could possibly lead to better convergence. In order to preserve the secant condition, one needs to redefine H_{k+1} as

$$H_{k+1} = \omega_k H_k + \frac{(p_k - \omega_k H_k q_k) f_k^H}{f_k^H q_k}. \tag{2.36}$$

Consequently ω_k is just a scaling factor for H_k and could be included in its evaluation. Richardson's method achieves the best convergence rate for positive $H_k A$ if $\omega = 2/(\lambda_1 + \lambda_n)$, where λ_1 and λ_n are the two extreme eigenvalues of $H_k A$. Therefore, one would expect that the introduction of ω_k would lead to a significant improvement only when H_k is a bad approximation of A^{-1}. The difficulty lies in the evaluation of ω_k for such a case, since in general $H_k A$ is not available in explicit form, and the approximation of its extreme eigenvalues via an iterative scheme may not converge quickly enough to avoid an unacceptable increase in the work per step.

2.6.3 The EN-like Method and Newton's Method

Since the EN-like method generates H_k, an approximation to A^{-1}, a relationship to matrix iterations that compute the inverse of a matrix is also likely. Such a method can be found in [40].

Choose an arbitrary matrix X_0, and let

$$X_{k+1} = X_k(2I - AX_k).$$

This method is based on Newton's method and possesses quadratic convergence in the sense that

$$I - AX_{k+1} = (I - AX_k)^2.$$

Investigating one step of the EN-like method, one can show that the direction vector evaluated in the EN-like method can be expressed as a convex combination of a direction vector that is evaluated as for Broyden's method using H_k, the previous approximation of A^{-1}, and another vector that uses H_{k+1}^N, a (hopefully) improved approximation of A^{-1}.

Theorem 2.6 *For the EN-like method, the direction vector $H_{k+1}r_k$ can be written:*

$$H_{k+1}r_k = (1 - \omega_k)H_k r_k + \omega_k H_{k+1}^N r_k \tag{2.37}$$

where

$$H_{k+1}^N = H_k(2I - AH_k) \tag{2.38}$$

and

$$\omega_k = \frac{f_k^H r_k}{f_k^H q_k}. \tag{2.39}$$

Proof:

$$\begin{aligned} H_{k+1}r_k &= H_k r_k + \frac{H_k E_k r_k f_k^H}{f_k^H q_k} r_k \\ &= H_k r_k + \omega_k H_k(I - AH_k)r_k \\ &= (1 - \omega_k)H_k r_k + \omega_k H_k^N r_k. \end{aligned}$$

□

Notice that with the direction vector in this form we can see that for $\omega_k = 0$, we have Broyden's method, for $\omega_k = 1$ we have a much faster converging but far more costly method based on Newton's method for approximating the inverse. Of course, in general ω_k would be neither 0 nor 1, since it strongly depends on the vectors f_k, q_k and r_k. However, local convergence considerations show that for some choices of f_k, ω_k converges towards 1 whenever H_0 is a good approximation for A^{-1} (see Chapter 4). In this case, H_{k+1} acts on r_k in a manner similar to the Newton iteration.

Using this knowledge, it is possible to try to improve the convergence by using an acceleration scheme as has been used for Newton's method. Such a scheme has been proposed in [35], and has the form

$$H_{k+1}^N = \alpha_k H_k(2I - AH_k),$$

where α_k is evaluated through the following recursions:

$$\begin{aligned}
\alpha_0 &= \frac{2}{\sigma_1^2 + \sigma_n^2}; \\
\rho_0 &= \alpha_0 \sigma_1^2; \\
\rho_k &= \alpha_{k-1}(2 - \rho_{k-1})\rho_{k-1}; \\
\alpha_k &= \frac{2}{1 + (2 - \rho_k)\rho_k},
\end{aligned}$$

where $k \geq 1$, and σ_1 and σ_n are the smallest and the largest singular value of A. The resulting H_{k+1} has the form

$$H_{k+1} = H_k + \frac{((\alpha_k - 1)H_k r_k + \alpha_k H_k E_k r_k) f_k^H}{f_k^H q_k}.$$

Note, however, that such a strategy destroys the secant condition and the resulting effect on convergence must be carefully considered. Also, one would expect this to be effective only if ω_k is close to 1. As ω_k approaches 0 the term with $H_k r_k$ begins to dominate, and the resulting method approaches Broyden's method. As is shown in Chapter 6, this can often lead to a degradation in convergence.

Family		Choice of f_k
Broyden	EN-like	
BBM	BEN	$f_k = q_k$
GCR-BM	EN	$f_k = (I - AH_k)^H (I - AH_k) q_k$
PBM	PEN	$f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$
HBM	-	$f_k = p_k - H_k q_k$
GBM	GEN	$f_k = H_k^H p_k$
dual GCR-BM	dual EN	$f_k = H_k^H (I - A^{-1} H_k^{-1})^H (I - A^{-1} H_k^{-1}) p_k$
dual PBM	dual PEN	$f_k = H_k^H (p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$

Table 2.1 Notation for Broyden and EN-like methods

2.7 Summary of Methods

In this chapter we have presented a variety of iterative methods based on rank-one updates. Among these are the families of Broyden and EN-like methods. Both have a free parameter f_k which must be specified in order to select particular family members.

There are two categories of choices for f_k: the first is inspired by Broyden's method, which approximates the inverse; the second one by Broyden's direct method. For each member of the first category, there is a dual member in the second category. The different choices for f_k, and the names used for the remainder of the thesis to denote the associated members of the Broyden and EN-like families, are summarized in Table 2.7. The corresponding scaling-invariant versions are denoted using the names listed and a lower case 's' prefix. The scaling-invariant EN method due to Vuik and van der Vorst is denoted SEN.

CHAPTER 3

EFFICIENCY CONSIDERATIONS

In this chapter more efficient implementations of the full methods are presented. These versions avoid the expensive explicite computation of the matrices H_k, which are generally dense. In order to limit memory requirements and computational complexity per iteration step, restarted and truncated versions are also considered. Finally, the computational complexity and memory requirements of these methods are investigated.

3.1 Implementations with Higher Efficiency

One of the disadvantages of the methods we have considered so far is their computational complexity. While A is a sparse matrix, H_k is in general dense and therefore its multiplication with a vector would require $O(n^2)$ number of operations, compared to $O(n)$ for the multiplication of A with a vector. One can develop more efficient versions for all of these methods by avoiding the actual computation of H_{k+1}. Such an approach can also be found in [8] and [15] for Broyden's method and in [13] for the EN method.

If we replace H_{k+1} by its definition, we get

$$\begin{aligned} H_{k+1}x &= H_k x + \frac{f_k^H x}{f_k^H q_k}(p_k - H_k q_k) \\ &= H_0 x + \sum_{i=0}^{k} \frac{f_i^H x}{f_i^H q_i}(p_i - H_i q_i). \end{aligned}$$

Using the following definitions:

$$p_k = H_k r_k, \tag{3.1}$$

$$q_k = Ap_k, \tag{3.2}$$

$$t_k = H_k q_k, \tag{3.3}$$

$$z_k = p_k - H_k q_k, \tag{3.4}$$

we can rewrite Broyden's method in the following form:

Algorithm 7. Broyden's method

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$, $p_0 = H_0 r_0$, $q_0 = Ap_0$,

For $k = 0, 1, \ldots$:

$$\begin{aligned}
\zeta_k &= f_k^H q_k \\
t_k &= H_0 q_k + \sum_{i=0}^{k-1} \frac{f_i^H q_k}{\zeta_i} z_i \\
z_k &= p_k - t_k \\
x_{k+1} &= x_k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k q_k \\
p_{k+1} &= (1 - \alpha_k) p_k + \frac{f_k^H r_k}{\zeta_k} z_k \\
q_{k+1} &= A p_{k+1}
\end{aligned}$$

end

Replacing f_k with q_k yields BBM. Similarly, for PBM, set $f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$. GBM, however, or any other member of the class for which the evaluation of f_k involves H_k^H, requires further transformation. The new version of GBM contains the terms $f_{k-i}^H q_k = p_{k-i}^H H_i q_k$. We therefore need to determine the vectors

$$t_k^{(i)} = H_i q_k. \tag{3.5}$$

Fortunately, they can be evaluated without significantly increasing the number of operations or the storage needed if one chooses the order of computations carefully. Unfortunately, the evaluation is highly recursive and leads to a decrease in parallelism. The new version for GBM is given below.

Algorithm 8. GBM

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$, $p_0 = H_0 r_0$, $q_0 = Ap_0$, $t_0 = H_0 q_0$,

For $k = 0, 1, \ldots$:

$$\begin{aligned}
t_k^{(0)} &= H_0 q_k \\
t_k^{(i)} &= t_k^{(i-1)} + \frac{p_{i-1}^H t_k^{(i-1)}}{\zeta_i} z_{i-1}, \quad i = 1, \ldots, k \\
\zeta_k &= p_k^H t_k^{(k)} \\
z_k &= p_k - t_k^{(k)} \\
x_{k+1} &= x_k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k q_k \\
p_{k+1} &= (1 - \alpha_k) p_k + \frac{p_k^H p_k}{\zeta_k} z_k \\
q_{k+1} &= A p_{k+1}
\end{aligned}$$

end

Using the definitions above, the general EN-like method can be rewritten as follows

Algorithm 9. EN-like method

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$,

For $k = 0, 1, \ldots$:

$$\begin{aligned}
p_k &= H_0 r_k + \sum_{i=0}^{k-1} \frac{f_i^H r_k}{\zeta_i} z_i \\
q_k &= A p_k \\
\zeta_k &= f_k^H q_k \\
t_k &= r_k - q_k \\
z_k &= H_0 t_k + \sum_{i=0}^{k-1} f_i^H t_k z_i \\
s_k &= p_k + \frac{f_k^H r_k}{\zeta_k} z_k \\
x_{k+1} &= x_k + s_k
\end{aligned}$$

$$r_{k+1} = r_k - As_k$$

end

For the BEN method, we set $f_k = q_k$. As for GBM, for the GEN method we need to evaluate, in addition to (3.5)

$$p_k^{(i)} = H_i r_k \tag{3.6}$$

The new efficient version is given by

Algorithm 10. GEN method ($f_k = H_k^H p_k$)

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$,

For $k = 0, 1, \ldots$:

$$\begin{aligned}
p_k^{(0)} &= H_0 r_k \\
p_k^{(i)} &= p_k^{(i-1)} + \frac{(p_{i-1}^{(i-1)})^H p_k^{(i-1)}}{\zeta_i} z_{i-1}, \quad i = 1, \ldots, k \\
q_k &= A p_k^{(k)} \\
t_k^{(0)} &= H_0 q_k \\
t_k^{(i)} &= t_k^{(i-1)} + \frac{(p_{i-1}^{(i-1)})^H t_k^{(i-1)}}{\zeta_i} z_{i-1}, \quad i = 1, \ldots, k \\
\zeta_k &= (p_k^{(k)})^H t_k^{(k)}) \\
z_k &= p_k^{(k)} - t_k^{(k)} \\
s_k &= p_k^{(k)} + \frac{(p_k^{(k)})^H p_k^{(k)}}{\zeta_k} z_k \\
x_{k+1} &= x_k + s_k \\
r_{k+1} &= r_k - As_k
\end{aligned}$$

end

For the EN method, where $f_k = E_k^H E_k q_k$, we use

$$\begin{aligned}
\tilde{c}_k &= E_k q_k, \\
\tilde{u}_k &= H_k E_k r_k.
\end{aligned}$$

Due to the orthogonality of the $\tilde{c}_i$ (see (ii) in Theorem 2.1), we have

$$q_i^H E_i^H E_i x = q_i^H E_i^H E_0 x.$$

This leads to the following version for SEN (see also [13] [51]):

Algorithm 11. SEN method

x_0, H_0 arbitrary, $r_0 \; = \; b - Ax_0$,

For $k = 0, 1, \ldots$:

$$\begin{aligned}
\gamma_k &= \frac{(AH_0 r_k)^H r_k}{\|AH_0 r_k\|^2} \\
\alpha_i &= \gamma_k \frac{\tilde{c}_i^H AH_0 r_k}{\rho_i}\,, \qquad i = 0, \ldots, k-1 \\
p_k &= \gamma_k H_0 r_k - \sum_{i=0}^{k-1} \alpha_i \tilde{u}_i \\
t_k &= (I - \gamma_k AH_0) r_k + \sum_{i=0}^{k-1} \alpha_i \tilde{c}_i \\
\beta_i &= \frac{\tilde{c}_i^H AH_0 t_k}{\rho_i}\,, \qquad i = 0, \ldots, k-1 \\
\tilde{u}_k &= H_0 t_k - \sum_{i=0}^{k-1} \beta_i \tilde{u}_i \\
\tilde{c}_k &= AH_0 t_k - \sum_{i=0}^{k-1} \beta_i \tilde{c}_i \\
\rho_k &= \|\tilde{c}_k\|^2 \\
\delta_k &= \frac{\tilde{c}_k^H t_k}{\rho_k} \\
x_{k+1} &= x_k + p_k + \delta_k \tilde{u}_k \\
r_{k+1} &= t_k - \delta_k \tilde{c}_k
\end{aligned} \tag{3.7}$$

end

The original EN method can be obtained by setting $\gamma_k = 1$.

If we do not require the evaluation of x_{k+1} at each iteration step, it is possible to achieve further savings in EN and SEN by avoiding the evaluation of u_k. Only the coefficients need to be saved. Such an approach has been used in [44], to improve the implementation for GCR given in Algorithm 6. The more efficient implementation of GCR is

Algorithm 12. eGCR

Initialization:

x_0, H_0 arbitrary, $r_0 \;=\; b - Ax_0$, $\tilde{x}_0 = x_0$,

For $k = 0, 1, \ldots, m$:

$$
\begin{aligned}
\alpha_{i,k} &= \frac{c_i^H A H_0 r_k}{\rho_i}\,, \qquad i = 0, ..., k-1 \\
\hat{u}_k &= H_0 r_k \\
\tilde{c}_k &= A H_0 r_k - \sum_{i=0}^{k-1} \alpha_{i,k} \tilde{c}_i \\
\rho_k &= \|\tilde{c}_k\|^2 \\
\delta_k &= \frac{\tilde{c}_k^H r_k}{\rho_k} \\
r_{k+1} &= r_k - \delta_k \tilde{c}_k
\end{aligned}
$$

end

$$x_{m+1} = x_0 + \hat{U}_m A_m^{-1} d_m,$$

where

$$
\begin{aligned}
d_m &= (\delta_0, ..., \delta_m)^H, \\
\hat{U}_m &= (\hat{u}_0, ..., \hat{u}_m), \\
A_m &= \begin{pmatrix} 1 & \alpha_{0,1} & ... & \alpha_{0,m} \\ & 1 & . & . \\ & & . & \alpha_{m-1,m} \\ & & & 1 \end{pmatrix}.
\end{aligned}
$$

Using the same approach, we can generate a more efficient version for SEN.

Algorithm 13. eSEN method

x_0, H_0 arbitrary, $r_0 \;=\; b - Ax_0$, $\tilde{x}_0 \;=\; x_0$,

For $k = 0, 1, \ldots m$

$$\gamma_k \;=\; \frac{(AH_0 r_k)^H r_k}{\|AH_0 r_k\|^2}$$

$$
\begin{aligned}
\beta_{i,k} &= \gamma_k \frac{\tilde{c}_i^H A H_0 r_k}{\rho_i}, \quad i = 0, \ldots, k-1 \\
t_k &= (I - \gamma_k A H_0) r_k + \sum_{i=0}^{k-1} \beta_{i,k} \tilde{c}_i \\
\hat{u}_k &= H_0 t_k \\
\alpha_{i,k} &= \frac{\tilde{c}_i^H A H_0 t_k}{\rho_i}, \quad i = 0, \ldots, k-1 \\
\tilde{c}_k &= A H_0 t_k - \sum_{i=0}^{k-1} \alpha_{i,k} \tilde{c}_i \\
\rho_k &= \|\tilde{c}_k\|^2 \\
\delta_k &= \frac{\tilde{c}_k^H t_k}{\rho_k} \\
\tilde{x}_{k+1} &= \tilde{x}_k + \gamma_k H_0 r_k \\
r_{k+1} &= t_k - \delta_k \tilde{c}_k
\end{aligned}
$$

end

$$
x_{m+1} = \tilde{x}_{m+1} + \hat{U}_m B_m^{-1}(d_m - A_m e),
$$

where $\hat{U}_m$, A_m and d_m are defined as above, and

$$
\begin{aligned}
e &= (1, 1, ..., 1)^H, \\
B_m &= \begin{pmatrix} 0 & \beta_{0,1} & \ldots & \beta_{0,m} \\ & 0 & . & . \\ & & . & \beta_{m-1,m} \\ & & & 0 \end{pmatrix}.
\end{aligned}
$$

The more efficient version for EN, which we call eEN method, is obtained by setting $\gamma_k = 1$ in Algorithm 13.

3.2 Restarted and Truncated Methods

Due to a significant increase in both operation count and memory requirement with each iteration, even the efficient versions of the methods considered are computationally expensive. Therefore, we consider restarted and truncated versions. An overview of restarted and truncated algorithms can be found in [38].

The restarted version of any of the methods can be described as follows:

Algorithm 14. Restarted method(m).

Do, until convergence,

1. Perform $m+1$ iterations of method,
2. set $x_0 = x_{m+1}$
3. go back to 1.

end;

Most of the efficient versions of the methods in the previous section also lend themselves to truncation. A truncated version of any of the methods (truncated method(m)) above can be obtained by including only updates of the last m iterations, i.e. for Algorithms 6 through 11 replacing all occurrences of $\sum_{i=0}^{k-1}$ by $\sum_{i=max(0,k-m+1)}^{k-1}$. Consequently, after the first m iterations, one keeps the newest m vectors in a moving window instead of discarding them as is done in the restarted version. One would expect such an approach to lead to better convergence, since more information is being kept. We see in Chapter 4 and Chapter 6, that this not always true. In fact, Deuflhard, Freund and Walter [8] saw in their experiments for Broyden's methods that this approach in general is worse than restarting. Our experiments show, however, that this result does not transfer to the EN-like methods (see Chapter 6). It is also possible to use more sophisticated truncation schemes, which can lead to better convergence, see [43] and [50]. The application of these to the family of EN-like methods is left as future work.

The truncation of Algorithms 12 and 13 is far more complicated than the other algorithms. De Sturler and Fokkema describe in [43] how to truncate Algorithm 12. The difficulty is that for this version x_k is not updated in each iteration step. However due to the truncation, the basis of the Krylov subspace and the parameters change at each iteration step and need to be updated. Consequently, the computational complexity is significantly increased and is at least as large as that of the truncated versions of Algorithms 6 and 11. Therefore, these truncated versions are not considered here.

One of the disadvantages of restarted and truncated methods is the fact that a new parameter k is introduced. It is unclear how to choose it, since a small k might lead to extremely slow convergence or possibly divergence, but a large k leads to a high number of operations per iteration step. We have developed adaptive versions that gradually increase the subspace size depending on the convergence rate. These versions are considered in detail in Chapter 5.

3.3 Complexity

In order to compare the methods with each other as well as with existing methods, it is necessary to develop a model. Various issues need to be considered:

- the total operation count,
- the type of operations,
- the memory requirements.

Since we are dealing with iterative methods, the total operation count consists of the number of operations per iteration ste, which we consider in this section, and the number of iterations required to achieve a certain accuracy. The second aspect is considered in more detail in Chapter 4.

It is important to consider the type of operations involved, since the performance of computational primitives can differ significantly from one another and depends strongly on the type of machine used. This is particularly true for parallel computers, our target architectures, which perform operations such as dense matrix-vector multiplications more efficiently than, e.g., general sparse matrix-vector multiplications.

In Table 3.1 the types and number of operations are given for a variety of methods. We have included GCR and the equivalent GMRES [39] here, since they are related to the methods considered (see next section), and CGS [41] and BiCGSTAB [47], since they are very popular solvers. For Broyden's methods, only the operation counts for optimal line search (where $\alpha_k = f_k^H r_k \; f_k^H q_k$) according to [8] are given. Those for GBM or BBM with $\alpha_k = 1$ are slightly lower. We use 'dmv' to denote dense matrix-vector multiplication. There are two types of dense matrix-vector operations included: first, any computations of inner products of the form $c_i^H x, \; i = 0, ..., k-1$, can also be performed as the multiplication of the vector x with the matrix $(c_0, ..., c_{k-1})$ of order $n \times k$, and second, the operation $\sum_{i=0}^{k-1} \alpha_i c_i$ can be performed as a multiplication of the matrix $(c_0, ..., c_{k-1})$ with the vector $(\alpha_0, ..., \alpha_{k-1})^H$. Both matrix-vector multiplications take about $2kn$ flops. We also consider the following vector operations: the inner product of two vectors ('dp'), the vector operation $x = x + \alpha y$ ('daxpy'), and simple vector additions, subtractions or multiplications of a vector with a scalar (+,-,*). Another important part of each algorithm is multiplication of a sparse matrix with a vector ('smv'). The flop-counts depend strongly on the linear system we are considering. They can be small for a very sparse matrix and require little time if the matrix is also well structured, such as banded. They can be large if the matrix is fairly dense or given in implicit form. Another unknown factor is the preconditioning step ('prec'), which also strongly depends on the preconditioner used. For a diagonal preconditioner, e.g., the cost is almost negligible; for an incomplete LU preconditioner, however, it can be quite expensive due to its potentially low degree of parallelism.

Note that for GMRES and the efficient versions eGCR, eEN and eSEN, a postprocessing step that consists of one 'dmv' ($2mn$ flops after m iterations) is required in order to obtain the actual solution, and an additional 'dmv' is required for GMRES if one needs to get the residual vector (as is necessary for restarted GMRES).

Note that the 'dmv's here are actually k inner products or k daxpys that can be performed simultaneously. Consequently, methods with 'dmv's, such as BBM, GCR, BEN, EN, etc., have a higher degree of parallelism than methods such as GMRES in its usual implementation with modified Gram-Schmidt, GBM and GEN, which have to perform those inner products and daxpys recursively. It is possible to use the classical Gram-Schmidt algorithm for GMRES to increase its parallelism. In this case its operation count, as given in Table 3.1, is identical to that of eGCR. Since it also can lead to instability, however, it has been suggested that the classical Gram-Schmidt algorithm be used twice [9].

In Tables 3.2 and 3.3 the number of floating point operations per iteration step for the truncated and restarted versions are given. We use the notation 'tMethod(m)' for the truncated versions and 'Method(m)' for the restarted versions. Note that tGCR(m) is better known as ORTHOMIN(m). The numbers are straightforward to obtain for the truncated methods, assuming that one uses a window of m vectors and neglects startup cost. For the restarted methods, one can give only the average number of iterations per iteration step, since the actual number changes due to the changing size of the window. The averaged operation count also takes into account the previously mentioned postprocessing step for the efficient methods. Clearly, using restarted methods leads to lower operation counts per iteration step, consequently the

Method	dmv	dp	daxpy	+,-,*	smv	prec
op. count	$2kn$	$2n$	$2n$	n		
BBM	2	2	3	1	1	1
GBM	-	k+2	k+3	1	1	1
GCR	3	2	2	-	1	1
eGCR	2	2	1	-	1	1
GMRES	-	k+2	k	-	1	1
BEN	4	2	1	3	2	2
GEN	-	2k+2	2k+1	2	2	2
EN	6	2	2	2	2	2
eEN	4	2	1	2	2	2
SEN	6	4	3	2	2	2
eSEN	4	4	3	2	2	2
CGS	-	2	5	1	2	2
BiCGSTAB	-	4	6	-	2	2

Table 3.1 Number and types of operations for various methods in k-th iteration step

Method	number of ops. per iteration
tBBM(m)	(4m+11)n + 1 smv + 1 prec
tGBM(m)	(4m+11)n + 1 smv + 1 prec
ORTHOMIN(m)	(6m+8)n + 1 smv + 1 prec
tBEN(m)	(8m+9)n + 2 smv + 2 prec
tGEN(m)	(8m+9)n + 2 smv + 2 prec
tEN(m)	(12m+10)n + 2 smv + 2 prec
tSEN(m)	(12m+16)n + 2 smv + 2 prec
CGS	15n + 2 smv + 2 prec
BiCGSTAB	20n + 2 smv + 2 prec

Table 3.2 Computational complexities for truncated versions

use of truncated methods pays off only if the number of iterations is significantly lower. We consider this tradeoff in more detail in Chapter 6.

Methods like GBM, BBM, GCR and GMRES have the advantage of using only one 'smv' and one 'prec' per iteration step, which in case of an expensive 'smv' and 'prec' makes the other methods attractive only if they converge at least twice as fast.

Our final complexity criterion is the memory required. Table 3.4 lists the amount of memory in terms of vectors of length n required in addition to the matrix and the righthand side. We ignore memory requirements of order k or k^2, since we assume that k is in general small compared to n.

Method	avg. number of ops. per iteration
BBM(m)	$(2m+11)n$ + 1 smv + 1 prec
GBM(m)	$(2m+11)n$ + 1 smv + 1 prec
eGCR(m)	$(2m+8)n$ + 1 smv + 1 prec
GMRES(m+1)	$(2m+8)n$ + 1 smv + 1 prec
BEN(m)	$(4m+9)n$ + 2 smv + 2 prec
GEN(m)	$(4m+9)n$ + 2 smv + 2 prec
eEN(m)	$(4m+10)n$ + 2 smv + 2 prec
eSEN(m)	$(4m+16)n$ + 2 smv + 2 prec

Table 3.3 Computational complexities for restarted versions

Method	memory required
(t)BBM(m)	$2m$+5
(t)GBM(m)	$2m$+5 or m+6
(t)GCR(m), eGCR(m)	$2m$+4
(t)BEN(m)	$2m$+6
(t)GEN(m)	$2m$+6
(t)EN(m), eEN(m)	$2m$+4
(t)SEN(m), eSEN(m)	$2m$+4
GMRES(m+1)	m+4
CGS	8
BiCGSTAB	8

Table 3.4 Additional work vectors required

Note that it is possible for GBM to save a substantial amount of storage by replacing z_i in Algorithm 8 by the expression

$$z_i = \frac{\zeta_i}{\|p_i\|^2}(p_{i+1} - (1 - \alpha_i)p_i). \tag{3.8}$$

However, this change requires an additional k daxpys per iteration step if $\alpha_i \neq 1$. Unfortunately, it is not possible to use a similar trick for GEN, since the method is more complicated and substituting the z_i by p_i would involve multiplications by A and H_0.

We see that unless m is small, CGS and BiCGSTAB require less memory than the other methods. All the other methods require approximately the same amount of memory, except for GMRES and GBM, which use about half the amount.

CHAPTER 4

CONVERGENCE THEORY

In earlier chapters we have defined the general form of the family of EN-like methods for sparse linear systems as well as efficient versions of the full, restarted, and truncated members of the family. The amount of work per iteration is greater for the new methods than for other well-known methods, and therefore convergence must improve in order for the family to be viable. We demonstrated that important relationships exist between the family of EN-like methods and the family of Broyden methods. In this chapter, we exploit these relationships to investigate the convergence behavior of the full, restarted, and truncated versions of EN-like methods. We also derive some new results for the family of Broyden methods.

In Section 4.1, we demonstrate that the EN-like methods terminate in a finite number of iterations. In Section 4.2, the local convergence properties of the full versions of the EN-like methods are analyzed. New results for certain members of the family of Broyden methods are derived, and it is demonstrated that the convergence of an EN-like method is twice as fast as the corresponding Broyden method. In Section 4.3, restarted Broyden methods and EN-like methods are considered, and convergence results for each are derived. Results for truncated versions of the EN-like methods (as well as any other method in the literature) are much more difficult to derive. In Section 4.4, we exploit the relationship of the EN-like methods to GCR in order to derive a convergence result for the truncated scaling-invariant EN-like method, tSEN(m). The result is also applicable to the full and restarted SEN methods. Section 4.5 contains a detailed summary of the convergence results.

Throughout this chapter we use the phrases EN-like method and Broyden method to refer to any member of the respective families. If results apply only to a subset of the families, the assumed forms of f_k and α_k are specified. The following definitions are also used:

$$E_k \quad := \quad I - AH_k, \tag{4.1}$$

$$\tilde{E}_k := I - A^{-1}H_k^{-1}, \quad (4.2)$$

$$r_k := b - Ax_k, \quad (4.3)$$

$$e_k := x_k - x. \quad (4.4)$$

4.1 Finite Termination

For the sake of simplicity, we use the following definitions:

$$v_k := \frac{f_k}{f_k^H q_k}, \quad (4.5)$$

$$P_k^{(0)} := E_k E_{k-1} ... E_0, \quad (4.6)$$

and

$$P_k := E_k E_{k-1} ... E_1. \quad (4.7)$$

Unless otherwise stated, we also assume, that we are working with the 2-norm.

One of the amazing and unexpected properties of Broyden's method is its finite termination property when $\alpha_k = 1$. This is shown by Gay in [24].

Theorem 4.1 *For $\alpha_k = 1$, $f_k^H q_k \neq 0$, Broyden's method converges within at most $2n$ steps.*

Proof: See [24]. □

Gay also shows that this upper bound on the number of iterations required for convergence is tight; i.e., convergence can take a full $2n$ steps.

Lemma 4.1 *If $k \geq 1$, $q_k \neq 0$, $v_k^H q_{k-1} \neq 0$, rank(E_k)=$n-1$, then, rank(E_{k+1}) $= n-1$ and $E_{k+1} q_k = 0$.*

Proof: See [24]. □

One interesting consequence of Lemma 4.1 is that, under its assumptions, the matrices H_k agree with the inverse of A only on a subspace of dimension 1. It is only on the last iteration that the solution vector x has the appropriate relationship to the null space of E_k to result in $x_k = x$.

Gay's proof of finite termination, while significant, does not offer an insight into the operation of Broyden's method. Recently, O'Leary [32] characterized the subspace that Broyden's method implicitly constructs. The subspace dimension increases every other step and is orthogonal to the last residual. The orthogonality property guarantees finite termination.

Lemma 4.2 *Assume $v_j^H q_{j-1} \neq 0, \;\; j = 1, ..., k,$ and $q_k \neq 0$. Let v_0 be in the range of E_0^H, and $k \leq 2n - 1$ be odd.*

Define the sequence of vectors

$$\begin{aligned} z_1^H E_1 &= 0, \\ z_i^H E_i &= z_{i-2}^H, \quad i = 3, 5, ..., k. \end{aligned}$$

Then, $z_i, \;\; i = 1, 3, ..., k,$ exist and are linearly independent. Additionally,

$$\begin{aligned} z_i^H P_j^{(0)} &= 0, \qquad j = i, ..., k, \\ z_i^H r_j &= 0, \qquad j = i + 1, ..., k. \end{aligned}$$

Proof: See [32]. □

O' Leary's approach can be used along with the relationship between Broyden's method and the general EN-like method to prove finite termination for the EN-like methods. Lemmas 4.3, 4.4, and 4.5 apply this approach and characterize the behavior of the rank of the error matrix E_k.

Lemma 4.3 *Assume $v_j^H q_{j-1} \neq 0, \;\; , j = 1, ..., k,$ and $q_k \neq 0$. Let v_0 be in the range of E_0^H, and $k \leq n - 1$.*

Define the sequence of vectors

$$\begin{aligned} z_1^H E_1 &= 0, \\ z_i^H E_i &= z_{i-1}^H, \quad i = 2, 3, ..., k. \end{aligned}$$

Then, $z_i, \;\; i = 1, 2, ..., k$ exist and are linearly independent. Additionally,

$$\begin{aligned} z_i^H P_j &= 0, \qquad j = i, ..., k + 1, \\ z_i^H r_j &= 0, \qquad j = i, ..., k + 1. \end{aligned}$$

Proof: The lemma is proved by induction.

Let us prove the above for $k = 1$:

Define z_1 through $z_1^H E_0 = v_0^H$. This is possible, since we assumed that v_0 is in the range of E_0. Then

$$z_1^H E_j = z_1^H E_0 (I - q_0 v_0^H) ... (I - q_{j-1} v_{j-1}^H) = 0, \quad j = 1, ..., k+1.$$

Consequently,

$$\begin{aligned} z_1^H P_j &= z_1^H E_j E_{j-1} ... E_1 = 0, \quad j = 1, ..., k+1, \\ z_1^H r_j &= z_1^H P_j r_0 = 0, \quad j = 1, ..., k+1, \\ z_1^H \tilde{q}_j &= z_1^H r_j - z_1^H r_{j+1} = 0, \quad j = 1, ..., k, \\ z_1^H q_j &= z_1^H \tilde{q}_j - v_j^H r_j z_1^H E_j q_j = 0, \quad j = 1, ..., k. \end{aligned}$$

Now, assume the vectors $z_1, ..., z_{i-1}$ exist and are linearly independent for $i \leq k$, and

$$\begin{aligned} z_{i-1}^H r_j &= z_{i-1}^H P_j = 0, \quad j = i-1, ..., k+1, \\ z_{i-1}^H \tilde{q}_j &= z_{i-1}^H q_j = 0, \quad j = i-1, ..., k. \end{aligned}$$

We define $z_i^H E_i = z_{i-1}^H$. Since $z_{i-1}^H q_{i-1} = 0$ and q_{i-1} is the only null vector of E_i due to the assumption $v_i^H q_{i-1} \neq 0$ (see Lemma 4.1), z_i exists. Also, we have the following equation

$$\begin{aligned} z_i^H E_j &= z_i^H E_i (I - q_i v_i^H) ... (I - q_{j-1} v_{j-1}^H) \\ &= z_{i-1}^H (I - q_i v_i^H) ... (I - q_{j-1} v_{j-1}^H) \\ &= z_{i-1}^H, \end{aligned}$$

since $z_{i-1}^H q_l = 0, \ l = i, ..., j-1$, due to the assumption.

Assume the vectors $z_1, ..., z_i$ are linearly dependent, i.e.,

$$z_i = \sum_{j=1}^{i-1} \gamma_j z_j, \tag{4.8}$$

since $z_1, ..., z_{i-1}$ are linearly independent due to the assumption. Consequently,

$$0 = (z_i - \sum_{j=1}^{i-1} \gamma_j z_j)^H E_i$$

$$
\begin{aligned}
&= z_i^H E_i - \sum_{j=1}^{i-1} \gamma_j z_j^H E_i \\
&= z_{i-1}^H - \sum_{j=2}^{i-1} \gamma_j z_{j-1}^H \\
&\neq 0.
\end{aligned}
$$

This is a contradiction, and the vectors $z_1, ..., z_i$ must therefore be linearly independent.

We have, for $j = i, ..., k+1$,

$$
\begin{aligned}
z_i^H P_j &= z_i^H E_j E_{j-1} ... E_1 \\
&= z_{i-1}^H E_{j-1} ... E_1 \\
&= z_{i-1}^H P_{j-1} \\
&= 0.
\end{aligned}
$$

This leads to

$$
\begin{aligned}
z_i^H r_j &= z_i^H P_j r_0 = 0, \quad j = i, ..., k+1, \\
z_i^H \tilde{q}_j &= z_i^H r_j - z_i^H r_{j+1} = 0, \quad j = i, ..., k, \\
z_i^H q_j &= z_i^H \tilde{q}_j - v_j^H r_j z_i^H E_j q_j \\
&= -v_i^H r_j z_{i-1}^H q_j = 0, \quad j = i, ..., k+1.
\end{aligned}
$$

□

Next, consider the case $v_k^H q_{k-1} = 0$. As Lemma 4.4 shows, for this case the rank of E_{k+1} can decrease compared to E_k. It is this property that Broyden's method with projected updates uses in order to achieve termination within n steps instead of the $2n$ steps required for Broyden's method with general v_k.

Lemma 4.4 *If $q_k \neq 0$, $v_k^H q_{k-1} = 0$, and rank(E_k) $= n-1$, then, rank(E_{k+1}) $= n-2$, and span$\{q_k, q_{k-1}\} = N(E_{k+1})$.*

Proof:

We know that $E_{k+1} q_k = 0$. Consequently, $q_k \in N(E_{k+1})$. In order to show the reduction of the rank of E_{k+1}, we need to find a vector $x \neq \lambda q_k$ for which $E_{k+1} x = 0$. Furthermore,

$$
E_{k+1} x = E_k (I - q_k v_k^H) x = 0, \tag{4.9}
$$

which is equivalent to

$$(I - q_k v_k^H)x = \mu q_{k-1}. \tag{4.10}$$

Since $v_k^H q_{k-1} = 0$, for all $y \in R(I - q_k v_k^H)$ $v_k^H y = 0$, $dim(R(I - q_k v_k^H)) = n-1$, and $span\{v_k\} \oplus R(I - q_k v_k^H) = \mathcal{C}^n$, x exists. Now,

$$x = \mu q_{k-1} + v_k^H x q_k, \tag{4.11}$$

which proves the second statement. □

Furthermore, the rank of E_k cannot increase, as k increases.

Lemma 4.5

$$rank(E_k) - 1 \leq rank(E_{k+1}) \leq rank(E_k), \tag{4.12}$$

where

$$rank(E_{k+1}) = rank(E_k) - 1 \tag{4.13}$$

if and only if there exists a vector y with $y^H E_k = v_k^H$.

Proof:

$$rank(E_k) = rank(E_k^H) = m. \tag{4.14}$$

Consequently, there exist $z_i, i = 1, ..., n-m$, with $z_i^H E_{k+1} = 0$. Now,

$$z_i^H E_{k+1} = z_i^H E_k (I - q_k v_k^H) = 0, \quad i = 1, ..., n-m. \tag{4.15}$$

In order for $rank(E_{k+1})$ to be smaller than $rank(E_k)$, we need z_{n-m+1} with $z_{n-m+1}^H E_k = v_k^H$.
□

The finite termination of the family EN-like methods can now be proved. Note that termination occurs within n steps. This satisfies the requirement that the convergence rates for the EN-like methods must be twice as fast as the corresponding Broyden methods.

Theorem 4.2 *For $f_k^H q_k \neq 0$, the EN-like method converges within at most n steps.*

Proof: Using the results of Lemma 4.5, we can redefine a sequence of z_i without the restriction $v_j^H q_{j-1} \neq 0$ of Lemma 4.3:

For $v_i^H q_{i-1} \neq 0$, we choose $z_i^H E_i = z_{i-1}$ as in the proof for Lemma 4.3. If $v_i^H q_{i-1} = 0$ and there exists a y with $y^H E_i = v_i^H$, we set $z_i = y$. If no such y exists, we choose z_i with $z_i^H E_i = 0$ and z_i linearly independent of $z_1, ..., z_{i-1}$. Such a z_i exists, since $rank(E_i)$ is smaller than $rank(E_{i-1})$.

Now, we can show, as in Lemma 4.3, that

$$z_i^H r_j = 0, \quad j = i, ..., k+1,$$

for $k \leq n-1$, from which we conclude finite termination within at most n steps. $\square$

4.2 Local Convergence

The finite termination property is of more theoretical than practical interest. Therefore, it is important to examine the convergence behavior of the methods. We first summarize, in Section 4.2.1, the convergence results for the Broyden methods found in the literature. In order to improve the results for Broyden methods and to derive similar results for the EN-like methods, we need to prove several basic lemmas concerning the behavior of the residual, r_k, the error vector, e_k, and the error matrices, E_k and $\tilde{E}_k$. These are developed in Section 4.2.2. Finally, the new convergence results for members of the families of Broyden and EN-like methods are derived in Section 4.2.3.

4.2.1 Previous Results for Broyden's Method

The convergence of the two best known members of the family of Broyden's methods, BBM and GBM, can be characterized in terms of the reduction of the residual and error vectors respectively by the following theorems from the literature [8]. Our goal is to demonstrate that the contraction coefficients in the correponding expressions for the EN-like methods are approximately squared, and therefore the desired behavior of a convergence rate double that of Broyden's methods can be expected.

Theorem 4.3 *Assume $\alpha_k = 1$ or $\alpha_k = q_k^H r_k / q_k^H q_k$ for BBM (Broyden's method with $f_k = q_k$). Assume additionally that $\|E_0\| \leq \delta < 1$.*

The following inequality then holds:

$$\|r_{k+1}\| \leq \delta \|r_k\|, \tag{4.16}$$

and the method converges.

Additionally, the convergence is q-superlinear; i.e., there exists a sequence c_k with $0 \leq c_k < 1$ and $lim_{k\to\infty} c_k = 0$, so that

$$\|r_{k+1}\| \leq c_k \|r_k\|. \tag{4.17}$$

Proof: See [8]. □

Theorem 4.4 *Assume one of the following choices of α_k:*

(i) $\alpha_k = 1$,

(ii) $\alpha_k = p_k^H p_k / p_k^H H_k q_k$,

(iii) $\alpha_k = p_k^H H_k q_k / q_k^H H_k^H H_k q_k$,

for GBM (Broyden's method with $f_k = H_k^H p_k$). Assume additionally that $H_k A$ nonsingular, and $\|\tilde{E}_0\| \leq \delta$, where $\delta < \frac{1}{2}$ for $\alpha_k = 1$, $\delta < \frac{1}{3}$ for the other choices for α_k.

The following inequalities then hold:

$$\|e_{k+1}\| \leq \frac{\delta}{1-\delta} \|e_k\| \tag{4.18}$$

for $\alpha_k = 1$, and

$$\|e_{k+1}\| \leq \frac{2\delta}{1-\delta} \|e_k\| \tag{4.19}$$

for α_k choices (ii) and (iii), and the method converges.

Additionally, the convergence is q-superlinear; i.e., there exists a sequence c_k with $0 \leq c_k < 1$ and $\lim_{k\to\infty} c_k = 0$, so that

$$\|e_{k+1}\| \leq c_k \|e_k\|. \tag{4.20}$$

Proof: See [8]. □

4.2.2 Basic Relationships and Lemmas

In order to investigate the convergence for the EN-like methods and certain Broyden methods, we must develop some basic lemmas. The progession of the lemmas has the following general scheme. The analyses of the error vector e_k and the residual vector r_k require knowledge of the behavior of $\tilde{E}_k$ and E_k, respectively. Lemmas 4.9 and 4.10 relate the norms of the two error matrices to the vectors that define the iterations of the two families, f_k and q_k. These relationships and specific choices of the vectors are used to demonstrate in Lemmas 4.11 and 4.12 that $||E_k||$ and $||\tilde{E}_k||$ are nonincreasing functions of k. Lemmas 4.13 and 4.14 combine the previous results to characterize the reduction in the norm of the residual for certain Broyden methods and EN-like methods, respectively. Lemmas 4.16 through 4.19 establish results that are needed to extend the convergence results for Broyden's methods. The behavior of the parameter α_k that is used explicitly in Broyden's method and implicitly in the EN-like method is considered in Lemmas 4.20 and 4.21.

Successive residuals of Broyden's method and EN-like methods have simple relationships via the error matrix E_k. These are summarized in Lemmas 4.6 and 4.7, respectively.

Lemma 4.6 *For Broyden's method,*

$$r_{k+1} = [(1-\alpha_k)I + \alpha_k E_k]r_k, \tag{4.21}$$

which for $\alpha_k = 1$ *reduces to*

$$r_{k+1} = E_k r_k. \tag{4.22}$$

Proof: See [8]. □

Lemma 4.7 *For the EN-like method,*

$$r_{k+1} = E_{k+1}E_k r_k. \tag{4.23}$$

Proof: Using $E_{k+1}q_k = 0$,

$$\begin{aligned} r_{k+1} &= E_{k+1}r_k \\ &= E_{k+1}(r_k - q_k) \\ &= E_{k+1}E_k r_k \end{aligned}$$

□

The lemmas above and the fact that the behavior of the error vector, e_k, for Broyden and EN-like methods is related to the matrix $\tilde{E}_k$ indicate the necessity of investigating the norms $||E_k||$ and $||\tilde{E}_k||$. Clearly, their behavior must be related to the particular member of the Broyden or EN-like families under consideration. Selecting a particular method involves choosing a form of f_k. Lemmas 4.9 and 4.10 relate the norms of the two error matrices to the choice of f_k and the auxiliary vector q_k.

Lemma 4.8 *For $x, y \in \mathcal{C}^n$ with $x^H y = 1$,*

$$\|I - xy^H\| = \|x\|\|y\|. \tag{4.24}$$

Proof: See [5]. □

We can now prove the following two lemmas.

Lemma 4.9

$$\|E_{k+1}\| \leq \|E_k\| \frac{\|f_k\|\|q_k\|}{|f_k^H q_k|} \tag{4.25}$$

Proof: Applying Lemma 4.8 to

$$\begin{aligned}
E_{k+1} &= I - AH_{k+1} \\
&= I - AH_k - A\frac{(p_k - H_k q_k)f_k^H}{f_k^H q_k} \\
&= E_k(I - \frac{q_k f_k^H}{f_k^H q_k})
\end{aligned}$$

leads to the assertion. □

Lemma 4.10

$$\|\tilde{E}_{k+1}\| \leq \|\tilde{E}_k\| \frac{\|H_k^{-H} f_k\|\|p_k\|}{|f_k^H H_k^{-1} p_k|} \tag{4.26}$$

Proof: Apply Lemma 4.8 to

$$\begin{aligned}
\tilde{E}_{k+1} &= I - A^{-1}H_{k+1}^{-1} \\
&= I - A^{-1}H_k^{-1} - A^{-1}\frac{(q_k - H_k^{-1}p_k)f_k^H H_k^{-1}}{f_k^H H_k^{-1} p_k} \\
&= \tilde{E}_k(I - \frac{p_k f_k^H H_k^{-1}}{f_k^H H_k^{-1} p_k}),
\end{aligned}$$

and the desired result follows. □

Clearly, this shows that if we choose $f_k = \gamma q_k$ (or $f_k = \gamma H_k^H p_k$) for some $\gamma \neq 0$, $\|E_k\|$ (or $\|\tilde{E}_k\|$) does not increase with increasing k. This choice of f_k yields exactly BBM and BEN (or GBM and GEN). There are other choices of f_k that also lead to nonincreasing norms. One is $f_k = E_k^H E_k q_k$, which yields EN and GCR-BM, and leads to nonincreasing $\|E_k\|$ as shown below. Using the same concept for nonincreasing $\|\tilde{E}_k\|$ leads to $f_k = H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, which yields the dual EN and dual GCR-BM. For this particular choice, we have

$$\begin{aligned} H_{k+1} &= H_k + \frac{(p_k - H_k q_k) p_k^H \tilde{E}_k^H \tilde{E}_k H_k}{p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k} \\ &= H_k + \frac{(p_k - H_k q_k)(p_k - e_k)^H (H_k - A^{-1})}{(p_k - A^{-1} r_k)^H (H_k q_k - p_k)} \\ &= H_k - \frac{(p_k - H_k q_k)(x_k + p_k - x)^H (H_k - A^{-1})}{(x_k + p_k - x)^H (p_k - H_k q_k)}. \end{aligned}$$

For the dual EN, this leads to the k-th direction vector

$$H_{k+1} r_k = p_k - \frac{\|\tilde{e}_k\|^2}{(\tilde{e}_k)^H (p_k - H_k q_k)} (p_k - H_k q_k), \tag{4.27}$$

which includes the error $\tilde{e}_k = e_k + p_k$ of the interim solution $x_k + p_k$. For the dual GCR-BM, one obtains the $(k+1)$-th direction vector

$$H_{k+1} r_{k+1} = \frac{\|e_{k+1}\|^2}{e_{k+1}^H (p_k - H_k q_k)} (p_k - H_k q_k). \tag{4.28}$$

As a result, the dual EN and dual GCR-BM are only of theoretical interest due to the occurrence of the actual errors in the evaluation of the direction vectors. Nevertheless, we include them in our theoretical results for completeness.

The choices of f_k that yield PBM, PEN, and the dual PBM and PEN also lead to nonincreasing norms of the error matrices.

Lemmas 4.11 and 4.12 summarize these results.

Lemma 4.11 *If $f_k = \gamma q_k$, $f_k = \gamma E_k^H E_k q_k$, or $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ (where $\tilde{q}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $q_i, i = 0, ..., k-1$), then*

$$\|E_{k+1}\| \leq \|E_k\|. \tag{4.29}$$

Proof: For $f_k = \gamma q_k$, (4.29) follows immediately from Lemma 4.9.

For $f_k = \gamma E_k^H E_k q_k$, (4.29) follows from Theorem 2.1, since $\|E_k\|$ equals its maximal singular value.

The proof of (4.29) for $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ is similar to that of (4.32) for $f_k = \gamma H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$ and can be found in [23].

It is easy to show, using the orthonormality of the $\tilde{q}_i$ that

$$f_k^H q_k = f_k^H f_k.$$

Furthermore, using Lemma 2.1,

$$E_k q_i = 0, \quad 0 \leq i < k,$$

and consequently also

$$E_k \tilde{q}_i = 0, \quad 0 \leq i < k,$$

which leads to

$$E_k f_k = E_k q_k. \tag{4.30}$$

Therefore,

$$E_{k+1} = E_k(I - \frac{f_k f_k^H}{f_k^H f_k}), \tag{4.31}$$

from which (4.29) follows. □

Lemma 4.12 *If $f_k = \gamma H_k^H p_k$, $f_k = \gamma H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, or $f_k = \gamma H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$ (where $\tilde{p}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $p_i, i = 0, ..., k-1$), then*

$$\|\tilde{E}_{k+1}\| \leq \|\tilde{E}_k\|. \tag{4.32}$$

Proof: A proof similar to that of Lemma 4.11 is possible. □

Lemmas 4.13 and 4.14 finish the development of the basic results needed to bound the residuals for certain Broyden and EN-like methods in terms of the error matrix E_k. Similar results relating e_k and $\tilde{E}_k$ are not needed for later proofs since their effects are developed implicitly.

Lemma 4.13 *If for BBM ($f_k = \gamma q_k$), GCR-BM ($f_k = \gamma E_k^H E_k q_k$), or PBM ($f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$), we use $\alpha_k = 1$ or $\alpha_k = q_k^H r_k / \|q_k\|^2$, then*

$$\|r_{k+1}\| \leq \|E_0\| \|r_k\| \leq \|E_0\|^{k+1} \|r_0\|. \tag{4.33}$$

Proof: The first part of the inequality follows trivially for $\alpha_k = 1$ with Lemma 4.6 and Lemma 4.11. For $\alpha_k = q_k^H r_k / \|q_k\|^2$, the inequality is obtained, according to [8], through

$$\begin{aligned} r_{k+1} &= r_k - \alpha_k q_k \\ &= \left(I - \frac{q_k q_k^H}{q_k^H q_k}\right)(r_k - q_k) \\ &= \left(I - \frac{q_k q_k^H}{q_k^H q_k}\right) E_k r_k. \end{aligned}$$

The second part follows from the first part using induction. □

Lemma 4.14 *For BEN ($f_k = \gamma q_k$), EN ($f_k = \gamma E_k^H E_k q_k$), or PEN ($f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$):*

$$\|r_{k+1}\| \leq \|E_0\|^2 \|r_k\| \leq \|E_0\|^{2k+2} \|r_0\|. \tag{4.34}$$

Proof: The first part of the inequality follows trivially from Lemma 4.7 and Lemma 4.11. The second part follows from the first part using induction. □

The results above all assumed the use of the 2-norm. Similar results can be developed for the Frobenius norm ($\|.\|_F$) of E_k and $\tilde{E}_k$ using a lemma from [6].

Lemma 4.15 *If A is an $n \times n$ matrix and x, y are two vectors of order n, then*

$$\|A + xy^H\|_F = \|A\|_F^2 + 2y^H A^H x + \|x\|^2 \|y\|^2.$$

Proof: See [6]. □

We can now prove the following lemmas. These lemmas are not only interesting in and of themselves, but their proofs also require the development of identities that are required later to prove the convergence results for certain Broyden methods.

Lemma 4.16 *Assume $f_k = \gamma q_k$, $f_k = \gamma E_k^H E_k q_k$, or $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ (where $\tilde{q}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $q_i, i = 0, ..., k-1$), then*

$$\|E_{k+1}\|_F \leq \|E_k\|_F. \tag{4.35}$$

Proof: Using Lemma 4.15, we get for $f_k = \gamma q_k$,

$$\|E_{k+1}\|_F^2 = \|E_k\|_F^2 - \frac{\|E_k q_k\|^2}{\|q_k\|^2}, \tag{4.36}$$

for $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$, recalling (4.31),

$$\|E_{k+1}\|_F^2 \quad = \quad \|E_k\|_F^2 - \frac{\|E_k f_k\|^2}{\|f_k\|^2}, \tag{4.37}$$

and for $f_k = \gamma E_k^H E_k q_k$

$$\|E_{k+1}\|_F^2 = \|E_k\|_F^2 - \frac{\|E_k^H E_k q_k\|^2}{\|E_k q_k\|^2}, \tag{4.38}$$

from which (4.35) follows. Note that for $f_k = \gamma E_k^H E_k q_k$ (4.35) also follows immediately from (v) of Theorem 2.1, since the Frobenius norm of E_k is just the square root of the sum of the squares of its singular values. □

Lemma 4.17 *Assume $f_k = \gamma H_k^H p_k$, $f_k = \gamma H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, or $f_k = \gamma H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$ (where $\tilde{p}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $p_i, i = 0, ..., k-1$), then*

$$\|\tilde{E}_{k+1}\|_F \leq \|\tilde{E}_k\|_F. \tag{4.39}$$

Proof: This lemma can be shown in an equivalent way as Lemma 4.16. □

The finite termination proofs for the family of Broyden methods in the literature all assume that $\alpha_k = 1$. In order to investigate the convergence of members of the Broyden family for which $\alpha_k \neq 1$, we must develop results bounding the asymptotic behavior of the relative effect of applying E_k and $\tilde{E}_k$ to the vector q_k as k increases. The results are summarized in Lemmas 4.18 and 4.19.

Lemma 4.18 *If $f_k = \gamma q_k$, $f_k = \gamma E_k^H E_k q_k$, or $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ (where $\tilde{q}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $q_i, i = 0, ..., k-1$), and $\|q_k\| \neq 0$, then*

$$\lim_{k \to \infty} \frac{\|E_k q_k\|}{\|q_k\|} = 0, \tag{4.40}$$

and for $f_k = \gamma E_k^H E_k q_k$, $\|E_k q_k\| \neq 0$, additionally,

$$\lim_{k \to \infty} \frac{\|E_k^H E_k q_k\|}{\|E_k q_k\|} = 0. \tag{4.41}$$

Proof: The proof of (4.46) for $f_k = \gamma H_k^H p_k$ can be found in [30]. Using the same strategy, one can prove the rest of the lemma.

Let us first consider $f_k = \gamma q_k$. Adding Equation (4.36) for $j \geq 0$, we obtain

$$\sum_{j=0}^{\infty} \frac{\|E_j q_j\|^2}{\|q_j\|^2} \leq \|E_0\|_F^2. \tag{4.42}$$

Consequently

$$\lim_{k\to\infty} \frac{||E_k q_k||}{||q_k||} = 0. \tag{4.43}$$

Now, consider $f_k = \gamma E_k^H E_k q_k$. As above, we conclude

$$\lim_{k\to\infty} \frac{||E_k^H E_k q_k||}{||E_k q_k||} = 0. \tag{4.44}$$

As shown in [13],

$$\begin{aligned}
\frac{||E_k^H E_k q_k||}{||E_k q_k||} &= \sup_{||x||=1} x^H E_k^H \frac{E_k q_k}{||E_k q_l||} \\
&\geq \left(\frac{q_k}{||q_k||}\right)^H E_k^H \frac{E_k q_k}{||E_k q_k||} \\
&= \frac{||E_k q_k||}{||q_k||}.
\end{aligned}$$

Consequently, (4.40) follows also for $f_k = \gamma E_k^H E_k q_k$.

Using the same strategy as above, one can prove for $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ that

$$\lim_{k\to\infty} \frac{||E_k f_k||}{||f_k||} = 0. \tag{4.45}$$

Since

$$||f_k|| = ||(I - \sum_{i=0}^{k-1} \tilde{q}_i \tilde{q}_i^H) q_k|| \leq ||q_k||$$

and due to (4.30)

$$\lim_{k\to\infty} \frac{||E_k q_k||}{||q_k||} \leq \lim_{k\to\infty} \frac{||E_k f_k||}{||f_k||} = 0.$$

□

Lemma 4.19 *Assume* $f_k = \gamma H_k^H p_k$, $f_k = \gamma H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, *or* $f_k = \gamma H_k^H (p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$ *(where* $\tilde{p}_i, i = 0, ..., k-1$ *are an orthonormal basis for the space spanned by* $p_i, i = 0, ..., k-1$*), and* $||p_k|| \neq 0$*, then*

$$\lim_{k\to\infty} \frac{||\tilde{E}_k p_k||}{||p_k||} = 0, \tag{4.46}$$

and for $f_k = \gamma H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, $||\tilde{E}_k p_k|| \neq 0$ *additionally,*

$$\lim_{k\to\infty} \frac{||\tilde{E}_k^H \tilde{E}_k p_k||}{||\tilde{E}_k p_k||} = 0. \tag{4.47}$$

Proof: The proof is similar to the one for Lemma 4.18. □

Particular members of the family of Broyden's methods require, in addition to f_k, the specification of the form of the parameter α_k. Also recall that, while members of the family of EN-like methods are not determined by the choice of α_k, it is possible to write them in a form where an α_k appears that is determined by the choice of f_k (see Lemma 2.3). Understanding the behavior of α_k is therefore of use in preparing the necessary foundation for the characterization of the convergence of members of the EN-like and Broyden families. This is done in Lemmas 4.20 and 4.21.

Lemma 4.20 *Assume $\|E_k\| \le \delta < 1$, and choose*

(i) $\alpha_k = q_k^H r_k / q_k^H q_k$, *or*

(ii) $\alpha_k = q_k^H E_k^H E_k r_k / q_k^H E_k^H E_k q_k$.

Then,

$$|1 - \alpha_k| \le \frac{\delta}{1 - \delta}. \tag{4.48}$$

Furthermore, assume choice (i) for α_k and $f_k = \gamma q_k$, $f_k = \gamma E_k^H E_k q_k$, or $f_k = \gamma(q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i)$ (where $\tilde{q}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $q_i, i = 0, ..., k-1$), or assume choice (ii) for α_k and $f_k = \gamma E_k^H E_k q_k$.

Then,

$$\lim_{k \to \infty} |1 - \alpha_k| = 0. \tag{4.49}$$

Proof: Consider $\|E_k\| \le \delta < 1$.

The proof for choice (i) is given in [8].

For choice (ii)

$$\begin{aligned} |1 - \alpha_k| &= \frac{|q_k^H E_k^H E_k (q_k - r_k)|}{\|E_k q_k\|^2} \\ &= \frac{|q_k^H E_k^H E_k E_k r_k|}{\|E_k q_k\|^2} \\ &\le \frac{1}{1-\delta} \frac{\|E_k^H E_k q_k\|}{\|E_k q_k\|} \\ &\le \frac{\|E_k^H\|}{1-\delta} \\ &\le \frac{\delta}{1-\delta}, \end{aligned}$$

which together with Lemma 4.18 proves the assertions. □

Lemma 4.21 *Assume $\|\tilde{E}_k\| \le \delta < 1$, and one of the following choices for α_k:*

(i) $\alpha_k = p_k^H p_k / p_k^H H_k q_k$,

(ii) $\alpha_k = p_k^H H_k q_k / \|H_k q_k\|^2$,

(iii) $\alpha_k = p_k^H \tilde{E}_k^H \tilde{E}_k p_k / p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k$, *or*

(iv) $\alpha_k = p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k / \|\tilde{E}_k H_k q_k\|^2$.

Then,

$$|1 - \alpha_k| \le \delta, \tag{4.50}$$

Furthermore, assume choice (i) or choice (ii) for α_k and $f_k = \gamma H_k^H p_k$, $f_k = \gamma H_k^H \tilde{E}_k^H \tilde{E}_k p_k$, or $f_k = \gamma H_k^H (p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$ (where $\tilde{p}_i, i = 0, ..., k-1$ are an orthonormal basis for the space spanned by $p_i, i = 0, ..., k-1$), and $\|p_k\| \ne 0$.

Then,

$$\lim_{k \to \infty} |1 - \alpha_k| = 0. \tag{4.51}$$

Proof: Consider $\|\tilde{E}_k\| \le \delta < 1$.

A proof of (4.50) for choice (i) is given in [8]. But, since it is not completely correct, we repeat it here with the necessary correction. Defining

$$\epsilon := \frac{\|\tilde{E}_k H_k q_k\|}{\|H_k q_k\|} \tag{4.52}$$

and using

$$q_k^H H_k^H \tilde{E}_k H_k q_k = c \|\tilde{E}_k H_k q_k\| \|H_k q_k\| \tag{4.53}$$

with $|c| \le 1$, one obtains for choice (i):

$$\begin{aligned}
|1 - \alpha_k| &= \frac{|p_k^H (H_k q_k - p_k)|}{p_k^H H_k q_k} \\
&= \frac{|p_k^H \tilde{E}_k H_k q_k|}{|p_k^H H_k q_k|} \\
&= \frac{|q_k^H H_k^H (I - \tilde{E}_k)^H \tilde{E}_k H_k q_k|}{|q_k^H H_k^H (I - \tilde{E}_k)^H H_k q_k|} \\
&= \frac{|q_k^H H_k^H \tilde{E}_k H_k q_k|}{\|H_k q_k\|^2} \frac{|1 - \|\tilde{E}_k H_k q_k\|^2 / q_k^H H_k^H \tilde{E}_k H_k q_k|}{|1 - q_k^H H_k^H \tilde{E}_k H_k q_k / \|H_k q_k\|^2|} \\
&= \frac{\|\tilde{E}_k H_k q_k\|}{\|H_k q_k\|} |c| \frac{|1 - \epsilon / c|}{|1 - c\epsilon|} \\
&\le \frac{\|\tilde{E}_k H_k q_k\|}{\|H_k q_k\|} \le \delta,
\end{aligned}$$

since

$$
\begin{aligned}
|c|\frac{|1-\epsilon/c|}{|1-c\epsilon|} &= \frac{|c-\epsilon|}{|1-c\epsilon|} \\
&= \left(\frac{|c|^2+\epsilon^2-2\epsilon Real(c)}{1+|c|^2\epsilon^2-2\epsilon Real(c)}\right)^{\frac{1}{2}} \leq 1.
\end{aligned}
$$

The proof of (4.50) for choice (ii) is given in [8].

Now, we prove (4.51):

$$
\begin{aligned}
\|\tilde{E}_k H_k q_k\| &= \|\tilde{E}_k(I-\tilde{E}_k)^{-1}p_k\| \\
&= \|(1-\tilde{E}_k)^{-1}\tilde{E}_k p_k\| \\
&\leq \frac{1}{1-\|\tilde{E}_k\|}\|\tilde{E}_k p_k\|
\end{aligned}
$$

and

$$
\begin{aligned}
\|p_k\| &= \|(I-\tilde{E}_k)H_k q_k\| \\
&\leq (1+\|\tilde{E}_k\|)\|H_k q_k\|,
\end{aligned}
$$

which leads to

$$
\lim_{k\to\infty}\frac{\|\tilde{E}_k H_k q_k\|}{\|H_k q_k\|} \leq \lim_{k\to\infty}\frac{1+\delta}{1-\delta}\frac{\|\tilde{E}_k p_k\|}{\|p_k\|} = 0. \tag{4.54}
$$

Together with Lemma 4.19, this proves (4.51) for choices (i) and (ii).

All that is left to prove is (4.50) for choices (iii) and (iv). Consider first choice (iii). As for choice (i), one shows

$$
|1-\alpha_k| \leq \frac{\|\tilde{E}_k^2 H_k q_k\|}{\|\tilde{E}_k H_k q_k\|} \leq \delta.
$$

Finally, consider choice (iv):

$$
\begin{aligned}
|1-\alpha_k| &= \frac{|q_k^H H_k^H \tilde{E}_k^H \tilde{E}_k(H_k q_k - p_k)|}{\|\tilde{E}_k H_k q_k\|^2} \\
&= \frac{|q_k^H H_k^H \tilde{E}_k^H \tilde{E}_k^2 H_k q_k|}{\|\tilde{E}_k H_k q_k\|^2} \\
&\leq \frac{\|\tilde{E}_k^2 H_k q_k\|}{\|\tilde{E}_k H_k q_k\|} \\
&\leq \|\tilde{E}_k\| \leq \delta.
\end{aligned}
$$

□

4.2.3 New Convergence Results for the Broyden and EN-like Families

We can now prove the following results for PBM and GCR-BM.

Theorem 4.5 *Assume* $\alpha_k = 1$ *or* $\alpha_k = q_k^H r_k / q_k^H q_k$ *for GCR-BM* ($f_k = E_k^H E_k q_k$)*, or PBM* ($f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$)*. Assume additionally that* $\|E_0\| \le \delta < 1$.

The following inequality then holds:

$$\|r_{k+1}\| \le \delta \|r_k\| \tag{4.55}$$

and the methods converge.

Assume $\alpha_k = f_k^H r_k / f_k^H q_k$ *for GCR-BM. Assume additionally that* $\|E_0\| \le \delta < \sqrt{2} - 1$.

Then,

$$\|r_{k+1}\| \le \delta(\delta + 2)\|r_k\|, \tag{4.56}$$

and the method converges.

Additionally, for all choices considered above, the convergence is q-superlinear, i.e. there exists a sequence c_k *with* $0 \le c_k < 1$ *and* $lim_{k\to\infty} c_k = 0$*, so that*

$$\|r_{k+1}\| \le c_k \|r_k\|. \tag{4.57}$$

Proof: (4.55) and the convergence of the methods follow trivially from Lemma 4.13.

Let us now prove (4.56). For $\alpha_k = f_k^H r_k / f_k^H q_k = q_k^H E_k^H E_k r_k / q_k^H E_k^H E_k q_k$:

$$\begin{aligned} \|r_{k+1}\| &= \|r_k - \alpha_k q_k\| \\ &= \|E_k r_k + (1 - \alpha_k) q_k\| \\ &\le \|E_k r_k\| + |1 - \alpha_k| \|(I - E_k) r_k\| \\ &\le (\delta + |1 - \alpha_k|(1 + \delta))\|r_k\|. \end{aligned}$$

The combination of these inequalities results in (4.56).

Let us now show q-superlinear convergence. For $\alpha_k = 1$, q-superlinear convergence follows from the finite termination property, which leads to $\|r_k\| = 0$ for $k > k_0$ for some $k_0 > 0$. For $\alpha_k = q_k^H r_k / q_k^H q_k$, the proof for GCR-BM and PBM is similar to that for GBM in [8]. Since

$$r_k = (AH_k)^{-1} q_k = (I - E_k)^{-1} q_k, \tag{4.58}$$

one can show for both choices of α_k, along the lines of the proof for Lemma 4.20,

$$\lim_{k\to\infty} \frac{\|r_{k+1}\|}{\|r_k\|} = \lim_{k\to\infty} \frac{1+\delta}{1-\delta}\frac{\|E_k q_k\|}{\|q_k\|} = 0. \tag{4.59}$$

For $\alpha_k = q_k^H E_k^H E_k r_k / q_k^H E_k^H E_k q_k$:

$$\lim_{k\to\infty} \frac{\|r_{k+1}\|}{\|r_k\|} = \lim_{k\to\infty} \left(\frac{1+\delta}{1-\delta}\frac{\|E_k q_k\|}{\|q_k\|} + (1+\delta)|1-\alpha_k|\right) = 0 \tag{4.60}$$

according to Lemma 4.18 and Lemma 4.20.

□

There is a similar theorem for GBM, dual PBM, and dual GCR-BM. For its proof we need the *Perturbation Lemma* from [33].

Lemma 4.22 (Perturbation Lemma) *Assume that A and C are matrices of order n, A is nonsingular with $\|A^{-1}\| \le \alpha$, $\|A - C\| \le \beta$, and $\alpha\beta < 1$.*

C is then also nonsingular, and

$$\|C^{-1}\| \le \frac{\alpha}{1-\alpha\beta}. \tag{4.61}$$

Proof: See [33]. □

The following theorem overlaps with Theorem 4.4, which states the results for GBM with the first three choices for α_k. We introduce two new choices for α_k and consider also the dual PBM and the dual GCR-BM.

Theorem 4.6 *Assume one of the following choices of α_k:*

(i) $\alpha_k = 1$,

(ii) $\alpha_k = p_k^H p_k / p_k^H H_k q_k$,

(iii) $\alpha_k = p_k^H H_k q_k / \|H_k q_k\|^2$,

(iv) $\alpha_k = \|\tilde{E}_k p_k\|^2 / p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k$, *or*

(v) $\alpha_k = p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k / \|\tilde{E}_k H_k q_k\|^2$

for GBM ($f_k = H_k^H p_k$)*, the dual GCR-BM* ($f_k = H_k^H \tilde{E}_k^H \tilde{E}_k p_k$)*, or the dual PBM* ($f_k = H_k^H(p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$)*. Assume additionally that* $H_k A$ *nonsingular, and* $\|\tilde{E}_0\| \leq \delta$*, where* $\delta < \frac{1}{2}$ *for* $\alpha_k = 1$*,* $\delta < \frac{1}{3}$ *for the other choices for* α_k*.*

The following inequalities then hold:

$$\|e_{k+1}\| \leq \frac{\delta}{1-\delta}\|e_k\| \tag{4.62}$$

for $\alpha_k = 1$*,*

$$\|e_{k+1}\| \leq \frac{2\delta}{1-\delta}\|e_k\| \tag{4.63}$$

for α_k *choices (ii) to (v), and the methods converge.*

Additionally, the convergence is q-superlinear; i.e., there exists a sequence c_k *with* $0 \leq c_k < 1$ *and* $\lim_{k\to\infty} c_k = 0$*, so that*

$$\|e_{k+1}\| \leq c_k \|e_k\| \tag{4.64}$$

for choices (i) to (iii) for α_k*.*

Proof: According to Lemma 4.12 and the assumption $\|\tilde{E}_0\| \leq \delta$, it follows that $\|\tilde{E}_k\| \leq \delta$. According to Lemma 4.22, this leads to

$$\|H_k A\| \leq \frac{1}{1-\delta} \tag{4.65}$$

and

$$\|I - H_k A\| \leq \frac{\delta}{1-\delta}. \tag{4.66}$$

Additionally, using any of the choices above for α_k (except $\alpha_k = 1$), one obtains using Lemma 4.21

$$\begin{aligned} \|I - \alpha_k H_k A\| &= \|(1-\alpha_k)I + \alpha_k(I - H_k A)\| \\ &\leq |1-\alpha_k| + |\alpha_k|\|I - H_k A\| \\ &\leq \delta + (1+\delta)\frac{\delta}{1-\delta} \\ &= \frac{2\delta}{1-\delta}. \end{aligned} \tag{4.67}$$

Now, (see also [8]) we obtain

$$\begin{aligned} \|e_{k+1}\| &= \|e_k + \alpha_k p_k\| \\ &\leq \|I - \alpha_k H_k A\|\|e_k\|, \end{aligned}$$

which, together with Lemma 4.21, leads to (4.62) and (4.63).

Using $e_k = -(I - \tilde{E}_k)p_k$ leads to

$$\frac{\|e_k + \alpha_k p_k\|}{\|e_k\|} \leq \frac{1}{1-\delta}\left(|1-\alpha_k| + \frac{\|\tilde{E}_k p_k\|}{\|p_k\|}\right).$$

Consequently,

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} \leq \lim_{k\to\infty} \frac{1}{1-\delta}\left(|1-\alpha_k| + \frac{\|\tilde{E}_k p_k\|}{\|p_k\|}\right) = 0 \tag{4.68}$$

for α_k choices (ii) to (iii). For $\alpha_k = 1$, q-superlinear convergence follows from the finite termination property, which leads to $\|e_k\| = 0$ for $k > k_0$ for some $k_0 > 0$. □

We get the following results for BEN, EN, and PEN.

Theorem 4.7 *Assume for BEN ($f_k = q_k$), EN ($f_k = E_k^H E_k q_k$), or PEN ($f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$), that $\|E_0\| \leq \delta < 1$.*

The following inequality then holds:

$$\|r_{k+1}\| \leq \delta^2 \|r_k\|, \tag{4.69}$$

and the methods converge.

Additionally, the convergence is q-superlinear; i.e., there exists a sequence c_k with $0 \leq c_k < 1$ and $lim_{k\to\infty} c_k = 0$, so that

$$\|r_{k+1}\| \leq c_k \|r_k\|. \tag{4.70}$$

Proof: The first part of the theorem follows trivially with Lemma 4.14.

Q-superlinear convergence follows from the finite termination property, which leads to $\|r_k\| = 0$ for $k > k_0$ for some $k_0 > 0$. □

Similarly, we can prove the following results for GEN and the dual EN method.

Theorem 4.8 *Assume for GEN ($f_k = H_k^H p_k$), or the dual EN method ($f_k = H_k^H \tilde{E}_k^H \tilde{E}_k p_k$), that $H_k A$ is nonsingular, and $\|\tilde{E}_0\| \leq \delta$, where $\delta < \sqrt{2} - 1$.*

The following inequality then holds:

$$\|e_{k+1}\| \leq \frac{2\delta^2}{(1-\delta)^2} \|e_k\|, \tag{4.71}$$

and the methods converge.

Additionally, the convergence is q-superlinear; i.e., there exists a sequence c_k with $0 \leq c_k < 1$ and $\lim_{k\to\infty} c_k = 0$, so that

$$\|e_{k+1}\| \leq c_k \|e_k\|. \tag{4.72}$$

Proof: Using (4.66) and (4.67), one obtains with

$$\tau_k = \frac{f_k^H r_k}{f_k^H q_k},$$

which is $\tau_k = p_k^H p_k / p_k^H H_k q_k$ or $\tau_k = p_k^H \tilde{E}_k^H \tilde{E}_k p_k / p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k$,

$$\begin{aligned}
\|e_{k+1}\| &= \|e_k + p_k + \tau_k(p_k - H_k q_k)\| \\
&= \|(I - \tau_k H_k A)(e_k + p_k)\| \\
&\leq \|I - \tau_k H_k A\| \|I - H_k A\| \|e_k\| \\
&\leq \left(\frac{2\delta}{1-\delta} \cdot \frac{\delta}{1-\delta}\right) \|e_k\| \\
&= \frac{2\delta^2}{(1-\delta)^2} \|e_k\|.
\end{aligned}$$

Q-superlinear convergence follows from the finite termination property, which leads to $\|e_k\| = 0$ for $k > k_0$ for some $k_0 > 0$. □

These theorems show that, under the above assumptions, the upper limits of the convergence rate for the EN-like methods are approximately squared compared to those of the corresponding Broyden methods, a fact that is necessary for the overall complexity of the EN-like methods to be competitive and that is useful for the development of the adaptive methods (see Chapter 5).

There is another interesting implication from the estimates for $\|r_{k+1}\|$ in Lemma 4.13 and Lemma 4.14. In Theorem 4.5 and Theorem 4.7, we considered only the case for which these lemmas guarantee convergence for Broyden's method and the EN-like method. Let us now also consider the case for which $\|E_0\|$ is large; i.e., H_0 is a bad approximation of A^{-1}. Clearly, this can lead to a dramatic increase in $\|r_k\|$ for increasing k, which can be significantly worse for the EN-like method than for the corresponding Broyden method (see Figure 2.7). As we see in Chapters 5 and 6, the occurrence of such a drastic divergence can be of advantage in the context of PARASPAR. It is even preferred to the convergence behavior of SEN, BBM with $\alpha_k = q_k^H r_k / \|q_k\|^2$, and GCR, which guarantee a nonincreasing $\|r_k\|$.

The SEN method is not included in the previous convergence results, since formally it does not belong into the class of EN-like methods. We show in a later section that it converges for a real system, if the symmetric part of AH_0 is positive definite (see Corollary 4.1).

4.3 Convergence Behavior of Restarted Versions

In practice, restarted or truncated versions of the methods may have to be used in order to maintain efficiency with respect to operations and memory. Certainly, the finite termination property is lost for these methods. The local convergence behavior of the restarted versions can be explored using the results derived above. The convergence behavior for restarted BBM and GBM with the choices of α_k of Theorems 4.3 and 4.4 is considered in [8]. We investigate the convergence of restarted versions of PBM, GCR-BM, the dual PBM, and the dual GCR-BM for various α_k and add two choices for α_k to the results for restarted GBM of [8].

According to the definition of these variants, one restarts the method after $m+1$ steps with $H_{m+1}^{(r)} = H_0$. Consequently, if $f_k = q_k$, $f_k = {E_k^{(r)}}^H E_k^{(r)} q_k$, or $f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$,

$$E_{m+1}^{(r)} = E_0. \tag{4.73}$$

For these choices of f_k, the following inequalities can be deduced based on Lemma 4.11:

$$\|E_j^{(r)}\| \le \|E_0\|, \quad j \ge 0, \tag{4.74}$$

and

$$\|E_k^{(r)}\| \le \|E_l^{(r)}\|, \quad 0 \le l < k < m+1. \tag{4.75}$$

Similarly, we obtain for the choices $f_k = (H_k^{(r)})^H p_k$, $f_k = (H_k^{(r)})^H (\tilde{E}_k^{(r)})^H \tilde{E}_k^{(r)} p_k$, and $f_k = (H_k^{(r)})^H (p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$,

$$\|\tilde{E}_j^{(r)}\| \le \|\tilde{E}_0\|, \quad j \ge 0, \tag{4.76}$$

and

$$\|\tilde{E}_k^{(r)}\| \le \|\tilde{E}_l^{(r)}\|, \quad 0 \le l < k < m+1. \tag{4.77}$$

These results and the proofs of the previous sections lead to the following theorems for restarted Broyden and EN-like methods:

Theorem 4.9 *Assume* $\alpha_k = 1$ *or* $\alpha_k = q_k^H r_k / q_k^H q_k$ *for BBM(m)* ($f_k = q_k$)*, GCR-BM(m)* ($f_k = (E_k^{(r)})^H E_k^{(r)} q_k$)*, or PBM(m)* ($f_k = q_k - \sum_{i=k_0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$) *Assume additionally that* $\|E_0\| \leq \delta < 1$.

The following inequality then holds:

$$\|r_{k+1}\| \leq \delta \|r_k\|, \tag{4.78}$$

and the restarted methods converge.

Assume $\alpha_k = f_k r_k / f_k^H q_k$ *for GCR-BM(m). Assume additionally that* $\|E_0\| \leq \delta < \sqrt{2} - 1$.

Then,

$$\|r_{k+1}\| \leq \delta(\delta + 2) \|r_k\|, \tag{4.79}$$

and the restarted method converges.

Theorem 4.10 *Assume one of the following choices of* α_k*:*

(i) $\alpha_k = 1$,

(ii) $\alpha_k = p_k^H p_k / p_k^H H_k^{(r)} q_k$,

(iii) $\alpha_k = p_k^H H_k^{(r)} q_k / q_k^H (H_k^{(r)})^H H_k^{(r)} q_k$,

(iv) $\alpha_k = \|\tilde{E}_k^{(r)} p_k\|^2 / p_k^H (\tilde{E}_k^{(r)})^H \tilde{E}_k^{(r)} H_k^{(r)} q_k$, *or*

(v) $\alpha_k = p_k^H (\tilde{E}_k^{(r)})^H \tilde{E}_k^{(r)} H_k^{(r)} q_k / \|\tilde{E}_k^{(r)} H_k^{(r)} q_k\|^2$,

for GBM(m) ($f_k = (H_k^{(r)})^H p_k$)*, the dual GCR-BM(m)* ($f_k = (H_k^{(r)})^H (\tilde{E}_k^{(r)})^H \tilde{E}_k^{(r)} p_k$)*, or the dual PBM(m)* ($f_k = (H_k^{(r)})^H (p_k - \sum_{i=0}^{k-1} \tilde{p}_i^H p_k \tilde{p}_i)$) *Assume additionally that* $H_k^{(r)} A$ *is nonsingular, and* $\|\tilde{E}_0\| \leq \delta$*, where* $\delta < \frac{1}{2}$ *for* $\alpha_k = 1$*, and* $\delta < \frac{1}{3}$ *for the other choices for* α_k.

The following inequalities then hold:

$$\|e_{k+1}\| \leq \frac{\delta}{1 - \delta} \|e_k\| \tag{4.80}$$

for $\alpha_k = 1$*, and*

$$\|e_{k+1}\| \leq \frac{2\delta}{1 - \delta} \|e_k\| \tag{4.81}$$

for α_k *choices (ii) to (v), and the restarted methods converge.*

Theorem 4.11 *Assume for BEN(m) ($f_k = q_k$), EN(m) ($f_k = (E_k^{(r)})^H E_k^{(r)} q_k$), or PEN(m) ($f_k = q_k - \sum_{i=0}^{k-1} \tilde{q}_i^H q_k \tilde{q}_i$), that $\|E_0\| \le \delta < 1$.*

The following inequality then holds:

$$\|r_{k+1}\| \le \delta^2 \|r_k\|, \tag{4.82}$$

and the restarted methods converge.

Theorem 4.12 *Assume for GEN(m) ($f_k = (H_k^{(r)})^H p_k$), or the dual EN(m) ($f_k = (H_k^{(r)})^H (\tilde{E}_k^{(r)})^H \tilde{E}_k^{(r)} p_k$), that $H_k^{(r)} A$ is nonsingular, and $\|\tilde{E}_0\| \le \delta$, where $\delta < \sqrt{2} - 1$.*

The following inequality then holds:

$$\|e_{k+1}\| \le \frac{2\delta^2}{(1-\delta)^2} \|e_k\| \tag{4.83}$$

and the restarted methods converge.

Note that the restarted theory predicts that EN-like methods converge at approximately twice the rate of the corresponding Broyden method. Unfortunately, it is not possible to prove superlinear convergence for the restarted methods.

The convergence of the restarted SEN method for a real system, with the symmetric part of AH_0 positive definite, is also guaranteed. However, this result (stated in Corollary 4.1 in the next section) is easier to see from the convergence results developed for the truncated SEN method.

4.4 Convergence Behavior of Truncated Versions

For truncated methods, it is far more difficult to derive convergence results. Like the restarted versions, they do not possess finite termination.

Based on the definition of truncation, we have

$$\begin{aligned} H_i^{(t)} &= H_i, \quad i = 0, ..., m-1, \\ H_{k+1}^{(t)} &= H_k^{(t)} + (p_k - H_k^{(t)} q_k) v_k^H - (p_{k-m} - H_{k-m}^{(t)} q_{k-m}) v_{k-m}^H, \quad k \ge m-1 \end{aligned}$$

and, therefore, the corresponding error matrices are

$$\begin{aligned} E_i^{(t)} &= E_i, \quad i = 0, ..., m-1, \\ E_{k+1}^{(t)} &= E_k^{(t)} (I - q_k v_k^H) + E_{k-m}^{(t)} q_{k-m} v_{k-m}^H, \quad k \ge m-1. \end{aligned}$$

Clearly, the second term on the righthand side can cause problems. Unlike the full or restarted method, we are not able to find an upper bound on $\|E_k^{(t)}\|$, and it is possible that an increase in $\|E_k^{(t)}\|$ prevents convergence. This supports reports in [8] that truncated Broyden methods performed worse than restarted methods.

There is a convergence result in the literature for a truncated method, ORTHOMIN, which is truncated GCR [12] [14]. A more thorough investigation shows that GCR and ORTHOMIN are related to SEN and tSEN.

Lemma 4.23 *If the scaling parameter γ_k in (3.7) equals 0, the SEN iteration step is reduced to a GCR iteration step.*

Proof: $\gamma_k = 0$ leads to $p_k = 0$ and $t_k = r_k$. □

Since $\gamma_k = (AH_0 r_k)^H r_k / \|AH_0 r_k\|^2$, this situation can occur only, when AH_0 is not positive definite. Moreover, when $\gamma_k = 0$, $\delta_k = 0$, and the algorithm stagnates.

Comparing GCR(k) (or ORTHOMIN(k)) and SEN(k) (or tSEN(k)), one finds the following equivalence for the special case $k = 0$ for which restarted and truncated versions are equivalent:

Lemma 4.24 *Two steps of GCR(0) are equivalent to one step of SEN(0), i.e. given the same initial vector x_0,*

$$x_{2k}^{GCR(0)} = x_k^{SEN(0)}. \tag{4.84}$$

Proof: The proof is straightforward. □

Investigating tSEN shows that it is possible to apply the proof technique that is used for ORTHOMIN. Consider the application of tSEN(m) to a real linear system Ax=b.

Theorem 4.13 *Assume that the symmetric part M of AH_0 is symmetric positive definite, R is the skew-symmetric part of AH_0, η_i is the angle between c_i and $AH_0 r_k$, and $\{r_k\}$ the sequence of residuals generated by tSEN(m), then*

$$\|r_{k+1}\| \le (1 - \sum_{i=\phi(m)}^{k-1} \cos^2\eta_i)^{1/2} (1 - \frac{\lambda_{min}^2(M)}{\lambda_{max}(H_0^T A^T A H_0)})^{1/2} \|r_k\|, \tag{4.85}$$

and

$$\|r_{k+1}\| \le (1 - \sum_{i=\phi(m)}^{k-1} \cos^2\eta_i)^{1/2} (1 - \frac{\lambda_{min}^2(M)}{\lambda_{min}(M)\lambda_{max}(M) + |\lambda_{max}(R)|^2})^{1/2} \|r_k\|, \tag{4.86}$$

where $\phi(m) = k - m$, and the method converges.

Proof: Using the estimates

$$\frac{r_k^T AH_0 r_k}{\|r_k\|^2} = \frac{r_k^T M r_k}{\|r_k\|^2} \geq \lambda_{min}(M)$$

and

$$\frac{\|AH_0 r_k\|^2}{\|r_k\|^2} \leq \lambda_{max}(H_0^T A^T AH_0),$$

one can show

$$\begin{aligned}
\|r_{k+1}\|^2 &= \|t_k - c_k^T t_k c_k\|^2 \\
&= \|t_k\|^2 - (c_k^T t_k)^2 \\
&= \|(I - \gamma_k AH_0) r_k\|^2 - \sum_{i=\phi(m)}^{k-1} (c_i^T (I - \gamma_k AH_0) r_k)^2 \\
&= (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i) \|(I - \gamma_k AH_0) r_k\|^2 \\
&= (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i)(\|r_k\|^2 - 2\gamma_k r_k^T AH_0 r_k + \gamma_k^2 \|AH_0 r_k\|^2) \\
&= (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i)(\|r_k\|^2 - \frac{(r_k^T AH_0 r_k)^2}{\|AH_0 r_k\|^2}) \\
&= (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i) \|r_k\|^2 (1 - \frac{r_k^T AH_0 r_k}{\|r_k\|^2} \cdot \frac{r_k^T AH_0 r_k}{\|AH_0 r_k\|^2}) \\
&\leq (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i)(1 - \frac{\lambda_{min}^2 (AH_0)}{\lambda_{max}(H_0^T A^T AH_0)}) \|r_k\|^2.
\end{aligned}$$

The second inequality can be proved similarly using the following inequality, which is valid for all $x \neq 0$ and proved in [12],

$$\frac{x^T AH_0 x}{\|AH_0 x\|^2} \geq \frac{\lambda_{min}(M)}{\lambda_{min}(M)\lambda_{max}(M) + |\lambda_{max}(R)|^2}.$$

□

In the same way, we can prove convergence for the full SEN method and the restarted method SEN(m) and obtain similar estimates for the norm of residual with a slight change in the lower sum index. Clearly, the full method is a special case of the truncated method, where $m = k$. For the restarted method, we just need to set $\phi(m) = 0$. This leads to the following corollary.

Method	Choice of α_k	Convergence result	Theorem
PBM GCR-BM	$\alpha_k = 1$ $\alpha_k = \frac{q_k^H r_k}{\|q_k\|^2}$	If $\|I - AH_0\| \leq \delta < 1$ then, (1)$\|r_{k+1}\| \leq \delta \|r_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.5
GCR-BM	$\alpha_k = \frac{f_k^H r_k}{f_k^H q_k}$	If $\|I - AH_0\| \leq \delta < \sqrt{2} - 1$ then, (1)$\|r_{k+1}\| \leq \delta(\delta + 2)\|r_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.5
dual PBM, dual GCR-BM	$\alpha_k = 1$	If $\|I - A^{-1}H_0^{-1}\| \leq \delta < 1/2$ then, (1) $\|e_{k+1}\| \leq \frac{\delta}{1-\delta}\|e_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.6
GBM, dual PBM dual GCR-BM	$\alpha_k = \frac{\|p_k\|^2}{p_k^H H_k q_k}$ $\alpha_k = \frac{p_k^H H_k q_k}{\|H_k q_k\|^2}$ $\alpha_k = \frac{\|\tilde{E}_k p_k\|^2}{p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k}$ $\alpha_k = \frac{p_k^H \tilde{E}_k^H \tilde{E}_k H_k q_k}{\|\tilde{E}_k H_k q_k\|^2}$	If $\|I - A^{-1}H_0^{-1}\| \leq \delta < 1/3$ then, (1) $\|e_{k+1}\| \leq \frac{\delta}{1-\delta}\|e_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.6

Table 4.1 Summary of convergence results for full Broyden methods

Corollary 4.1 *Under the conditions of Theorem 4.13, SEN and SEN(m) converge. Additionally, Inequalities (4.85) and (4.86) are valid, with $\phi(m) = 0$ for SEN and SEN(m).*

If we recall the equivalent estimate for ORTHOMIN or GCR [12] [14],

$$\|r_{k+1}\| \leq \left(1 - \frac{\lambda_{min}^2(AH_0)}{\lambda_{max}(H_0^T A^T A H_0)}\right)\|r_k\|, \tag{4.87}$$

we see that the main difference between the two estimates is the factor

$$1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i,$$

which equals 1 only when $AH_0 r_k$ is orthogonal to all $c_i, i = \phi(m), ..., k-1$. In all other cases, the estimate is better. It is optimal when $AH_0 r_k$ is parallel to one of the c_i.

4.5 Summary

In the previous sections, we have derived some new convergence results for the Broyden methods. Specifically, we have investigated new choices for f_k (which yield GCR-BM, the dual GCR-BM, PBM, and the dual PBM) as well as some new choices for α_k. Although the convergence of PBM and the dual PBM has been investigated in the context of solving nonlinear

Class	Method	Convergence result	Theorem
General	All EN-like methods	If $f_k^H q_k \neq 0$ then, the method converges within at most n steps	Theorem 4.2
Full EN-like methods	EN, BEN, PEN	If $\|I - AH_0\| \leq \delta < 1$ then, (1)$\|r_{k+1}\| \leq \delta^2 \|r_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.7
	GEN, dual EN	If $\|I - A^{-1}H_0^{-1}\| \leq \delta < \sqrt{2} - 1$ then, (1) $\|e_{k+1}\| \leq \frac{2\delta^2}{(1-\delta)^2} \|e_k\|$ (2) the methods converge q-superlinearly.	Theorem 4.8
Restarted EN-like methods	EN(m), BEN(m) PEN(m)	If $\|I - AH_0\| \leq \delta < 1$ then, (1) $\|r_{k+1}\| \leq \delta^2 \|r_k\|$ (2) the methods converge q-linearly.	Theorem 4.11
	GEN(m) dual EN(m)	If $\|I - A^{-1}H_0^{-1}\| \leq \delta < \sqrt{2} - 1$ then, (1) $\|e_{k+1}\| \leq \frac{2\delta^2}{(1-\delta)^2} \|e_k\|$ (2) the methods converge q-linearly.	Theorem 4.12
Scaling invariant EN-like methods	SEN, SEN(m), tSEN(m)	If symmetric part of AH_0 is positive definite then, (1) the methods converge (2) $\|r_{k+1}\| \leq C(1 - \frac{\lambda_{min}^2(M)}{\lambda_{max}(H_0^T A^T A H_0)})^{1/2} \|r_k\|$ (3) $\|r_{k+1}\| \leq C(1 - \frac{\lambda_{min}^2(M)}{\lambda_{min}(M)\lambda_{max}(M) + \|\lambda_{max}(R)\|^2})^{1/2} \|r_k\|$ where $C = (1 - \sum_{i=\phi(m)}^{k-1} \cos^2 \eta_i)^{1/2}$	Theorem 4.13, Corollary 4.1

Table 4.2 Summary of convergence results for EN-like methods

systems [23], it has not been presented in this form before. The new convergence results for the full Broyden methods are summarized in Table 4.5. We have not summarized those for the restarted versions here, which are found in Theorems 4.9 and 4.10, since they are very similar. The major difference is that the restarted methods converge only q-linearly.

We have also derived several convergence results for the EN-like methods, which are summarized in Table 4.5. For many of the results, we have adapted analytical techniques used in the literature to investigate the convergence properties of the EN-like methods by exploiting their relationships with other well-known iterative methods. When alternative techniques were available, we opted for those that provided insight into the proper implementation of the EN-like methods. For example, the relationship to Broyden's method was used to prove the finite termination property of convergence within n steps (as opposed to Broyden's $2n$). We preferred O'Leary's approach, which does not only prove the finite termination property, like Gay's approach, but also characterizes it. The reduction in the upper bound on the number of steps to convergence was the first important improvement relative to Broyden's method, given that the EN-like methods require approximately twice as much work per iteration as Broyden's method.

Although finite termination is important, it does little to aid in the development of efficient methods. Fortunately, the relationship to Broyden's method again provided the mechanisms to show q-superlinear convergence and derive estimates for the residuals or errors of the full and restarted EN-like methods, which indicate we can expect convergence to be at least twice as fast as the related Broyden's methods.

Unfortunately, the connection to Broyden's method does not yield convergence results for the SEN method, which formally does not belong in the class of EN-like methods. Here, convergence results were obtained by using the connection between GCR and SEN, and the fact that the introduction of the scaling parameter leads to a minimization property that the EN-like methods are lacking.

In spite of the relationship between GCR or GMRES and the EN method, it was not possible to apply the proof technique of GMRES to the EN-like methods, since the convergence proof for GMRES makes use of the fact that the residual polynomial is the minimal polynomial, which is not the case for the EN-like methods. Like GCR and GMRES, the EN-like methods are polynomial methods, and their k-th residual can be expressed as $p(AH_0)r_0$, where p is a polynomial of degree $2k$. The vectors $z_0, ..., z_{k-1}$ (or $c_0, ..., c_{k-1}$ for EN) span a k-dimensional subspace of the $2k$-dimensional Krylov subspace $\mathcal{K}_{2k}(AH_0, AH_0r_0)$. (This was shown for the original EN method in [51].) Future work characterizing the relationship between the k-dimensional space produced by the EN-like methods to the (typically different) Krylov subspace produced by GMRES may be a key factor in determining when EN-like methods outperform GMRES (which they are seen to do in the results in Chapter 6).

We did not prove convergence results for the truncated EN-like versions, which is, in general, a very complex task. Our empirical results (see Chapter 6) show, however, that these versions are often effective. In fact, the success of the EN-like methods seems to be derived from the property that, in many cases, their convergence does not degrade as much as other methods when restarted or truncated versions are used. We plan to investigate more sophisticated truncation schemes that might be more robust in terms of convergence as well as lend themselves to more rigorous convergence analysis. We also plan to consider them in the context of application-specific circumstances that could give additional insight into their convergence behavior.

CHAPTER 5

PRECONDITIONING

The convergence theory has shown that, in general, it is desirable to start these methods with an H_0 that is a fairly good approximation of the inverse of the matrix of the linear system to be solved. This is essentially equivalent to determining a good preconditioner for the system. In this chapter, we consider two ways of preconditioning: preconditioning with an inner iterative method, and the use of an incomplete ILU-factorization based on numerical dropping. We present two techniques for the first approach, the straightforward use of an iterative solver as a preconditioner and orthogonality preserving inner/outer iteration schemes. The second preconditioning approach is presented in the context of PARASPAR, a robust software package. Finally, we describe the adaptive versions of the methods, which are intimately related to PARASPAR and therefore presented in this chapter.

5.1 Preconditioning with Iterative Methods

The idea of using an iterative method as a preconditioner can be found in the CGT-method introduced by Rutishauser, who used the Chebyshev method as inner method to precondition the conjugate gradient method, an approach equivalent to Chebyshev polynomial preconditioning [36].

Methods have also been proposed for the nonsymmetric case. These include FGMRES [37] with GMRES as the outer method, GMRESR [48] with GCR as the outer method and GMRES as the inner method, and others considered further in this and the following section.

If the inner method is a polynomial method, i.e., the residuals can be expressed in terms of a polynomial in A applied to the original residual r_0 (which is the case for all the methods considered here), this approach can be considered as a type of polynomial preconditioning.

This type of preconditioning is defined by evaluating any occurences of the form $z = H_0 y$ by applying m iterations of an inner iterative method to the linear system $Az = y$. If the inner method is a polynomial method, this can also be expressed as

$$z = P_{m,k} y. \tag{5.1}$$

Consequently, since the coefficients of the polynomials in general depend on the original residual used in the iterative process, which in turn depends on the vector y, we encounter a different preconditioner in each iteration step of the outer method.

Since this approach is possible for any of the methods considered in the previous sections, a large variety of methods can be developed by choosing a pair of methods and specifying which is the inner and which is the outer method. We denote these methods by outer method/inner method. Note that GCR/GMRES(m) is equivalent to GMRESR(m). For our experiments the notation inner method(m) indicates $m+1$ iterations of the full inner method, except for GMRES(m), which indicates m iterations of full GMRES. For the outer methods we consider full, restarted, and truncated methods, which are denoted as in Chapter 3.

Let us now consider the complexity of these methods. Clearly, one iteration step of any of these combinations is far more expensive than the plain methods. The complexity expressions in Chapter 3 were all of the form

$$(n_1 k + n_2)n + l(smv + prec),$$

where l was either 1 or 2. The k-th iteration step of outer method/inner method(m) therefore requires

$$(n_1^{(o)} k + n_2^{(o)} + l^{(o)}(m+1)(\frac{m}{2} n_1^{(i)} + n_2^{(i)}))n + l^{(o)}(1 + l^{(i)}(m+1))(smv + prec)$$

floating point operations and $m^{(i)}(k) + m^{(o)}(k) - 2$ vectors of memory not counting those necessary for the matrix, preconditioner, righthand side, and solution, where $m^{(i)}(k)$ and $m^{(o)}(k)$ are the memory requirements for the inner and outer methods, respectively.

The performance of these methods is considered in detail in Chapter 6.

5.2 Preserving Orthogonality

A potential drawback of the inner/outer schemes in the previous section is that the Krylov subspace generated by the outer iteration is ignored when the inner method is applied to the equation $Az = y$. This was observed by de Sturler and Fokkema for the case of GMRESR [43]. Here, the outer method generates a minimal residual polynomial that is completely ignored by the inner method, which searches for a new minimal residual polynomial. They therefore suggest (the inner method to solve) the following system

$$(I - C_{k-1}C_{k-1}^H)Ax = r_k, \tag{5.2}$$

where C_{k-1} is the matrix consisting of the orthogonal vectors $c_0, ..., c_{k-1}$ from the outer method. This approach keeps the inner residual orthogonal to the outer Krylov subspace. This improves in many cases the convergence of GMRESR in spite of the higher number of operations per iteration step. They refer to the method as GCRO.

Since EN and SEN also generate orthogonal vectors $c_0, ..., c_{k-1}$, it is possible to use the same approach with EN or SEN as outer method. For every occurrence of $w = H_0 v$, one can apply an inner iterative method to $(I - C_{k-1}C_{k-1}^H)Aw = v$. We denote these new orthogonality preserving methods outer method/inner method-O.

Fokkema and de Sturler also noticed that it is possible to simplify the outer iteration for GCRO significantly, since much of the information it needs is generated in the inner iteration. The simplified outer iteration for GCRO is given through

Algorithm 15. GCR/inner method-O

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$

For $k = 0, 1, \ldots, m$:

$$\begin{aligned} solve \quad & (I - C_{k-1}C_{k-1}^H)A\tilde{u}_k = r_k \\ \tilde{c}_k &= r_k - r^{inner} \\ \rho_k &= \|\tilde{c}_k\|^2 \\ \delta_k &= \frac{\tilde{c}_k^H r_k}{\rho_k} \\ r_{k+1} &= r_k - \delta_k \tilde{c}_k \end{aligned}$$

end

$$x_k = x_0 + \tilde{U}_m A_m^{-1} d_m,$$

where $\alpha_{i,k}$ and r^{inner}, the residual of the inner method, are evaluated by the inner method.

Additionally, one can show that if the inner method is GMRES or GCR,

$$\begin{aligned} \delta_k &= 1, \\ r_{k+1} &= r^{inner}. \end{aligned}$$

Using the same reasoning for EN and SEN as outer method, we get the following algorithm, where setting $\gamma_k = 1$ yields EN/inner method-O,

Algorithm 16. SEN/inner method-O

Initialization:

x_0, H_0 arbitrary, $r_0 = b - Ax_0$, $\tilde{x}_0 = x_0$,

For $k = 0, 1, \ldots, m$:

$$\begin{aligned} solve \quad & (I - C_{k-1}C_{k-1}^H)A\hat{y}_k = r_k \\ t_k &= (1-\gamma_k)r_k + \gamma_k r_1^{inner} \\ solve \quad & (I - C_{k-1}C_{k-1}^H)A\hat{u}_k = t_k \\ \tilde{c}_k &= t_k - r_2^{inner} \\ \rho_k &= \|\tilde{c}_k\|^2 \\ \delta_k &= \frac{\tilde{c}_k^H t_k}{\rho_k} \\ \tilde{x}_{k+1} &= \tilde{x}_k + \gamma_k \hat{y}_k \\ r_{k+1} &= t_k - \delta_k \tilde{c}_k \end{aligned}$$

end

$$x_{m+1} = \tilde{x}_{m+1} + \hat{U}_m B_m^{-1}(d_m - A_m e),$$

where $\alpha_{i,k}$, r_1^{inner}, γ_k, $\beta_{i,k}$ and r_2^{inner} are evaluated in the preceding inner iterations.

The new forms have the same memory requirements as the original inner-outer schemes, and the number of floating point operations in the k-th iteration step of outer method/inner method(m)-O is

$$(s^{(o)} + l^{(o)}(m+1)(\frac{m}{2}n_1^{(i)} + n_2^{(i)} + 2t^{(o)}kl^{(i)}))n + l^{(o)}l^{(i)}(m+1)(smv + prec),$$

where $s^{(o)}$ equals 7 for GCR, 9 for EN, and 12 for SEN as outer methods, and $t^{(o)}$ equals 1 if the outer method is full or restarted, and 2 if the outer method is truncated. Note that the number of sparse matrix vector operations is decreased, and the number of dense matrix vector operations is significantly increased. This can be of advantage on an architecture with very fast dense matrix vector operations or in the case of an expensive sparse matrix vector multiplication and/or preconditioning step.

5.3 PARASPAR

A hybrid software package called PARASPAR, which is based on Y12M [22] [52] and combines both iterative and direct methods, has been used as a framework for some of our experiments. Although direct methods (sparse Gaussian elimination schemes) generally achieve high accuracy, they are often too time consuming and can have difficulty exploiting parallelism. Iterative methods have more parallelism and, when convergent, require a significantly lower computing time. Unfortunately, they lack the robustness of direct methods. PARASPAR takes advantage of the desirable qualities of both groups while attempting to minimize their disadvantages.

One of PARASPAR's strengths is its robustness, which makes it a very promising code for those linear systems that are problematic for many existing iterative solvers and preconditioners. The elements of the package that lead to its robustness are an ILU-preconditioner that uses numerical dropping and various pivoting strategies, a sophisticated stopping criterion used for the iterative methods, and an effective heuristic strategy to determine a good preconditioner (see Figure 5.1). During the evaluation of the preconditioner, elements are dropped when they are below the given drop tolerance. If the preconditioner based on the current drop tolerance does not exit (the approximate factorization fails) or is not effective (the iterative method fails), the package automatically decreases the drop tolerance, reevaluates the preconditioner, and tries again. This is repeated until the solution has been obtained with the desired accuracy. In the worst case, the drop tolerance is zero, and a complete LU-factorization with a few postprocessing steps of the iterative method is performed. The complete algorithm is given in Figure 5.1.

```
Initial drop tolerance τ and desired accuracy ε given
Do until (x_k is accepted):
        if (LU(τ) exists) then
                M ← LU(τ)
                if (not converged or too slow) then
                        τ ← f_1(τ)
                endif
        else
                τ ← f_2(τ)
        endif
end
```

Figure 5.1 PARASPAR

PARASPAR has the option of searching for and applying parallel pivots and can be very effective on a parallel computer. For further efficiency, it also switches to dense matrix techniques when the portion of the sparse matrix yet to be factored becomes small enough or dense enough due to fillin for dense matrix techniques to be more efficient than sparse matrix techniques on the target architecture.

A key question for PARASPAR is the choice of drop tolerance. There are two major problems: first, how to choose the initial drop tolerance and second, how to alter the drop tolerance, if we have to reevaluate the preconditioner. Certainly, this depends on the matrix considered and possibly on the iterative method chosen.

For the experiments in Chapter 6, the following strategy is used. In order to reduce the influence of extreme matrix values, column-row equilibration is performed before the creation of the preconditioner. Elements are dropped, if they satisfy

$$|a_{i,j}^{(k)}| \leq max(|a_{i,k+1}^{(k)}|, \ldots, |a_{i,n}^{(k)}|)\tau, \tag{5.3}$$

where $a_{i,j}^{(k)}$ denotes an element at stage k and τ the drop tolerance. The initial drop tolerance is set to 0.0625, which is, in case of failure, gradually decreased by a factor of 1/1024 if the ILU-factorization fails or 1/32 if the iterative method fails.

5.4 Adaptive Versions

All the methods under investigation have been added to PARASPAR. In the process of doing so, we have considered the following issues:

- the choice of stopping criterion,
- an adaptive procedure that automatically alters the size of the underlying subspace of the methods,
- criteria to detect and deal with failures of the methods, e.g., breakdown, divergence, or stagnation,
- criteria to evaluate the quality of the preconditioner.

All the above points are closely related; therefore, we treat them together in this section. Our strategy varies somewhat for the different methods we are considering; however, the overall concept is very similar.

5.4.1 Stopping Criterion

The EN-like and Broyden methods naturally use a right preconditioner. All iterative methods in the original implementation of PARASPAR use a left preconditioner. Consequently, the vector r_k that is evaluated in the EN-like and Broyden method is the actual residual and not the preconditioned residual. Therefore, it appears reasonable to replace the sophisticated stopping criterion in PARASPAR with the simpler one

$$\frac{\|r_k\|}{\|r_0\|} \leq \epsilon. \tag{5.4}$$

In order to avoid the loss of accuracy during the evaluation of the r_k of each algorithm leading to false convergence, we explicitly evaluate $b - Ax_k$, when the above criterion indicates convergence, and restart the algorithm with $x_0 = x_k$, when $b - Ax_k$ does not fulfill the stopping criterion.

5.4.2 Adaptive Strategy

We implemented the truncated versions of the methods with the option of restarting whenever potentially curable failures of the truncated methods occur. Consequently, the resulting version is a combination of a truncated and a restarted version with emphasis on truncation. We denote the methods with *adaptive tMethod(l)* or *ad. tMethod(l)*, where l is the largest subspace dimension allowed. We prefer truncated methods in this context, since it is possible to increase the subspace in the next iteration step. For restarted methods, the increase of the subspace dimension from k to $k+1$ does not take effect until up to k iteration steps later.

In order to determine when to increase the subspace, we monitor the convergence rate

$$\rho_k := \frac{\|r_{k+1}\|}{\|r_k\|} \tag{5.5}$$

assuming, since $Ae_k = r_k$, that this is also a good estimate for the convergence rate

$$\eta_k := \frac{\|e_{k+1}\|}{\|e_k\|} \tag{5.6}$$

for GBM and GEN. The subspace dimension is increased by one if either

$$\rho_{k+1} > \rho_k \tag{5.7}$$

or

$$\rho_k \geq c, \tag{5.8}$$

where c was determined empirically. The value $c = 0.8$ was observed to be a good choice for the EN-like methods and $c = 0.9$ for the Broyden-methods and GCR. Note that the value of c for the Broyden methods is approximately the square of the value for the EN-like methods, which is consistent with the convergence theory in Chapter 4.

5.4.3 Treatment of Failures of the Iterative Methods

Let us now consider the different possibilities of failure for the various methods.

Breakdowns occur when we encounter a divide by zero for any of the methods. For this case, in general, we restart the iterative solver.

A breakdown occurs for EN, SEN, or GCR (or ORTHOMIN), when $\|\tilde{c}_k\| \approx 0$. Since $\tilde{c}_k = (I - C_{k-1}C_{k-1}^H)AH_0w$, this means that $AH_0w \in R(C_{k-1})$; consequently a restart may help, since it generates new c_i. It is also possible that for EN or SEN a breakdown occurs, because $r_k - \gamma_k AH_0 r_k = 0$; i.e., r_k is an eigenvector of AH_0. In this case, a restart leads to another breakdown. We did not encounter this type of breakdown in any of our experiments.

A breakdown occurs for GEN and GBM, when $p_k^H H_k q_k \approx 0$, and for BEN and BBM, when $\|q_k\| \approx 0$. Since $r_k \approx 0$ should have led to the termination of the algorithm, and $q_k = AH_k r_k$, this indicates a singular H_k. Unless k is small, in which case we should reevaluate the preconditioner, a restart can possibly avoid another breakdown. For GEN and GBM, this breakdown could also be caused by the orthogonality of p_k and $H_k q_k$.

There is another possibility of breakdown for SEN: $\|AH_0 r_k\| \approx 0$. Assuming that r_k is not close to zero, this strongly indicates a singular preconditioner, in which case one should terminate the iterative method and reevaluate the preconditioner.

Other possibilities of failure for the iterative methods are divergence or stagnation. These can be detected through monitoring the convergence rate. For GCR (or ORTHOMIN), BBM and SEN, theoretically, the convergence rate ρ_k can never be greater than one. A convergence rate greater than one in practice for any of those methods indicates severe ill-conditioning, and one needs to terminate the algorithm and reevaluate the preconditioner. For the other methods, there is no limit on ρ_k. A very large ρ_k is an indicator for divergence, and in this case the preconditioner is reevaluated. In fact, the propensity of a particular method to diverge dramatically is a significant benefit for PARASPAR's hybrid strategy. (It is demonstrated in Chapter 6 that some EN-like methods have this fortunate tendency.) In order to also detect stagnation or a slow increase of the convergence rate, we monitor the mean value of the convergence rates of the last four steps. If the mean value stays larger than 0.99995 for a limited number l of iteration steps (we set $l = 5$), the algorithm is restarted.

There is another possibility of stagnation for SEN and ORTHOMIN (or GCR). If the scaling parameter γ_k in SEN or the parameter δ_k in ORTHOMIN (or GCR) becomes 0 or very small, SEN or ORTHOMIN (or GCR) stagnates, and restarting does not improve the situation. In this case the algorithm is terminated and the preconditioner reevaluated.

Finally, we allow only a limited number of restarts (5) and limit the total number of iterations per preconditioner to 300, in order to avoid spending too much time due to extreme low convergence. When either limit is exceeded, the preconditioner is reevaluated.

5.4.4 Estimates for the Quality of the Preconditioner

There are several parameters that are indicators for the quality of the preconditioner. We have already mentioned that in SEN $\|AH_0 r_k\| \approx 0$ indicates a singular preconditioner. Furthermore, several of the algorithms evaluate as a byproduct, i.e., without extra cost, a lower bound for $\|AH_0\|$ or $\|AH_k\|$. Large values of these are not desirable, since the optimal $H_k = A^{-1}$ yields $\|AH_k\| = 1$. We obtain lower bounds for ORTHOMIN through

$$\frac{\|\tilde{c}_k\|}{\|r_k\|} \leq \|AH_0\|;$$

for SEN through

$$\frac{\|\tilde{c}_k\|}{\|r_k\|} \leq \|AH_0\|^2$$

and

$$\frac{\|AH_0 r_k\|}{\|r_k\|} \leq \|AH_0\|;$$

for EN through

$$\frac{\|\tilde{c}_k\|}{\|r_k\|} \leq \|AH_0\|(1 + \|AH_0\|),$$

which, since we are interested in very large parameters only, is approximately

$$\frac{\|\tilde{c}_k\|}{\|r_k\|} \leq \|AH_0\|^2;$$

for GBM and GEN through

$$\frac{p_k^H H_k q_k}{\|p_k\|^2} \leq \|H_k A\|;$$

and for BBM and BEN through

$$\frac{q_k^H r_k}{\|r_k\|^2} \leq \|AH_k\|.$$

For all methods, we terminate the algorithm and reevaluate the preconditioner when the above estimates indicate that $\|AH_0\|$ (or $\|H_0 A\|$ for GBM and GEN) are larger than a value η. For GBM, GEN, BBM, and BEN, the inequalities are indicators for quality of the preconditioner only, when $k = 0$. In this case, the preconditioner is reevaluated. If the values are large, when $k \neq 0$, the methods are restarted. Certainly, the question arises, how to choose η. If η is too small, a potentially successful iterative process might be terminated, whereas a too large η can lead to a large amount of unnecessary iterations. For our experiments in Chapter 6, we chose η to be 1000.

CHAPTER 6

NUMERICAL EXPERIMENTS

In this chapter, we illustrate some of the properties of the algorithms described in the previous chapters with numerical experiments. For our experiments, we use several matrices derived from partial differential equations as well as a few test matrices from the Harwell-Boeing collection. All runs are performed on an Alliant FX/80, a parallel computer with 8 vector processors, using vectorization and parallelization. For the stopping criterion, we choose $\epsilon = 10^{-8}$. For GBM and BBM, we choose $\alpha_k = f_k^H r_k / f_k^H q_k$. Note that the implementation of GMRES here uses classical Gram-Schmidt and is therefore more efficient than the usual implementation with modified Gram-Schmidt. For the test problems considered, the use of classical Gram-Schmidt does not effect the stability. The first three matrices are obtained through a standard five point finite difference discretization of the following two-dimensional partial differential equation, which is taken from [48],

$$-u_{xx} - u_{yy} + \beta(u_x + u_y) = f \quad on\ \Omega \tag{6.1}$$

with Dirichlet boundary conditions

$$u = c \quad on\ \partial\Omega, \tag{6.2}$$

where $\Omega = [0,1] \times [0,1]$.

We choose step size $h = 0.01$, which leads to a matrix of order 9801.

6.1 Problem 1

For our first problem we choose $\beta = 1$. This is an example for which CGS and BiCGSTAB converge fairly quickly. Since their computational complexity per iteration and memory requirement are low, they are hard to beat in such a case. On the other hand, GMRES(k+1) converges very slow for this case, even for an optimal (with regard to time) k. Since CGS and

BiCGSTAB are very efficient with respect to storage and GMRES succeeds with $k = 34$, we restrict the memory used by the Broyden and EN-like methods to k smaller than 17, which corresponds for GMRES to k smaller than 34. The results are summarized in Table 6.1.

GEN, GBM, and BBM are not competitive. Restarted eEN, eSEN, and BEN are about 50 % faster than GMRES; i.e., their performance is closer to GMRES than to CGS and BiCGSTAB. This is due to a significant reduction in additional flops, i.e., dmv's, daxpys, and inner products, which also offsets the increase in sparse matrix vector multiplications for restarted eEN and eSEN.

ORTHOMIN, tGEN, tGBM as well as tBBM fail. However, tEN, tSEN and tBEN work very well, and are only about 50 % slower than CGS and BiCGSTAB. tBEN(3) is 39 % slower and efficient in memory. The decrease in additional flops for tBEN is caused by the smaller computational complexity per iteration step of tBEN compared to that of tEN and tSEN. So we see that for this problem one member of the EN-like method performs competitively with the very efficient CGS and BiCGSTAB.

Since memory requirement can be crucial for GMRES, the Broyden and EN-like methods, times are plotted vs. k in Figure 6.1. These results are interesting, since now we see that if we have only a limited amount of memory available, the EN-like methods become even more attractive in comparison to GMRES. Consider for example the case $k = 4$, i.e., about 12 memory vectors; now eSEN(4) is more than three times faster than GMRES(9). Since these times are of course machine dependent and the results can vary significantly on different architectures, we also show the number of sparse matrix vector multiplications vs. k in Figure 6.2, and the number of additional flops/n vs. k in Figure 6.3 for GMRES, BEN, eSEN, and tSEN. We used tSEN as an example for the truncated method since it is here overall (i.e., for almost all k) the best performing of the truncated methods. In Figure 6.2, we see that, for $k = 4$, GMRES(9) needs about three times as many sparse matrix vector multiplications as eSEN(4) and six times as many as tSEN(4). With increasing k, the difference decreases. Whereas overall with increasing k the number of sparse matrix vector multiplications decreases, this is not the case for the additional operations. For GMRES, the number of flops is high overall; for the other methods, it increases steadily with the strongest increase for the truncated method tSEN.

Method	Time	no. of its.	no. of smv	add. flops/n
CGS	13.5	246	492	3690
BiCGSTAB	13.7	236	472	4720
GMRES(33)	45.7	838	838	60336
eEN(8)	31.9	460	920	19320
eSEN(4)	29.0	466	932	14912
BEN(12)	28.7	346	692	19722
GEN(16)	60.7	603	1206	44019
GBM(8)	61.0	1375	1375	37125
BBM(12)	78.4	1673	1673	58555
tEN(3)	21.2	271	542	12466
tSEN(3)	20.3	256	512	13312
tBEN(3)	18.8	263	526	8679
tGEN, tGBM	diverges			
ORTHOMIN, tBBM	stagnate			

Table 6.1 Times for Problem 1

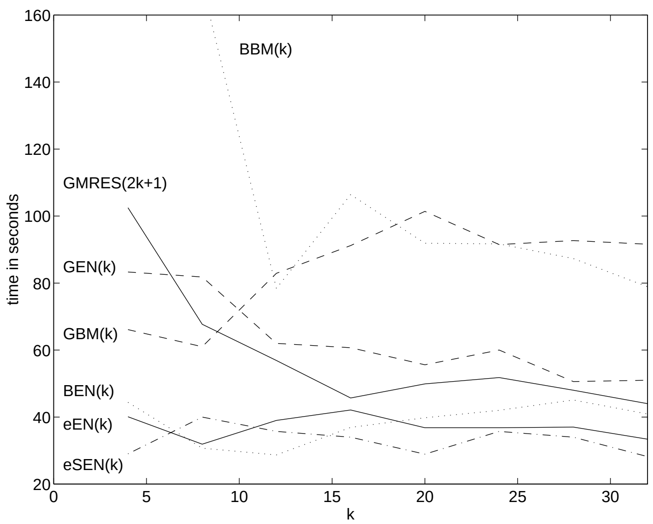

Figure 6.1 Times for restarted methods with varying k

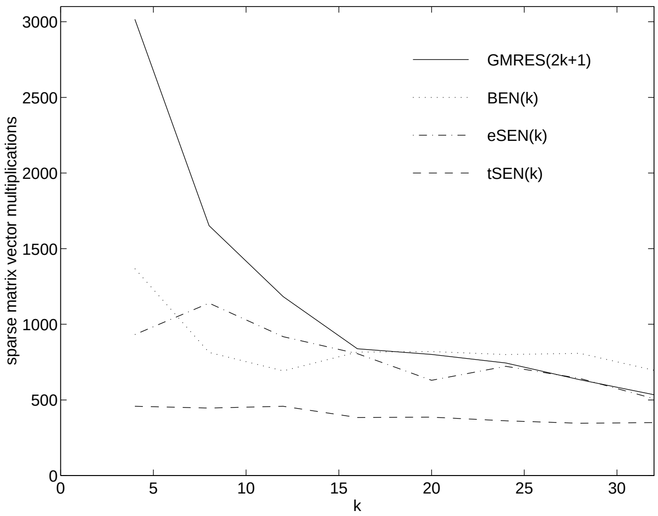

Figure 6.2 Number of matrix vector operations for several methods with varying k

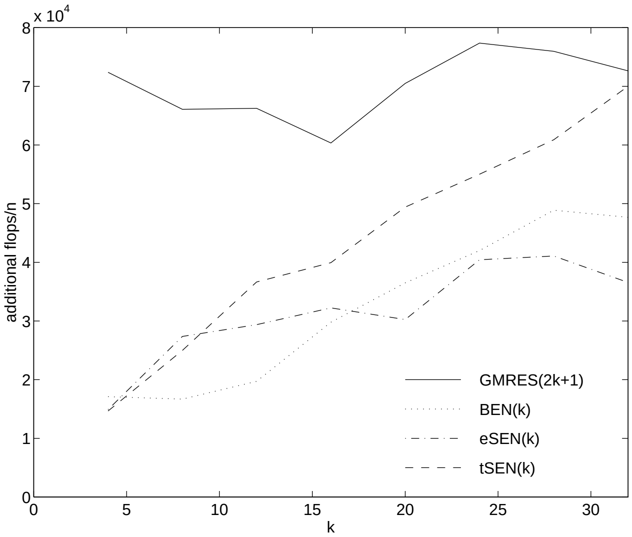

Figure 6.3 Number of additional flops/n for several methods with varying k

Method	Time	no. of its.	no. of smv	add. flops/n
CGS	diverges			
BiCGSTAB	false convergence			
GMRES(3)	9.6	338	338	4056
eEN(2)	9.6	171	342	3078
eSEN(1)	10.0	177	354	3540
sBEN(2)	10.5	170	340	3234
sGEN(8)	14.9	191	382	8217
GBM(10)	14.1	295	295	9145
sGBM(2)	12.1	331	331	5631
BBM(1)	28.3	825	825	10725
tEN(2)	12.7	181	362	6154
tSEN(1)	13.7	205	410	5740
tsBEN(2)	16.8	236	472	6476
tsGEN(2)	21.8	326	652	8806
ORTHOMIN, tBBM	stagnate			
tGBM	diverges			

Table 6.2 Times for Problem 2

6.2 Problem 2

Even though CGS and BiCGSTAB are the fastest for the previous problem and require little storage, their robustness can be a problem. This is evident in Problem 2, where we choose $\beta = 500$.

For this example, BEN and GEN fail. They both run into the situation we have also observed in Figure 2.6, which leads to extremely large residuals and failure. However, use of the scaling-invariant versions described in Section 2.4 with $z = (1, ..., 1)^T$ turns out to be successful, see Table 6.2. Truncation leads to slower convergence and is consequently not competitive with the restarted methods.

For this example, GMRES converges very quickly. As a matter of fact, it converges best for small k. Under these circumstances, GMRES is difficult to beat. Nevertheless, the times for eSEN with comparable memory requirement, and sBEN and eEN with a slightly larger memory requirement are competitive. Among the truncated methods, tEN is only about 30 % slower. The increase in time is caused by the larger complexity per iteration step of the truncated method compared to the restarted method.

Method	Time	no. of its.	no. of smv	add. flops/n
CGS	diverges			
BiCGSTAB	false convergence			
GMRES(17)	47.9	1166	1166	46640
eEN(8)	31.5	459	918	19278
eSEN(8)	41.0	578	1156	27744
sBEN(4)	48.8	730	1460	19714
GBM(8)	76.9	1722	1722	46494
tEN(8)	60.4	554	1108	58724
ORTHOMIN, tBBM	stagnate			
tGBM	diverges			

Table 6.3 Times for Problem 3

6.3 Problem 3

The first two problems illustrate cases where BiCGSTAB and CGS do very well and GMRES very poorly and vice versa. In the first case, members of the family of EN-like methods are far better than GMRES, and the fastest only 39 % slower than BiCGSTAB and CGS, in the second case, some EN-like methods are competitive with the better method, GMRES. For Problem 3, we choose

$$\beta(x,y) \;=\; \begin{cases} 1 & if (x,y) \in [0.4, 0.6] \times [0.4, 0.6] \\ 500 & elsewhere. \end{cases} \tag{6.3}$$

This is an example where eEN wins. Both CGS and BiCGSTAB fail as in the previous example (see Table 6.3), and GMRES converges slowly. The truncated methods are not competitive here. However, sBEN with a smaller memory requirement is competitive with GMRES. Both eSEN and eEN have superior convergence with comparable memory requirement. GMRES(17) is about 50 % slower than eEN(8) due to both a higher count in sparse matrix vector operations as well as a significantly higher additional flop count.

6.4 Testing the Adaptive Schemes

The following example is taken from the Harwell-Boeing collection. SHERMAN3 is a matrix of order 5005 with 20033 nonzeroes. We precondition it with the incomplete LU factorization of PARASPAR using a drop tolerance of 0.0625 (pILU(0.0625)). For this example, we test the adaptive versions of the methods described in Chapter 5. Figure 6.4 shows the number of sparse matrix vector multiplications vs. the norms of the residuals. We see here that the best method is the adaptive ORTHOMIN, closely followed by the adaptive tEN. BiCGSTAB and GMRES(53) need about twice as many sparse matrix vector multiplications. If we consider the solution times achieved on the Alliant FX/80, which are given in Table 6.4 (note that the time it took to generate the preconditioner (5.6 seconds) is not included), we see that the adaptive tEN is actually the fastest. It is followed by the adaptive tBEN, which here, in spite of a very similar memory requirement and iteration count as tSEN, is more than 10 % faster than tSEN. We see the influence of the larger computational complexity of tSEN. ORTHOMIN, in spite of its small smv count, is about 20 % slower than the adaptive tEN. Note, however, that ORTHOMIN requires almost twice as many additional work vectors, and consequently its operation count for dense matrix vector operations is significantly higher than for the EN-like methods.

For this example, ORTHOMIN and tSEN have a tendency to stagnate. The smallest k for which ORTHOMIN(k) does not stagnate is $k = 48$; for tSEN it is $k = 30$. Note that in both cases the adaptive version finds a k close to these values.

To compare these methods further, we restrict the memory requirement for the adaptive ORTHOMIN to that of the optimal adaptive tEN, which is a maximum of $k = 26$. The results for these experiments can be seen in Figure 6.5. As mentioned above, ORTHOMIN(k) fails for $k \leq 26$. The adaptive scheme converges however, since it is a hybrid of a truncated and a restarted method. Figure 6.5 shows phases of stagnation in the ad. ORTHOMIN(26) (curve 'C') which are overcome through restarts. The ad. ORTHOMIN(26) is four times as slow as the optimal adaptive ORTHOMIN, which requires far more memory.

The peak in the curve for the adaptive ORTHOMIN(26) is caused by false convergence. The residual seems to indicate convergence where the actual error stagnates, see Figures 6.5 and 6.4. In order to overcome this problem, we reevaluate the residual by $r_k = b - Ax_k$ and restart at this point with the new residual, which finally leads to convergence.

We include eGCR(26) in this experiment to compare it with the ad. ORTHOMIN(26). The ad. ORTHOMIN(26) converges faster than eGCR(26) (see Figure 6.5), but is actually somewhat slower with regard to time here, since its number of additional flops is higher. This is not true if we choose $k = 25$ or $k = 27$ for which the ad. ORTHOMIN(k) is faster than eGCR(k) (ad. ORTHOMIN(25): 15.2 secs., eGCR(25): 19.3 secs., ad. ORTHOMIN(27): 15.3 secs.,

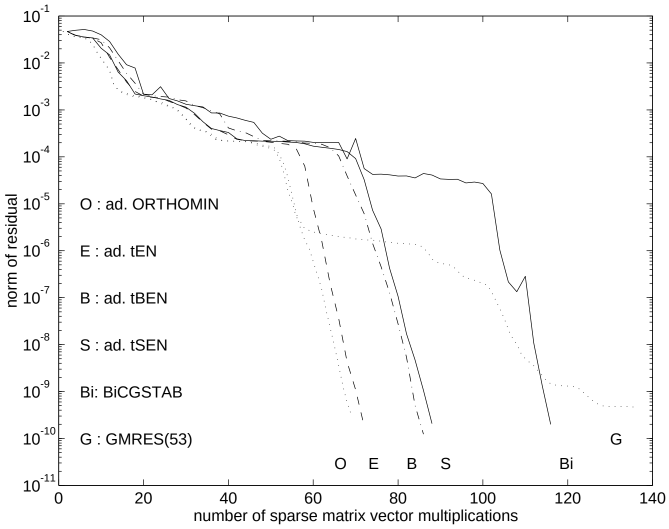

Figure 6.4 Residuals for SHERMAN3 with pILU(0.0625)

eGCR(27): 27.4 secs.), or choose a lower accuracy for the stopping criterion (e.g., $\epsilon = 10^{-6}$, for which the ad. ORTHOMIN(26) takes 12.7 and eGCR(26) 14.2 seconds).

In summary, the results above show that the adaptive methods are very efficient and superior to the restarted and truncated methods. When memory is not restricted and the truncated method has a tendency to stagnate, they determine a subspace size close to the smallest necessary for the truncated method to succeed. If memory is limited, the adaptive method is likely to succeed where the truncated method fails, since it is a hybrid of a truncated and restarted method. The overall fastest method for this example is the adaptive tEN.

6.5 Testing the Methods inside PARASPAR

The following example is LNS_3937 from the Harwell-Boeing collection, a matrix of order 3937 with 25407 nonzeroes. Many standard preconditioners fail for this matrix. We use this example to demonstrate the influence of the characteristics of the various methods on the solution time in the context of PARASPAR.

We use as initial drop tolerance 0.05. The resulting preconditioner is not good enough to solve the problem. Table 6.5 shows the reaction of the various methods to this situation. Here, the time to evaluate the preconditioners is included. Both CGS and BiCGSTAB iterate until the

Method	time (sec.)	no. of its.	no. of smv	max. k
ad. ORTHOMIN	5.9	68	68	46
ad. tEN	4.9	35	70	26
ad. tBEN	5.5	42	84	30
ad. tSEN	6.2	43	86	30
BiCGSTAB	5.7	57	114	-
GMRES(53)	9.4	135	135	52
eEN(26)	9.5	81	162	26
tEN(26)	5.8	39	78	26
ORTHOMIN(48)	9.5	94	94	48
eGCR(26)	18.8	319	319	26
ORTHOMIN(26)	stagnates			26
ad. ORTHOMIN(26)	20.5	269	269	26

Table 6.4 Solution times for various methods for SHERMAN3 with pILU(0.0625)

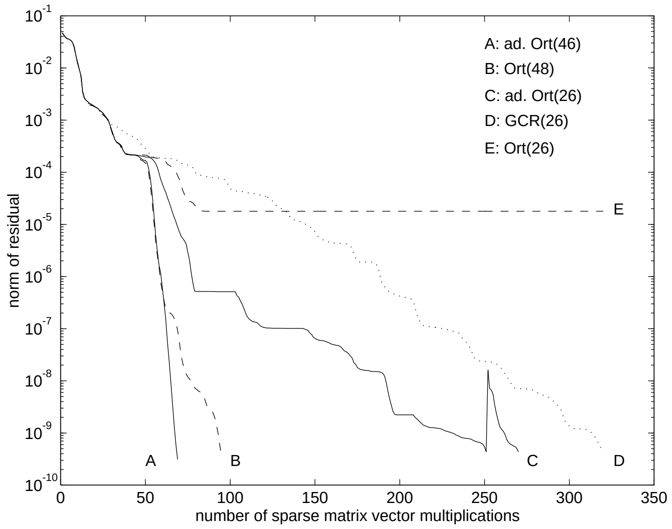

Figure 6.5 Adaptive vs. truncated ORTHOMIN for SHERMAN3 with pILU(0.0625)

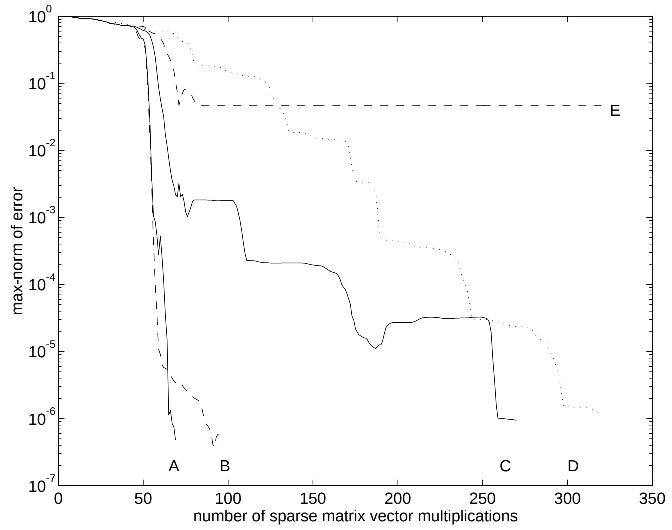

Figure 6.6 Adaptive vs. truncated ORTHOMIN for SHERMAN3 with pILU(0.0625)

maximum number of iterations is reached. CGS displays a very erratic convergence behavior, which is typical for CGS with extreme peaks and is still far from the solution after 300 iterations. BiCGSTAB converges more smoothly with less serious peaks and reaches $\|r_k\| \approx 10^{-5}$ after 300 iterations.

The adaptive tEN, tBEN, tGEN, and tGBM encounter a significant increase in $\|r_k\|$ for increasing k from the very beginning, which clearly indicates divergence, and consequently a new preconditioner can be detected quickly. Both the adaptive ORTHOMIN, tSEN and tBBM take longer to determine their failure, since for these methods $\|r_k\|$ cannot increase theoretically, a bad preconditioner very likely will lead to stagnation or extreme slow convergence, which takes longer to detect, since we do not want to terminate the iteration process, if there is still hope for convergence. We see that tGEN and tBBM do not even converge with the second preconditioner pILU(1.56e-3) and a third preconditioner evaluation is necessary, which leads to pILU(4.88e-5).

This example shows that in the context of PARASPAR the disastrous behavior of many of the EN-like methods when facing an ill-conditioned problem is of advantage, since one wastes no time in useless iterations.

In practice, PARASPAR is often used to determine a preconditioner by this adaptive procedure, which is then used for subsequent systems with related matrices. In this case, the rapid adaptation to determine the preconditiner is not as crucial, while the time to solve a system

Method	tot.time	no. of trials	no. of its. trial 1	trial 2	trial 3	last run sol. time	k
CGS	42.4	2	300	22		3.1	-
BiCGSTAB	43.5	2	300	19		2.6	-
ad. tEN	17.5	2	7	15		2.3	9
ad. tBEN	18.6	2	10	21		3.2	12
ad. tSEN	35.5	2	122	19		3.0	13
ad. tGEN	37.7	3	7	37	3	0.6	2
ad. ORTHOMIN	25.5	2	115	30		2.4	15
ad. tGBM	24.6	2	13	97		9.3	50
ad. tBBM	40.6	3	59	64	15	1.8	12

Table 6.5 Times for LNS_3937 with pILU(0.05) in the first trial

using the final preconditioner is of great interest. We have therefore included the times of the last run and largest k used. Once again, a member of the EN-like family is preferred. With pILU(1.56e-3), the adaptive tEN is the fastest method.

We include another example, which shows the usefulness of the preconditioner quality check. The quality of the preconditioner in the previous example, where we use a drop tolerance of 0.05, was not low enough to be detected by the estimates described in Section 6.4. Early termination is caused through detecting divergence (for ad. tEN, tBEN, tGEN, and tGBM) or stagnation (for ad. tSEN, ORTHOMIN and tBBM). The lone exception is the ad. tEN, where the preconditioner quality check leads to termination after 7 iterations instead of 10 iterations due to divergence, when this option is turned off.

For the second example, we degrade the quality of the preconditioner by choosing a higher drop tolerance of 0.08. If we run the adaptive codes as described with the quality check turned on, all EN-like and Broyden methods detect in the first iteration step the low quality of the preconditioner and reevaluate a new preconditioner. CGS breaks down after 223 iterations due to a divide by a near zero value, and BiCGSTAB is terminated after 300 iterations, since it reaches the maximal number of iterations (see Table 6.6). If we switch off the preconditioner quality estimates, all EN-like and Broyden methods perform more iterations with the first preconditioner with the most significant increases for the ad. tSEN, the ad. ORTHOMIN and the ad. tBBM. The actual times and number of iterations for both runs are given in Table 6.6. Clearly, the quality check saves a significant amount of time.

Method	no. of trials	total time		no. of its., trial 1		last run		
		w. ch.	w/o ch.	w. ch.	w/o ch.	its.	sol. time	k
CGS	2	-	33.2	-	223	31	3.9	-
BiCGSTAB	2	-	39.5	-	300	29	3.6	-
ad. tEN	2	16.1	16.8	1	10	28	3.9	13
ad. tBEN	2	16.5	17.2	1	10	31	4.3	17
ad. tSEN	2	16.0	28.9	1	83	27	3.9	15
ad. tGEN	2	21.2	21.9	1	10	61	9.2	26
ad. ORTHOMIN	2	15.8	21.7	1	89	45	3.6	26
ad. tGBM	4	52.6	54.0	1	48	4	0.6	1
ad. tBBM	2	18.6	21.5	1	63	83	6.4	38

Table 6.6 Times for LNS_3937 with pILU(0.08) in the first trial

6.6 Preconditioning with Iterative Methods

In this section, we present a few experiments using an iterative method as a preconditioner. We use Problem 1, and consider two type of experiments. For the first we run some combinations of EN and GCR or GMRES, respectively, with a fixed number of inner iterations m, but the full method outside. While those combinations are fairly competitive with regard to time, there is a significant difference, however, with regard to memory requirement clearly favoring those involving EN (see Table 6.7). Note also that these times are below those achieved when we apply unpreconditioned CGS and BiCGSTAB to this problem (see Table 6.1).

The second type of experiment chooses a fixed memory requirement of 20 additional memory vectors and runs different combinations (both restarted and truncated) of the above methods (see Table 6.7). For comparison we have added GMRES(17) and eEN(8), which have the same memory requirement. For combinations with restarted outer methods, those involving eEN are faster than eGCR(5)/GMRES(5), which is a form of GMRESR, or the unpreconditioned GMRES(17) and, with one exception, eEN(8). For combinations with truncated outer methods, those with ORTHOMIN (tGCR) as outer method (which includes tGMRESR) fail, whereas those with tEN as outer method achieve the best times and lowest sparse matrix vector operation counts.

Overall, the results show that the use of EN in the context of inner/outer iterations schemes is competitive with GMRESR and superior when memory is limited.

Method	time	no. of its.	no. of smv	add. flops/n	memory vectors
GMRESR(6)	12.5	48	331	8640	107
eEN/GMRES(6)	12.6	27	368	7290	65
eGCR/eEN(5)	10.9	27	351	4860	70
eEN/eEN(5)	11.8	15	390	4590	46
eGCR(5)/GMRES(5)	37.2	203	1215	15834	20
eEN(5)/GMRES(5)	24.2	66	792	9900	20
eGCR(2)/eEN(5)	29.0	77	1001	10164	20
eGCR(5)/eEN(2)	35.2	187	1307	11220	20
eEN(2)/eEN(5)	22.1	30	770	7740	20
eEN(5)/eEN(2)	18.8	50	700	5700	20
tGMRESR, tGCR/eEN	stagnate				20
tEN(5)/GMRES(5)	15.5	40	476	7600	20
tEN(2)/eEN(5)	16.8	23	580	6302	20
tEN(5)/eEN(2)	17.3	43	602	6622	20
GMRES(17)	67.7	1652	1652	66080	20
eEN(8)	31.9	460	920	19320	20
GCRO(6)	24.7	52	308	36244	115
eEN/GMRES(6)-O	23.8	32	378	29856	75
eGCR/eEN(5)-O	18.1	28	328	21700	72
eEN/eEN(5)-O	17.5	17	388	17289	50
eGCR(5)/GMRES(5)-O	29.4	155	774	18135	20
eEN(5)/GMRES(5)-O	27.9	75	746	17175	20
eGCR(2)/eEN(5)-O	27.5	71	852	12425	20
eGCR(5)/eEN(2)-O	37.4	189	1132	20601	20
eEN(2)/eEN(5)-O	22.9	30	710	10350	20
eEN(5)/eEN(2)-O	24.8	63	752	13419	20

Table 6.7 Times for inner/outer methods for Problem 1

Method	time	no. of its.	no. of smv	add. flops/n	memory vectors
GMRESR(5)	5.0	27	159	3240	64
GCRO(5)	4.6	22	108	6094	54
eEN/GMRES(5)	5.8	16	192	3040	42
eEN/GMRES(5)-O	5.7	15	145	6135	40
eGCR/eEN(4)	5.7	18	192	2376	50
eGCR/eEN(4)-O	4.3	13	122	4381	40
eEN/eEN(4)	6.4	11	226	2530	36
eEN/eEN(4)-O	7.0	11	206	6479	36
eGCR(4)/GMRES(9)	12.9	42	420	6720	22
eGCR(4)/GMRES(9)-O	8.5	27	243	6021	22
eEN(4)/GMRES(9)	10.2	17	332	5338	22
eEN(4)/GMRES(9)-O	9.0	15	255	6615	22
eGCR(4)/eEN(4)	12.5	41	443	4346	22
eGCR(4)/eEN(4)-O	7.6	24	240	4248	22
eEN(4)/eEN(4)	9.4	16	338	3296	22
eEN(4)/eEN(4)-O	9.5	15	300	5235	22

Table 6.8 Times for inner/outer schemes for SHERMAN3 with ILU(0)

6.7 Testing the Orthogonality Preserving Methods

In this section, we test the inner/outer iteration schemes that preserve the orthogonality and compare them to the original inner/outer iteration schemes. Applying the methods to Problem 1 in a similar way as the original inner/outer schemes, we achieve mixed results. When we use the full methods outside, there is little improvement in the number of sparse matrix vector operations, and a significant increase in additional flops about a factor of 4. Consequently, e.g., GCRO(6) is twice as slow as GMRESR(6). The use of full methods on the outside makes it necessary to orthogonalize the inner space against an outer space with a dimension that increases with every outer iteration leading to the extreme number of additional flops.

This effect can be controlled to some extent by using restarted (or truncated) methods as outer methods. The results in Table 6.7 show that the orthogonality preserving methods are now competitive with their original counterparts.

Keep in mind that Problem 1 is a sparse matrix with 5 elements per row and a fairly inexpensive sparse matrix vector multiply (about 9n flops) without preconditioning. Since the orthogonality preserving methods in general trade sparse for dense matrix vector multiplications, one would expect better results for a complex sparse matrix vector multiplication and preconditioning step. We therefore consider the Harwell-Boeing matrix SHERMAN3, preconditioned with ILU(0), which requires a forward and a backward solve.

The orthogonality preserving methods fare better here. Even for the full outer methods, we encounter a significant decrease in smv's and a less dramatic increase in additional flops, which is, in the case of GCR as outer method, also caused by a decrease in the number of iterations. The difference becomes even more significant if we use restarted GCR on the outside. The large decrease in the number of iterations leads to a large decrease in smv's and no increase in total flops. The original inner/outer schemes are 50 % slower than the orthogonality preserving schemes.

CHAPTER 7

NONLINEAR METHODS

In the previous chapters, we exploited the relationships between EN-like methods and other better known methods. In particular, many of the techniques used to analyze properties of Broyden's method and improving its performance were exploited to understand and implement effective versions of the EN-like methods. In this chapter, we explore another aspect of Broyden's method to develop a family of EN-like methods for solving nonlinear systems and derive theoretical convergence results.

7.1 Description of Methods

Broyden's method was originally developed as a modification of Newton's method for solving nonlinear systems

$$F(x) = 0. \tag{7.1}$$

As such, it is computationally more efficient per iteration than Newton's method, since it avoids recomputing the Jacobian for each step and replaces the evaluation of the Jacobian with the addition of a rank-one update to the previous approximation of the Jacobian B_k or its inverse H_k. The drawback of this modification is the associated degradation of convergence. Whereas Newton's method possesses q-quadratic convergence, one can show for Broyden's method only superlinear or $2n$-step q-quadratic convergence.

The nonlinear Broyden method is usually presented in the following form:

Algorithm 14. Nonlinear Broyden's method (approximating the Jacobian of F).

Initialization:

x_0, H_0 arbitrary,

For $k = 0, 1, \ldots$:

$$\begin{aligned} Solve\ B_k p_k &= -F(x_k) \\ x_{k+1} &= x_k + p_k \\ q_k &= F(x_{k+1}) - F(x_k) \\ B_{k+1} &= B_k + \frac{(q_k - B_k p_k) g_k^T}{g_k^T p_k} \end{aligned}$$

end

B_k is here an approximation of the Jacobian of F. Approximating the inverse of the Jacobian leads to the following algorithm:

Algorithm 15. Nonlinear Broyden's method.
Initialization:
x_0, H_0 arbitrary,
For $k = 0, 1, \ldots$:

$$\begin{aligned} p_k &= -H_k F(x_k) \\ x_{k+1} &= x_k + p_k \\ q_k &= F(x_{k+1}) - F(x_k) \\ H_{k+1} &= H_k + \frac{(p_k - H_k q_k) f_k^T}{f_k^T q_k} \end{aligned}$$

end

where $f_k = H_k^T g_k$. Note that the two versions of Broyden's method are equivalent if we choose $B_0 = H_0^{-1}$.

Similarly, there are two nonlinear versions of the EN-like method. If we approximate the Jacobian, the algorithm is:

Algorithm 16. Nonlinear EN-like method (approximating the Jacobian of F).
Initialization:
x_0, H_0 arbitrary,

For $k = 0, 1, \ldots$:

$$\begin{aligned} Solve\ \ B_k p_k &= -F(x_k) \\ q_k &= F(x_k + p_k) - F(x_k) \\ B_{k+1} &= B_k + \frac{(q_k - B_k p_k) g_k^T}{g_k^T p_k} \\ Solve\ \ B_{k+1} \tilde{p}_k &= -F(x_k) \\ x_{k+1} &= x_k + \tilde{p}_k \end{aligned}$$

end

If we approximate the inverse of the Jacobian of F, the algorithm is:

Algorithm 17. Nonlinear EN-like method.

Initialization:

x_0, H_0 arbitrary,

For $k = 0, 1, \ldots$:

$$\begin{aligned} p_k &= -H_k F(x_k) \\ q_k &= F(x_k + p_k) - F(x_k) \\ H_{k+1} &= H_k + \frac{(p_k - H_k q_k) f_k^T}{f_k^T q_k} \\ \tilde{p}_k &= -H_{k+1} F(x_k) \\ x_{k+1} &= x_k + \tilde{p}_k \end{aligned}$$

end

with $f_k = H_k^T g_k$. For the EN-like method also, both versions are equivalent if $B_0 = H_0^{-1}$.

Just as in the linear case, one can rewrite one step of the nonlinear EN-like method in a form that resembles the corresponding nonlinear Broyden method.

Theorem 7.1 *One step of the nonlinear EN-like method is equivalent to*

$$\begin{aligned} \tilde{x}_{k+1} &= x_k + p_k \\ x_{k+1} &= \tilde{x}_{k+1} + \alpha_k \hat{p}_k \end{aligned}$$

with

$$\alpha_k = -\frac{f_k^T F(x_k)}{f_k^T (F(\tilde{x}_{k+1}) - F(x_k))} \quad \textit{and} \quad \hat{p}_k = -H_k F(\tilde{x}_{k+1}).$$

Proof: The proof is straightforward. □

7.2 The One-Dimensional Nonlinear EN-like Method

If the class of nonlinear systems is restricted to one-dimensional problems, it is known that Broyden's method is equivalent to the secant method

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{F(x_k) - F(x_{k-1})} F(x_k). \tag{7.2}$$

The one-dimensional nonlinear EN-like method is

$$\begin{aligned} p_k &= -\frac{F(x_k)}{b_k}, \\ b_{k+1} &= \frac{F(x_k + p_k) - F(x_k)}{p_k}, \\ x_{k+1} &= x_k - \frac{F(x_k)}{b_{k+1}}. \end{aligned}$$

It is a special case of the finite-difference Newton's method, which allows a more general choice of p_k (see [11, p. 29]).

The following theorem, which states that the secant method has 2-step q-quadratic convergence, can be found in [11].

Theorem 7.2 *Let $F : (a,b) \to \mathcal{R}$, $F \in C^1((a,b))$, $F' \in Lip_\gamma((a,b))$. Assume $\rho \le |F'(x)| \le \alpha$ for some positive α, ρ, and for all $x \in (a,b)$. Assume $F(x) = 0$ has a solution $x^* \in (a,b)$.*

Then there exists $\eta > 0$ with $|x_0 - x^*| < \eta$, so that the sequence $\{x_k\}$, generated by the secant method, is defined, $x_k \in (a,b)$, $k = 0, 1, ...,$ and $\{x_k\}$ converges two-step q-quadratically to x^*; i.e.,

$$|x_{k+2} - x^*| \le C |x_k - x^*|^2. \tag{7.3}$$

Proof: See [11]. □

Interestingly enough, in the one-dimensional case, the EN-like method has q-quadratic convergence just like Newton's method.

Theorem 7.3 *Let $F : (a, b) \to \mathcal{R}$, $F \in C^1((a, b))$, $F' \in Lip_\gamma((a, b))$. Assume $\rho \leq |F'(x)| \leq \alpha$ for some positive α, ρ, and for all $x \in (a, b)$. Assume $F(x) = 0$ has a solution $x^* \in (a, b)$. Define additionally $b_0 = F(x_0 + \epsilon) - F(x_0))/\epsilon$, with $x_0, x_0 + \epsilon \in (a, b)$.*

Then there exists $\eta > 0$ with $|x_0 - x^*| < \eta$, so that the sequence $\{x_k\}$, generated by the EN-like method, is defined, $x_k, x_k + p_k \in (a, b)$, $k = 0, 1, \ldots,$, and $\{x_k\}$ converges q-quadratically to x^*, i.e.

$$|x_{k+1} - x^*| \leq C|x_k - x^*|^2. \tag{7.4}$$

Proof: This theorem follows from the fact that the one-dimensional EN-like method is a special case of the finite difference Newton's method. The proof for the latter method can be found in [11]. □

Considering the complexity of the two methods, the secant method requires one function evaluation and 5 operations per iteration step, whereas the nonlinear EN-like method requires two function evaluations and 6 operations per iteration step, and is therefore, in view of the fact that it converges twice as fast, slightly more efficient than the secant method.

7.3 Convergence Theory

For the n-dimensional case, a variety of convergence results for Broyden's method can be found in the literature. We first review these and identify the key ideas in their proofs that are needed to prove similar results for the EN-like methods. Throughout the discussion, we assume $B_0 = H_0^{-1}$, which leads to $B_k = H_k^{-1}$. Therefore, the versions that approximate the Jacobian or its inverse are equivalent, and we need consider only the effect on convergence of the form of the vector f_k chosen.

The 2-step q-quadratic convergence property of the secant method in one dimension does not hold for the n-dimensional case. One can, however, show superlinear convergence for the nonlinear Broyden's 'good' method with $f_k = H_k^T p_k$ (GBM) and the nonlinear Broyden's 'bad' method with $f_k = q_k$ (BBM) [11].

Theorem 7.4 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|B_0 - J(x^*)\| \leq \delta$, then nonlinear GBM is well defined and converges superlinearly, in the following sense

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.5}$$

Proof: One shows convergence by proving that $\|B_k - J(x*)\| < 2\delta$ and $\|e_{k+1}\| < \tau\|e_k\|$.

One shows specifically

$$\begin{aligned} \|B_k - J(x^*)\| &\leq (2-\tau^k)\delta,\ k = 0, 1, ..., && (7.6) \\ \|e_k\| &\leq \tau^k\epsilon,\ k = 0, 1, ..., && (7.7) \\ & && (7.8) \end{aligned}$$

with the following conditions for ϵ and δ:

$$\begin{aligned} \beta\delta &\leq \frac{\tau}{2+2\tau}, && (7.9) \\ \gamma\epsilon &\leq 2\frac{1-\tau}{1+\tau}\delta. && (7.10) \end{aligned}$$

The complete proof is given in [11] for the special case $\tau = \frac{1}{2}$. □

Theorem 7.5 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in \mathcal{R}^n$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|H_0 - J(x^*)^{-1}\| \leq \delta$, nonlinear BBM is well defined and converges superlinearly, in the following sense

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.11}$$

Proof: One shows convergence by proving by induction that $\|H_k - J(x^*)^{-1}\| < 2\delta$ and $\|e_{k+1}\| \leq \tau\|e_k\|$ with $\tau \in (0, 1)$.

One proves specifically

$$\begin{aligned} \|H_k - J(x^*)^{-1}\| &\leq (2-\tau^k)\delta,\ k \geq 0, && (7.12) \\ \|e_k\| &\leq \tau^k\epsilon,\ k \geq 0, && (7.13) \\ & && (7.14) \end{aligned}$$

with the following conditions for ϵ and δ:

$$\begin{aligned} \gamma\epsilon &\leq \frac{2(1-\tau)\delta}{(1-\tau^2)\beta\delta + (1+\tau)\beta^2}, && (7.15) \\ 2\alpha\delta + (\frac{\beta}{2} + \delta)\gamma\epsilon &\leq \tau. && (7.16) \end{aligned}$$

The proof is similar to the proof of Theorem 7.4 (see [11]) and the proof for nonlinear BEN, which is given below. □

In [24], Gay went a step further and showed local $2n$-step q-quadratic convergence for nonlinear GBM, using the finite termination property of Broyden's method.

Theorem 7.6 *Let $F : \mathcal{R}^n \rightarrow \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, $J \in Lip_\gamma(D)$, and $F(x) - F(y) = \int_0^1 J(y + t(x-y))(x-y)dt$ for $x, y \in D$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|B_0 - J(x^*)\| \leq \delta$, nonlinear GBM is well defined and converges 2n-step q-quadratically; i.e.,

$$\|x_{k+2n} - x^*\| \leq C_1 \|x_k - x^*\|^2. \tag{7.17}$$

Proof: See [24]. □

Theorem 7.7 *Let $F : \mathcal{R}^n \rightarrow \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, $J \in Lip_\gamma(D)$, and $F(x) - F(y) = \int_0^1 J(y + t(x-y))(x-y)dt$ for $x, y \in D$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|H_0 - J(x^*)^{-1}\| \leq \delta$, nonlinear BBM is well defined and converges 2n-step q-quadratically, i.e.

$$\|x_{k+2n} - x^*\| \leq C_1 \|x_k - x^*\|^2. \tag{7.18}$$

Proof: See [24]. □

For each of the two choices of f_k in Broyden's method there is a corresponding EN-like method: nonlinear GEN uses $f_k = H_k^T p_k$ and corresponds to nonlinear GBM; nonlinear BEN uses $f_k = q_k$ and corresponds to nonlinear BBM. As in the linear case, these relationships can be exploited to show similar convergence results for nonlinear GEN and BEN.

The following lemmas are useful in evaluating some of the bounds that occur in the proofs of the theorems.

Lemma 7.1 *Let $F : \mathcal{R}^n \rightarrow \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ open and convex, $J(x) \in Lip_\gamma(D)$. Then, for any $u, v \in D$,*

$$\|F(v) - F(u) - J(x)(v-u)\| \leq \gamma \frac{\|v-x\| + \|u-x\|}{2} \|v-u\|. \tag{7.19}$$

Proof: See [11], p. 77. □

Lemma 7.2 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ open and convex, $J(x) \in Lip_\gamma(D)$. If additionally $J(x)^{-1}$, then there exists $\epsilon > 0$, $0 < \alpha < \beta$, such that*

$$\alpha\|v-u\| \leq \|F(v)-F(u)\| \leq \beta\|v-u\| \tag{7.20}$$

for all $u, v \in D$ for which $max(\|v-x\|, \|u-x\|) \leq \epsilon$.

Proof: See [11], p. 77. □

The following lemma gives an estimate of the norm of the difference between the approximation B_k of the Jacobian of F at the solution, $J(x^*)$, and $J(x^*)$ itself, and the corresponding inverses.

Lemma 7.3 *For $f_k = H_k^T p_k$:*

$$\|B_{k+1} - J(x^*)\| \leq \|B_k - J(x^*)\| + \frac{\|q_k - J(x*)p_k\|}{\|p_k\|}. \tag{7.21}$$

For $f_k = q_k$:

$$\|H_{k+1} - J(x^*)^{-1}\| \leq \|H_k - J(x^*)^{-1}\| + \frac{\|p_k - J(x*)^{-1}q_k\|}{\|q_k\|} \tag{7.22}$$

Proof: See [11], p. 176. □

Note that an increase in the norms of the differences as k increases is possible (as opposed to the linear case where it is not). Therefore, the proofs of the convergence results must guarantee that the norms of the second terms on the righthand sides of (7.21) and (7.22) remain small.

Since the proof of superlinear convergence for the EN-like method is quite lengthy, we break it into two parts. We will first prove convergence (Theorems 7.8 and 7.9) for nonlinear GEN and BEN, and then superlinear convergence (Theorems 7.10 and 7.11).

Consider first nonlinear GEN.

Theorem 7.8 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists with $\|J(x^*)^{-1}\| \leq \beta$, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|B_0 - J(x^*)\| \leq \delta$, nonlinear GEN is well defined and converges.

Proof: Define

$$
\begin{aligned}
\tilde{x}_{k+1} &:= x_k + p_k, \\
e_k &:= x_k - x^*, \\
\tilde{e}_k &:= \tilde{x}_k - x^*.
\end{aligned}
$$

One proves by induction that $\|B_k - J(x*)\| < 2\delta$ and $\|e_{k+1}\| < \tau^2\|e_k\|$. Specifically, we show that

$$\|B_k - J(x^*)\| \leq (2 - \tau^{2k})\delta, \; k \geq 0, \tag{7.23}$$

$$\|e_k\| \leq \tau^{2k}\epsilon, \; k \geq 0, \tag{7.24}$$

$$\|\tilde{e}_k\| \leq \tau^{2k-1}\epsilon, \; k \geq 1, \tag{7.25}$$

where ϵ and δ are chosen so that

$$\beta\delta \leq \frac{\tau^2}{3 + 2\tau^2}, \tag{7.26}$$

$$\gamma\epsilon \leq 2(1-\tau)\delta. \tag{7.27}$$

Clearly, the assertion is true for $k = 0$, due to the assumptions, and for (7.25) with $k = 1$, the proof for which is omitted here since it is a special case of the proof for general k given below.

The following equations and inequalities are needed for the proof.

$$B_k\tilde{e}_{k+1} = B_k e_k - F(x_k) + F(x^*) \tag{7.28}$$

and, for nonsingular B_k,

$$\tilde{e}_{k+1} = B_k^{-1}((B_k - J(x^*))e_k + (-F(x_k) + F(x^*) + J(x^*)e_k)). \tag{7.29}$$

Consequently, with Lemma 7.1,

$$\|\tilde{e}_{k+1}\| \leq \|B_k^{-1}\|(\|B_k - J(x^*)\|\|e_k\| + \frac{\gamma}{2}\|e_k\|^2). \tag{7.30}$$

One can show similarly for nonsingular B_{k+1} that

$$\|e_{k+1}\| \leq \|B_{k+1}^{-1}\|(\|B_{k+1} - J(x^*)\|\|e_k\| + \frac{\gamma}{2}\|e_k\|^2). \tag{7.31}$$

Let us examine $\|B_{k+1} - J(x^*)\|$. According to Lemma 7.1,

$$\begin{aligned}
\|q_k - J(x^*)p_k\| &= \|F(x_k + p_k) - F(x_k) - J(x^*)p_k\| && (7.32)\\
&\leq \frac{\gamma}{2}(\|\tilde{e}_{k+1}\| + \|e_k\|)\|p_k\|. && (7.33)
\end{aligned}$$

Combining this result with Lemma 7.3 leads to

$$\|B_{k+1} - J(x^*)\| \leq \|B_k - J(x^*)\| + \frac{\gamma}{2}(\|\tilde{e}_{k+1}\| + \|e_k\|). \qquad (7.34)$$

We also need an estimate for $\|B_k^{-1}\|$. According to (7.23), (7.26), and the Perturbation Lemma [33, p. 45], B_k^{-1} exists, and

$$\begin{aligned}
\|B_k^{-1}\| &\leq \frac{\beta}{1 - 2\beta\delta} && (7.35)\\
&\leq (1 + \frac{2}{3}\tau^2)\beta. && (7.36)
\end{aligned}$$

Assume that the assertion is true, for $i = 1, ..., k$. Then, with (7.30), (7.36), (7.23), (7.24), (7.26), and (7.27) it follows that

$$\begin{aligned}
\|\tilde{e}_{k+1}\| &\leq (1 + \frac{2}{3}\tau^2)\beta((2 - \tau^{2k})\delta + \frac{\gamma}{2}\tau^{2k}\epsilon)\|e_k\|\\
&\leq (1 + \frac{2}{3}\tau^2)\beta(2\delta - \tau^{2k}\delta + (1 - \tau)\tau^{2k}\delta)\|e_k\|\\
&\leq \frac{\tau^2}{3}(2 - \tau^{2k+1})\|e_k\|\\
&\leq \tau\|e_k\|\\
&\leq \tau^{2k+1}\epsilon.
\end{aligned}$$

Now, with (7.34), (7.23), (7.24), (7.25), and (7.27), we have

$$\begin{aligned}
\|B_{k+1} - J(x^*)\| &\leq (2 - \tau^{2k})\delta + \frac{\gamma}{2}(\tau^{2k+1}\epsilon + \tau^{2k}\epsilon)\\
&\leq 2\delta - \tau^{2k})\delta + (1 - \tau)\delta(\tau + 1)\tau^{2k}\\
&= (2 - \tau^{2(k+1)})\delta.
\end{aligned}$$

Finally, using (7.31), (7.36), (7.23), (7.24), (7.26), and (7.27),

$$\begin{aligned}
\|e_{k+1}\| &\leq (1 + \frac{2}{3}\tau^2)\beta((2 - \tau^{2(k+1)})\delta + \frac{\gamma}{2}\tau^{2k}\epsilon)\|e_k\|\\
&= (1 + \frac{2}{3}\tau^2)\beta\delta(2 - \tau^{2k+2} + (1 - \tau)\tau^{2k})\|e_k\|\\
&\leq \tau^2\|e_k\|\frac{1}{3}(2 - \tau^{2k+2} + \tau^{2k} - \tau^{2k-1})\\
&\leq \tau^2\|e_k\|\\
&\leq \tau^{2(k+1)}\epsilon,
\end{aligned}$$

which proves that

$$\lim_{k\to\infty} \|e_k\| = \lim_{k\to\infty} \tau^{2k}\epsilon = 0,$$

since $\tau < 1$. $\square$

Note that the induction part of the convergence proof for nonlinear GEN indicates that it converges twice as fast as nonlinear GBM, if the same τ is chosen and ϵ and δ fulfill the conditions for both methods.

There is a similar theorem for nonlinear BEN.

Theorem 7.9 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$, $J(x^*)^{-1}$ exists with $\|J(x^*)^{-1}\| \leq \beta$, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|H_0 - J(x^*)^{-1}\| \leq \delta$, nonlinear BEN is well defined and converges.

Proof: The proof is similar to the proof of Theorem 7.8. Define $\alpha := \|J(x^*)\|$.

Similarly to the proof of the previous theorem, we show by induction that $\|H_k - J(x^*)^{-1}\| < 2\delta$ and $\|e_{k+1}\| \leq \tau^2\|e_k\|$.

We prove specifically that

$$\|H_k - J(x^*)^{-1}\| \leq (2-\tau^{2k})\delta,\ k \geq 0, \tag{7.37}$$

$$\|\tilde{e}_k\| \leq \tau^{2k-1}\epsilon,\ k \geq 1, \tag{7.38}$$

$$\|e_k\| \leq \tau^{2k}\epsilon,\ k \geq 0. \tag{7.39}$$

We assume for γ and ϵ

$$\gamma\epsilon \leq \frac{2(1-\tau)\delta}{(1-\tau^2)\beta\delta + \beta^2}, \tag{7.40}$$

$$2\alpha\delta + (\frac{\beta}{2} + \delta)\gamma\epsilon \leq \tau^2. \tag{7.41}$$

Clearly, the assertion is true for $k = 0$, due to the assumptions of the theorem. The proof for (7.38) with $k = 1$ is omitted here, since it is a special case for the proof for general k, which is given below.

We will need the following equations and inequalities for the proof.

$$\tilde{e}_{k+1} = e_k - H_kJ(x^*)e_k - H_kF(x_k) + H_kF(x^*) + H_kJ(x^*)e_k. \tag{7.42}$$

Consequently, with Lemma 7.1,

$$\|\tilde{e}_{k+1}\| \leq \|J(x^*)\|\|H_k - J(x^*)^{-1}\|\|e_k\| + \frac{\gamma}{2}\|H_k\|\|e_k\|^2). \tag{7.43}$$

Since

$$\|H_k\| \leq \|J(x^*)^{-1}\| + \|H_k - J(x^*)^{-1}\|, \tag{7.44}$$

one obtains

$$\|\tilde{e}_{k+1}\| \leq \|H_k - J(x^*)^{-1}\|\|e_k\|(\alpha + \frac{\gamma}{2}\|e_k\|) + \frac{\beta\gamma}{2}\|e_k\|^2. \tag{7.45}$$

One can show similarly

$$\|e_{k+1}\| \leq \|H_{k+1} - J(x^*)^{-1}\|\|e_k\|(\alpha + \frac{\gamma}{2}\|e_k\|) + \frac{\beta\gamma}{2}\|e_k\|^2. \tag{7.46}$$

Let us examine $\|H_{k+1} - J(x^*)^{-1}\|$. Using Lemma 7.1,

$$\begin{aligned}
\|q_k\| &= \|F(x_k + p_k) - F(x_k) - J(x^*)p_k + J(x^*)p_k\| \\
&\geq \|J(x^*)p_k\| - \frac{\gamma}{2}(\|\tilde{e}_{k+1}\| + \|e_k\|)\|p_k\| \\
&\geq \left(\frac{1}{\beta} - \frac{\gamma}{2}(\|\tilde{e}_{k+1}\| + \|e_k\|)\right)\|p_k\|.
\end{aligned}$$

Combining this result with Lemma 7.3 and (7.33) leads to

$$\|H_{k+1} - J(x^*)^{-1}\| \leq \|H_k - J(x^*)^{-1}\| + \frac{\beta^2\gamma(\|\tilde{e}_{k+1}\| + \|e_k\|)}{2 - \beta\gamma(\|\tilde{e}_{k+1}\| + \|e_k\|)}. \tag{7.47}$$

Assume that the assertion is true, for $i = 1, ..., k$.

Using (7.45), (7.37), (7.39), and (7.41) leads to

$$\begin{aligned}
\|\tilde{e}_{k+1}\| &\leq (2 - \tau^{2k})\delta\|e_k\|(\alpha + \frac{\gamma}{2}\tau^{2k}\epsilon) + \beta\frac{\gamma}{2}\tau^{2k}\epsilon\|e_k\| \\
&= \tau\|e_k\|\frac{1}{\tau}(2\alpha\delta - \tau^{2k}\alpha\delta + \tau^{2k}\gamma\delta\epsilon - \frac{\tau^{4k}}{2}\gamma\delta\epsilon + \frac{\tau^{2k}}{2}\beta\gamma\epsilon) \\
&\leq \tau\|e_k\|\frac{1}{\tau}(2\alpha\delta + \gamma\delta\epsilon + \frac{1}{2}\beta\gamma\epsilon) \\
&\leq \tau\|e_k\| \\
&\leq \tau^{2k+1}\epsilon.
\end{aligned}$$

With (7.47), (7.37), (7.39), (7.38), and (7.40),

$$\begin{aligned}
\|H_{k+1} - J(x^*)^{-1}\| &\leq ((2 - \tau^{2k})\delta + \frac{\beta^2\gamma\epsilon(\tau^{2k+1} + \tau^{2k})}{2 - \beta\gamma\epsilon(\tau^{2k+1} + \tau^{2k})} \\
&\leq (2 - 2^{2(k+1)})\delta + \tau^{2k}\left((\tau^2 - 1)\delta + \frac{2(1 - \tau^2)\delta\beta^2}{2\beta^2 + 2(1 - \tau^2)\beta\delta - 2(1 - \tau^2)\beta\delta}\right) \\
&\leq (2 - 2^{2(k+1)})\delta.
\end{aligned}$$

Then, using (7.46), (7.37), (7.39), and (7.41),

$$\begin{aligned}
\|e_{k+1}\| &\leq (2-\tau^{2(k+1)})\delta\|e_k\|(\alpha+\tau^{2k}\frac{\gamma}{2}\epsilon)+\beta\frac{\gamma}{2}\tau^{2k}\epsilon\|e_k\| \\
&\leq \tau^2\|e_k\|\frac{1}{\tau^2}(2\alpha\delta+\gamma\delta\epsilon+\frac{1}{2}\beta\gamma\epsilon) \\
&\leq \tau^2\|e_k\| \\
&\leq \tau^{2(k+1)}\epsilon,
\end{aligned}$$

which concludes the proof for linear convergence. □

Just as for nonlinear GBM and GEN, the induction part of the convergence proof for nonlinear BEN indicates that it converges twice as fast as nonlinear BBM under equivalent conditions.

Having established convergence for both nonlinear GEN and BEN, we now demonstrate that it is, in fact, superlinear.

Nonlinear GEN and BEN share an important property, which is summarized in the following lemma and is useful in establishing the superlinear convergence results. Specifically, the lemma shows that for nonlinear GEN the approximation of the Jacobian B_k converges towards the Jacobian in the direction of p_k for increasing k, and an equivalent result for nonlinear BEN, where we, however, consider the inverses of those matrices.

Lemma 7.4 *Under the assumptions of Theorem 7.8, for $f_k = H_k^T p_k$:*

$$\lim_{k\to\infty}\frac{\|(B_k - J(x^*))p_k\|}{\|p_k\|} = 0. \tag{7.48}$$

Under the assumptions of Theorem 7.9, for $f_k = q_k$:

$$\lim_{k\to\infty}\frac{\|(H_k - J(x^*)^{-1})q_k\|}{\|q_k\|} = 0. \tag{7.49}$$

Proof: See [11], pp. 183-184. □

Theorems 7.10 and 7.11 demonstrate the superlinear convergence for nonlinear GEN and BEN, respectively.

Theorem 7.10 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|B_0 - J(x^*)\| \leq \delta$, nonlinear GEN is well defined and converges superlinearly, in the following sense

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.50}$$

Proof: Similarly as in (7.31),

$$\begin{aligned}
\frac{\|e_{k+1}\|}{\|e_k\|} &\leq \|B_{k+1}^{-1}\|\left(\frac{\|(B_{k+1} - J(x^*))e_k\|}{\|e_k\|} + \frac{\gamma}{2}\|e_k\|\right) \\
&\leq \frac{3}{2}\beta\left(\frac{\|(B_k - J(x^*))e_k\|}{\|e_k\|} + \frac{\|F(x_k + p_k)\|}{\|p_k\|} + \frac{\gamma}{2}\|e_k\|\right).
\end{aligned}$$

We need to examine further the terms on the righthand side of the previous inequality.

$$-F(x_k + p_k) = (B_k - J(x^*))p_k - F(x_k + p_k) + F(x_k) + J(x^*)p_k. \tag{7.51}$$

This leads to

$$\frac{\|F(x_k + p_k)\|}{\|p_k\|} \leq \frac{\|(B_k - J(x^*))p_k\|}{\|p_k\|} + \frac{\gamma}{2}(\|\tilde{e}_{k+1}\| + \|e_k\|).$$

Since $\tilde{e}_{k+1}$ and e_k converge to zero for increasing k, and using Lemma 7.4,

$$\lim_{k\to\infty} \frac{\|F(x_k + p_k)\|}{\|p_k\|} = 0. \tag{7.52}$$

According to Lemma 7.2, there exist $c > 0$, $k_0 \geq 0$ so that

$$\|F(x_k + p_k)\| = \|F(x_k + p_k) - F(x^*)\| \geq c\|\tilde{e}_{k+1}\| \tag{7.53}$$

for all $k \geq k_0$.

$$\begin{aligned}
0 = \lim_{k\to\infty} \frac{\|F(x_k + p_k)\|}{\|p_k\|} &\geq \lim_{k\to\infty} c\frac{\|\tilde{e}_{k+1}\|}{\|p_k\|} \\
&\geq \lim_{k\to\infty} \frac{c\|\tilde{e}_{k+1}\|}{\|e_k\| + \|\tilde{e}_{k+1}\|} \\
&= \lim_{k\to\infty} \frac{c\|\tilde{e}_{k+1}\| \ \|e_k\|}{1 + \|\tilde{e}_{k+1}\| \ \|e_k\|}.
\end{aligned}$$

Consequently,

$$\lim_{k\to\infty} \frac{\|\tilde{e}_{k+1}\|}{\|e_k\|} = 0. \tag{7.54}$$

Since $\tilde{e}_{k+1} = e_k + p_k$,

$$\frac{\|(B_k - J(x^*))e_k\|}{\|e_k\|} \leq \|B_k - J(x^*)\|\frac{\|\tilde{e}_{k+1}\|}{\|e_k\|} + \frac{\|(B_k - J(x^*))p_k\|}{\|p_k\|} \cdot \frac{\|\tilde{e}_{k+1}\| + \|e_k\|}{\|e_k\|}.$$

This proves

$$\lim_{k\to\infty} \frac{\|(B_k - J(x^*))e_k\|}{\|e_k\|} = 0. \tag{7.55}$$

Combining these equations leads to

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.56}$$

□

Theorem 7.11 *Let $F : \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in \mathcal{R}^n$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, and $J \in Lip_\gamma(D)$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \le \epsilon$ and $\|H_0 - J(x^*)^{-1}\| \le \delta$, nonlinear BEN is well defined and converges superlinearly, in the following sense

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.57}$$

Proof: Using the definitions introduced for the proof of Theorem 7.9,

$$\begin{aligned}
\frac{\|e_{k+1}\|}{\|e_k\|} &\le \left(\frac{\|(I - H_{k+1}J(x^*))e_k\|}{\|e_k\|} + \frac{\gamma}{2}\|H_{k+1}\|\|e_k\|\right) && (7.58)\\
&\le \frac{\|(I - H_kJ(x^*))e_k\|}{\|e_k\|} + \alpha\frac{\|H_kF(x_k + p_k)\|}{\|q_k\|} + \frac{\gamma}{2}(\beta + 2\delta)\|e_k\|). && (7.59)
\end{aligned}$$

We need to examine further the terms on the righthand side of the previous inequality.

$$\begin{aligned}
(H_k - J(x^*)^{-1})q_k &= J(x^*)^{-1}(J(x^*)H_k - I)(F(x_k + p_k) - F(x_k))\\
&= J(x^*)^{-1}(J(x^*)p_k - F(x_k + p_k) + F(x_k)) + H_kF(x_k + p_k).
\end{aligned}$$

This leads to

$$\begin{aligned}
\frac{\|(H_k - J(x^*)^{-1})q_k\|}{\|q_k\|} &\ge \frac{\|H_kF(x_k + p_k)\|}{\|q_k\|} - \frac{\gamma}{2}\frac{\|p_k\|}{\|q_k\|}(\|\tilde{e}_{k+1}\| + \|e_k\|)\\
&\ge \frac{\|H_kF(x_k + p_k)\|}{\|q_k\|} - \frac{2\beta\gamma}{2 - 3\beta\gamma\epsilon}(\|\tilde{e}_{k+1}\| + \|e_k\|).
\end{aligned}$$

Together with Lemma 7.4 and Theorem 7.9, one obtains

$$\lim_{k\to\infty} \frac{\|H_kF(x_k + p_k)\|}{\|q_k\|} = 0. \tag{7.60}$$

Since

$$q_k = F(x_k + p_k) - F(x_k) - J(x^*)p_k + J(x^*)p_k), \tag{7.61}$$

and according to Lemma 4.22, (7.37) and (7.41), H_k^{-1} exists, and

$$\|H_k^{-1}\| \leq \frac{\alpha}{1-2\alpha\delta}, \tag{7.62}$$

$$\lim_{k\to\infty} \frac{\|F(x_k+p_k)\|}{\|p_k\|} = \lim_{k\to\infty} \frac{\alpha}{1-2\alpha\delta}(\alpha+\frac{3}{4}\gamma\epsilon)\frac{\|F(x_k+p_k)\|}{\|q_k\|} = 0. \tag{7.63}$$

Using the same reasoning as in the proof of Theorem 7.8 (see (7.53)ff.), one obtains

$$\lim_{k\to\infty} \frac{\|\tilde{e}_{k+1}\|}{\|e_k\|} = 0. \tag{7.64}$$

Now,

$$\begin{aligned} \|(I-H_kJ(x^*))e_k\| &= \|e_k+p_k+H_k(F(x_k)-F(x^*)-J(x^*)e_k)\| \\ &\leq \|\tilde{e}_{k+1}\| + (\beta+2\delta)\frac{\gamma}{2}\|e_k\|^2. \end{aligned}$$

Combining these equations with (7.59) leads to

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0. \tag{7.65}$$

□

Just as Gay [24] showed local $2n$-step q-quadratic convergence for Broyden's method, using the finite termination property of Broyden's method and the local convergence properties in [6], we show local n-step q-quadratic convergence for nonlinear GEN and BEN in Theorems 7.12 and 7.13, respectively.

Theorem 7.12 *Let $F: \mathcal{R}^n \to \mathcal{R}^n$, $F \in C^1(D)$, $D \subset \mathcal{R}^n$ be open and convex. Assume there exists $x^* \in D$ so that $F(x^*) = 0$, $J(x^*)^{-1}$ exists, $J \in Lip_\gamma(D)$, and $F(x) - F(y) = \int_0^1 J(y+t(x-y))(x-y)dt$ for $x, y \in D$.*

Then, there exist $\epsilon, \delta > 0$, so that for $\|x_0 - x^*\| \leq \epsilon$ and $\|B_0 - J(x^*)\| \leq \delta$, nonlinear GEN is well defined and converges n-step q-quadratically, i.e.

$$\|x_{k+n} - x^*\| \leq C_2\|x_k - x^*\|^2, \tag{7.66}$$

Proof: We use the following definitions:

$$\alpha := \|J(x^*)\|, \tag{7.67}$$

$$\beta := \|J(x^*)^{-1}\|, \tag{7.68}$$

$$\begin{aligned}
\hat{F}(x) &:= J(x^*)(x-x^*), &(7.69)\\
\hat{x}_0 &:= x_k, &(7.70)\\
\hat{B}_0 &:= B_k, &(7.71)\\
\hat{p}_j &:= -\hat{B}_j^{-1}\hat{F}(\hat{x}_j), &(7.72)\\
\hat{q}_j &:= \hat{F}(\hat{x}_j+\hat{p}_j)-\hat{F}(\hat{x}_j), &(7.73)\\
\hat{B}_{j+1} &:= \hat{B}_j + \frac{(\hat{q}_j-\hat{B}_j\hat{p}_j)\hat{p}_j^T}{\hat{p}_j^T\hat{p}_j}, &(7.74)\\
\hat{x}_{j+1} &:= \hat{x}_j + \hat{B}_{j+1}^{-1}\hat{F}(\hat{x}_j). &(7.75)
\end{aligned}$$

Additionally, assume (7.26) and (7.27).

If we use GEN to solve the linear system $\hat{F}(x)=0$, then, according to the finite termination property that was proved in Chapter 4, there exists $0 \le m \le n$ with $\hat{x}_m = x^*$.

We will show by induction that for $0 \le j \le n$,

$$\begin{aligned}
\|x_{k+i}-\hat{x}_i\| &\le C_{4,i}\|x_k-x^*\|^2, &(7.76)\\
\|B_{k+i+1}-\hat{B}_{i+1}\|\|x_{k+i}-x^*\| &\le C_{3,i}\|x_k-x^*\|^2. &(7.77)
\end{aligned}$$

Clearly, (7.76) is true for $i=0$ due to (7.70), and (7.77) can be shown for $i=0$ just as for more general i, as is done below using (7.70) and (7.71).

Assume that (7.77) holds for $0 \le i \le j-1$, and (7.76) for $0 \le i \le j$.

$$\|B_{k+j+1}-\hat{B}_{j+1}\| \le \|B_{k+j}-\hat{B}_j\| + \|\frac{(q_{k+j}-B_{k+j}p_{k+j})p_{k+j}^T}{p_{k+j}^Tp_{k+j}} - \frac{(\hat{q}_j-\hat{B}_j\hat{p}_j)\hat{p}_j^T}{\hat{p}_j^T\hat{p}_j}\|. \qquad (7.78)$$

The first term on the righthand side can be dealt with through the induction hypothesis. For further investigation of the second term, we define $J_k := \int_0^1 J(x_{k+j}+tp_{k+j})dt$.

Now

$$\begin{aligned}
\|J_k - J(x^*)\| &\le \int_0^1 \|J((x_{k+j}+tp_{k+j})-J(x^*)\|dt\\
&\le \gamma\|x_{k+j}-x^*\|. &(7.79)
\end{aligned}$$

Note that $\hat{q}_j = J(x^*)\hat{p}_j$ and $q_{k+j} = J_kp_{k+j}$.

$$\begin{aligned}
\|\frac{(q_{k+j}-B_{k+j}p_{k+j})p_{k+j}^T}{p_{k+j}^Tp_{k+j}} - \frac{(\hat{q}_j-\hat{B}_j\hat{p}_j)\hat{p}_j^T}{\hat{p}_j^T\hat{p}_j}\| &\le \|[(J_k-J(x^*))-(B_{k+j}-\hat{B}_j)]\frac{p_{k+j}p_{k+j}^T}{p_{k+j}^Tp_{k+j}}\|\\
&\quad +\|(J(x^*)-\hat{B}_j)\left(\frac{p_{k+j}p_{k+j}^T}{p_{k+j}^Tp_{k+j}} - \frac{\hat{p}_j\hat{p}_j^T}{\hat{p}_j^T\hat{p}_j}\right)\|
\end{aligned}$$

$$
\begin{aligned}
&\leq \gamma\|x_k - x^*\| + \|B_{k+j} - \hat{B}_j\| && (7.80)\\
&\quad +8\delta\frac{\|p_{k+j} - \hat{p}_j\|}{\|p_{k+j}\|}, && (7.81)
\end{aligned}
$$

since $\|J(x^*) - \hat{B}_j\| \leq \|J(x^*) - \hat{B}_0\| = \|J(x^*) - B_k\| \leq 2\delta$ due to $\hat{F}$ being linear. Furthermore,

$$
\begin{aligned}
\|(\frac{p_{k+j}p_{k+j}^T}{p_{k+j}^T p_{k+j}} - \frac{\hat{p}_j\hat{p}_j^T}{\hat{p}_j^T\hat{p}_j}\| &= 2\|\frac{p_{k+j}}{\|p_{k+j}\|} - \frac{\hat{p}_j}{\|\hat{p}_j\|}\|\\
&\leq \frac{2}{\|p_{k+j}\|}(\|p_{k+j} - \hat{p}_j\| + |\|\hat{p}_j\| - \|p_{k+j}\||)\\
&\leq 4\frac{\|p_{k+j} - \hat{p}_j\|}{\|p_{k+j}\|}. && (7.82)
\end{aligned}
$$

In order to obtain an upper limit on $\|p_{k+j} - \hat{p}_j\|$, we need

$$
\begin{aligned}
\|F(x_{k+j}) - \hat{F}(x_{k+j})\| &= \|\int_0^1 [J(x^* + t(x_{k+j} - x^*) - J(x^*)](x_{k+j} - x^*)dt\|\\
&\leq \gamma\|x_{k+j} - x^*\|\int_0^1 \|x_{k+j} - x^*\|tdt\\
&\leq \frac{\gamma}{2}\|x_{k+j} - x^*\|^2
\end{aligned}
$$

Then, using (7.24) and (7.76),

$$
\begin{aligned}
\|F(x_{k+j}) - \hat{F}(\hat{x}_j)\| &\leq \|F(x_{k+j}) - \hat{F}(x_{k+j})\| + \|\hat{F}(x_{k+j}) - \hat{F}(\hat{x}_j)\|\\
&\leq \frac{\gamma}{2}\|x_{k+j} - x^*\|^2 + \beta\|x_{k+j} - \hat{x}_j\|\\
&\leq C_{5,j}\|x_k - x^*\|^2. && (7.83)
\end{aligned}
$$

Since

$$
\begin{aligned}
\|F(x_{k+j})\| &= \|F(x_{k+j}) - F(x^*) - J(x^*)(x_{k+j} - x^*) + J(x^*)(x_{k+j} - x^*)\|\\
&\leq (\frac{\gamma\epsilon}{2} + \alpha)\|x_{k+j} - x^*\|, && (7.84)
\end{aligned}
$$

one obtains together with (7.77), (7.83), and (7.35)

$$
\begin{aligned}
\|p_{k+j} - \hat{p}_j\| &= \|B_{k+j}^{-1}F(x_{k+j}) - \hat{B}_j^{-1}\hat{F}(\hat{x}_j)\|\\
&\leq \|B_{k+j}^{-1}\|\|B_{k+j} - \hat{B}_j\|\|\hat{B}_j^{-1}\|\|F(x_{k+j})\| + \|\hat{B}_j^{-1}\|\|F(x_{k+j}) - \hat{F}(\hat{x}_j)\|\\
&\leq (\frac{\gamma\epsilon}{2} + \alpha)\|B_{k+j}^{-1}\|\|\hat{B}_j^{-1}\|\|B_{k+j} - \hat{B}_j\|\|x_{k+j-1} - x^*\| + \|\hat{B}_j^{-1}\|\|F(x_{k+j}) - \hat{F}(\hat{x}_j)\|\\
&\leq C_{6,j}\|x_k - x^*\|^2. && (7.85)
\end{aligned}
$$

Now, one can also show that

$$
\|p_{k+j}\| \geq \frac{2 - \beta\gamma\epsilon}{2\beta(\alpha + 2\delta)}\|x_{k+j} - x^*\|. \qquad (7.86)
$$

Combining (7.78) with (7.81), (7.85), and (7.86), we can now prove (7.77).

$$\begin{aligned}\|B_{k+j+1}-\hat{B}_{j+1}\|\|x_{k+j}-x^*\| &\leq 2\|B_{k+j}-\hat{B}_j\|\|x_{k+j-1}-x^*\|+C_{7,j}\|x_k-x^*\|^2\\ &\leq C_{3,j}\|x_k-x^*\|^2.\end{aligned}$$

Now, as for (7.76), we obtain, using (7.77), (7.76), and (7.35),

$$\begin{aligned}\|x_{k+j+1}-\hat{x}_{j+1}\| &\leq \|x_{k+j}-\hat{x}_j\|+\|B_{k+j+1}^{-1}F(x_{k+j})-\hat{B}_{j+1}^{-1}\hat{F}(\hat{x}_j)\|\\ &\leq \|x_{k+j}-\hat{x}_j\|+\|B_{k+j+1}^{-1}(F(x_{k+j})-F(x^*)-J(x^*)(x_{k+j}-x^*))\\ &\quad +B_{k+j+1}^{-1}J(x^*)(x_{k+j}-x^*)-\hat{B}_{j+1}^{-1}\hat{F}(\hat{x}_j)\|\\ &\leq \|x_{k+j}-\hat{x}_j\|+\frac{\gamma}{2}\|B_{k+j+1}^{-1}\|\|x_{k+j}-x^*\|^2\\ &\quad +\alpha\|\hat{B}_{j+1}^{-1}\|(\|B_{k+j+1}^{-1}\|\|B_{k+j+1}-\hat{B}_{j+1}\|\|x_{k+j}-x^*\|+\|x_{k+j}-\hat{x}_j\|)\\ &\leq C_{4,j+1}\|x_k-x^*\|^2.\end{aligned}$$

This leads to

$$\|x_{k+n}-x^*\|\leq\|x_{k+m}-x^*\|=\|x_{k+m}-\hat{x}_m\|\leq C_2\|x_k-x^*\|^2,$$

where $C_2=C_{4,n}$ independent of k. □

Theorem 7.13 *Let $F:\mathcal{R}^n\to\mathcal{R}^n$, $F\in C^1(D)$, $D\subset\mathcal{R}^n$ be open and convex. Assume there exists $x^*\in D$ so that $F(x^*)=0$, $J(x^*)^{-1}$ exists, $J\in Lip_\gamma(D)$, and $F(x)-F(y)=\int_0^1 J(y+t(x-y))(x-y)dt$ for $x,y\in D$.*

Then, there exist $\epsilon,\delta>0$, so that for $\|x_0-x^*\|\leq\epsilon$ and $\|H_0-J(x^*)^{-1}\|\leq\delta$, nonlinear BEN is well defined and converges n-step q-quadratically; i.e.,

$$\|x_{k+n}-x^*\|\leq C_2\|x_k-x^*\|^2, \tag{7.87}$$

Proof: We will use the definitions given for the proof of the previous theorem, but also $H_j=B_j^{-1}$ and the following definitions:

$$\hat{H}_0 := H_k, \tag{7.88}$$

$$\hat{p}_j := -\hat{H}_j\hat{F}(\hat{x}_j), \tag{7.89}$$

$$\hat{H}_{j+1} := \hat{H}_j+\frac{(\hat{p}_j-\hat{H}_j\hat{q}_j)\hat{q}_j^T}{\hat{q}_j^T\hat{q}_j}, \tag{7.90}$$

$$\hat{x}_{j+1} := \hat{x}_j+\hat{H}_{j+1}\hat{F}(\hat{x}_j). \tag{7.91}$$

Additionally, assume (7.40), (7.41), and

$$\beta\gamma\epsilon < 1. \tag{7.92}$$

We will show by induction that for $0 \le i \le n$,

$$\begin{aligned} \|x_{k+i} - \hat{x}_i\| &\le C_{4,i}\|x_k - x^*\|^2, &(7.93)\\ \|H_{k+i+1} - \hat{H}_{i+1}\|\|x_{k+i} - x^*\| &\le C_{3,i}\|x_k - x^*\|^2. &(7.94)\end{aligned}$$

Clearly, (7.93) is true for $i = 0$ due to (7.70). (7.94) can be proven for $i = 0$, using (7.71) and (7.70), as is done for more general i below.

Assume (7.93) is true for $0 \le i \le j$, and (7.94) for $0 \le i \le j-1$. Also, due to (7.79), (7.92) and Lemma 4.22, J_k^{-1} exists and

$$\|J_k^{-1}\| \le \frac{\beta}{1-\beta\gamma\epsilon}. \tag{7.95}$$

Then using a similar argument as in (7.82)

$$\begin{aligned} \|H_{k+j+1} - \hat{H}_{j+1}\| &= \|H_{k+j} + \frac{(p_{k+j} - H_{k+j}q_{k+j})q_{k+j}^T}{q_{k+j}^T q_{k+j}} - \hat{H}_j - \frac{(\hat{p}_j - \hat{H}_j\hat{q}_j)\hat{q}_j^T}{\hat{q}_j^T\hat{q}_j}\| \\ &\le \|H_{k+j} - \hat{H}_j\| + \|[(J_k^{-1} - J(x^*)^{-1}) - (H_{k+j} - \hat{H}_j)]\frac{q_{k+j}q_{k+j}^T}{q_{k+j}^T q_{k+j}}\| \\ &\quad + \|(J(x^*)^{-1} - \hat{H}_j)\left(\frac{q_{k+j}q_{k+j}^T}{q_{k+j}^T q_{k+j}} - \frac{\hat{q}_j\hat{q}_j^T}{\hat{q}_j^T\hat{q}_j}\right)\| &(7.96)\\ &\le 2\|H_{k+j} - \hat{H}_j\| + \beta\|J_k^{-1}\|\|J_k - J(x^*)\| + 8\delta\frac{\|q_{k+j} - \hat{q}_j\|}{\|q_{k+j}\|}. &(7.97)\end{aligned}$$

Further investigation shows

$$\begin{aligned} \|q_{k+j} - \hat{q}_j\| &= \|J_k p_{k+j} - J(x^*)\hat{p}_j\| \\ &\le \|J_k - J(x^*)\|\|p_{k+j}\| + \alpha\|p_{k+j} - \hat{p}_j\| &(7.98)\end{aligned}$$

and, using (7.84), (7.83), and (7.93),

$$\begin{aligned} \|p_{k+j} - \hat{p}_j\| &\le \|H_{k+j} - \hat{H}_j\|\|F(x_{k+j})\| + \|\hat{H}_j\|\|F(x_{k+j}) - \hat{F}(\hat{x}_j)\| \\ &\le (\frac{\gamma\epsilon}{2} + \alpha)\|H_{k+j} - \hat{H}_j\|\|x_{k+j-1} - x^*\| + (\beta + 2\delta)C_{5,j}\|x_k - x^*\|^2 \\ &\le C_{6,j}\|x_k - x^*\|^2. &(7.99)\end{aligned}$$

Combining (7.97) with (7.98), (7.99), (7.79), (7.94), and

$$\frac{(1-2\alpha\delta)(2-\beta\gamma\epsilon)}{2\alpha\beta}\|x_{k+j} - x^*\| \le \|p_{k+j}\| \le \frac{\beta}{1-\beta\gamma\epsilon}\|q_{k+j}\|, \tag{7.100}$$

we obtain

$$\begin{aligned}
\|H_{k+j+1} - \hat{H}_{j+1}\|\|x_{k+j} - x^*\| &\leq (2C_{3,j-1} + \frac{\beta^2}{1-\beta\gamma\epsilon} + 8\delta(\frac{\|p_{k+j}\|}{\|q_{k+j}\|} + \alpha C_{6,j}\frac{\|e_{k+j}\|}{\|q_{k+j}\|}))\|x_k - x^*\|^2 \\
&\leq C_{3,j}\|x_k - x^*\|^2.
\end{aligned}$$

We conclude the proof, using (7.84), (7.83), (7.94), and (7.93) with

$$\begin{aligned}
\|x_{k+j+1} - \hat{x}_{j+1}\| &= \|x_{k+j} - H_{k+j+1}F(x_{k+j}) - \hat{x}_j + \hat{H}_{j+1}\hat{F}(\hat{x}_j)\| \\
&\leq \|x_{k+j} - \hat{x}_j\| + \|H_{k+j+1} - \hat{H}_{j+1}\|\|F(x_{k+j})\| + \|H_{k+j}\|\|F(x_{k+j}) - \hat{F}(\hat{x}_j)\| \\
&\leq C_{4,j+1}\|x_k - x^*\|^2.
\end{aligned}$$

□

The theorems in this section indicate that the nonlinear EN-like methods converge twice as fast as the corresponding nonlinear Broyden methods.

7.4 Summary

In this chapter, we have developed nonlinear versions of the EN-like methods. Using their relationships to Broyden methods, we have shown that they converge superlinearly and n-step q-quadratically. Compared to Broyden's method, they require almost twice as much work per iteration step, but as the theorems above show, they also converge twice as fast. Since this situation is similar to the linear case, in which we achieved promising results with restarted and truncated methods, we plan to further investigate these methods, particularly their limited storage versions.

Another promising topic for future research is to examine the use of nonlinear EN-like or Broyden methods that approximate the Jacobian as outer methods and linear EN-like or Broyden methods to solve the occurring inner linear systems and the effects that these combinations have on the convergence.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this dissertation, we introduced a new family of methods for sparse linear systems, the EN-like methods.

We showed relationships of this family to the following methods:

- Broyden methods,
- GCR, ORTHOMIN, GMRES,
- a combination of a Galerkin step followed by a step of Richardson's method, and
- Newton's method for computing A^{-1}.

In order to increase the efficiency of the EN-like methods, we developed versions that do not require the explicit computation of $H_k v$, where H_k is the approximation to A^{-1} and v a vector, but replace them with the multiplication of the sparse matrix H_0 and several vector updates. For a few methods, EN and SEN, it was possible to achieve further savings by avoiding the evaluation of the updates for x_{k+1} in each iteration step and instead accumulate the appropriate coefficients. Since even the efficient versions are computationally expensive, with regard to operation count as well as storage, we also considered restarted and truncated versions. Our investigation of the computational complexity of the EN-like methods showed that their operation count per iteration step doubles compared to the corresponding Broyden methods or GMRES. The memory requirement is comparable to that of the Broyden methods and GCR or ORTHOMIN, but twice as large as that of GMRES. In order to offset the complexity, the new methods need to converge at least twice as fast as the Broyden methods or GMRES.

We were able to derive various convergence results. These comprised some new convergence results for Broyden methods and results for the EN-like methods, using their relationships to the Broyden methods and GCR. These results are as follows. All EN-like methods terminate within at most n steps. The full versions of some members of the Broyden family (PBM, GCR-BM, dual PBM, dual GCR-BM, and GBM with some new choices for α_k) and the EN-like family (EN, BEN, PEN, GEN, and the dual EN) converge superlinearly. If one uses a good initial guess for the approximation of the inverse, H_0, their restarted versions converge linearly under the same conditions. Estimates for residuals or errors for both full and restarted methods show sufficient improvement to offset the increased work. Using the relationship between GCR or ORTHOMIN and SEN, we showed convergence for the full, restarted and truncated SEN-method, if the matrix AH_0 is positive definite. We derived no further theoretical convergence results for truncated versions, but our numerical experiments give empirical evidence that often the truncated versions of some EN-like methods converge well. It also appears that they are less prone to stagnation or divergence than ORTHOMIN or truncated Broyden methods. Nevertheless, we also encountered problems with the truncated versions.

We dealt with these by developing an adaptive scheme that automatically increases the dimension of the underlying subspace. This version also restarts the method when it encounters certain potentially avoidable problems.

Overall, our numerical experiments show that several members (particularly EN, SEN, BEN) of the family of EN-like methods are competitive and in many cases better than other existing methods. Even though methods like CGS and BiCGSTAB may converge faster for many problems, EN-like methods are in general more robust, since, like GMRES, they have the option of increasing the size of the Krylov subspace. Additionally, they are often more efficient with regard to memory requirement than GMRES or ORTHOMIN.

We also considered the methods in the context of inner/outer iteration schemes such as GMRESR. Here, it is interesting to consider them as candidates for the outer as well as the inner methods. We examined several combinations of inner/outer iteration schemes involving EN, which proved to be competitive to GMRESR with regard to time and often superior when memory was limited. Since the EN and the SEN method generate orthogonal vectors $c_0, ..., c_{k-1}$, which are orthogonal to the residual r_k, it is possible to generate inner/outer iteration schemes, with EN or SEN as the outer method, which keep the residual of the inner method orthogonal to the outer space, as has been done for the orthogonality preserving version of GMRESR, GCRO. The new orthogonality preserving methods show a significant decrease in sparse matrix vector multiplications, combined with an increase in dense matrix vector multiplications, which leads to mixed numerical results. The new methods perform well in comparison to GCRO.

We embedded the EN-like methods into the robust software package PARASPAR. They are promising candidates in this context for several reasons: They are relatively robust. They are competitive or superior to GMRES or ORTHOMIN and in general use less memory. They evaluate as a byproduct, without additional cost, parameters that are indicators for the quality of the preconditioner. Some EN-like methods also show drastic divergence when applied to an ill-conditioned problem. Consequently, it is possible to recognize quickly when to reevaluate the preconditioner, before excessive time has been spent in the iterative method with an ineffective preconditioner.

Finally, using the relationship to Broyden's method, we developed nonlinear EN-like methods. We showed q-superlinear convergence and n-step q-quadratic convergence. As in the linear case, the methods converge twice as fast as Broyden methods, which offsets the fact that their computational complexity is about twice as large.

There are a variety of additional interesting topics for future work based on the results in the thesis:

- One can exploit the relationship of the EN-like method to Richardson's method and Newton's method. Accelerating these methods could lead to a more efficient EN-like method.

- Due to the relationship to Newton's method for computing A^{-1}, an investigation into the interaction of EN-like methods and the idea of approximate inverse preconditioners might yield an effective iterative method/preconditioner combination with less of the difficulties associated with maintaining sparsity in the approximate inverses.

- Since truncation has shown to be promising for many problems, but also has its difficulties for other problems, we would like to consider more sophisticated truncation schemes [43], [50]. Since for some problems restarted methods perform far better than truncated methods, the development of hybrids of restarted and truncated methods appears to be a promising approach.

- Orthogonality preserving methods are trading sparse for dense matrix vector operations. It would be of interest to investigate them in the context of suitable applications.

- Recall that the EN-like method is to Broyden's method what the Gauss-Seidel method is to the Jacobi method; i.e., an accelerated version based on using updated information. Also recall that we could interpret EN-like methods as trading computation for space. The development of a general acceleration scheme for other iterative method based on these two aspects of EN-like methods in combination with the more sophisticated truncation and adaptive strategies referred to above might yield an effective approach for large problems that have difficulty converging with small Krylov spaces; or it might lead to a new way of looking at existing methods.

- Since restarted and truncated versions have shown to be competitive or superior to the corresponding Broyden methods and GMRES in the linear case, the development of limited storage nonlinear versions appears to be a promising approach. Such versions have been considered before in the context of quasi-Newton methods [26], [31]. Additionally, it will be interesting to investigate these new nonlinear methods in an application specific context, e.g., the integration of differential-algebraic systems for power systems analysis involves nonlinear/linear systems, which are very difficult to solve.

REFERENCES

[1] I. Bogle, J. Perkins, *A new sparsity preserving Quasi-Newton update for solving nonlinear equations*, SIAM J. Sci. Stat. Comput. 11(1990) 621-630.

[2] C. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19(1965) 577-593.

[3] C. Broyden, *Quasi-Newton methods and their application to function minimization*, Math. Comp., 21(1967) 368-381.

[4] C. Broyden, *A new method of solving nonlinear simultaneous equations*, Computer J., 12(1969) 94-99.

[5] C. Broyden, *The convergence of single-rank quasi-Newton methods*, Math. Comp., 24(1970) 365-382.

[6] C. Broyden, J. Dennis, J. More, *On the local and superlinear convergence of Quasi-Newton methods*, J. Inst. Maths Applics 12(1973) 223-245.

[7] T. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, C. Tong *QMRCGSTAB: A quasi-minimal residual variant of the BI-CGSTAB algorithm for nonsymmetric systems*, SIAM J. Sci. Stat. Comp. 15 (1994) 338-347.

[8] P.Deuflhard, R. Freund, A. Walter, *Fast secant methods for the iterative solution of large nonsymmetric linear systems*, IMPACT of Computing in Science and Engineering, 2(1990) 244-276.

[9] J. Demmel, M. Heath, H. van der Vorst, *Parallel numerical linear algebra*, Acta Numerica (1993), 111-197.

[10] J. Dennis, Jr., J. More, *Quasi-Newton methods, motivation and theory*, SIAM Review 1(1977), 46-89.

[11] J. Dennis, Jr., R. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[12] S. Eisenstat, H. Elman, M. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal. (1983) 345-357.

[13] T. Eirola, O. Nevanlinna, *Accelerating with rank-one updates*, Lin. Alg. and its Appl., 121(1989), 511-520.

[14] H. Elman, *Iterative methods for large, sparse, nonsymmetric systems of linear equations*, Research Report 229, Yale University, 1982.

[15] M. Engleman, G. Strang, K.-J. Bathe, *The application of quasi-Newton methods in fluid mechanics*, International Journal for Numerical Methods in Engineering, 17(1981) 707-718.

[16] V. Faber, T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21(1984) 352-362.

[17] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis Dundee 1975, G. Watson, ed., Springer-Verlag, New York, 1976.

[18] R. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. on Scientific Computing 14 (1993) 470ff.

[19] R. Freund, N. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numerische Mathematik 60(1991) 315-339.

[20] R. Freund, T. Szeto, *A quasi-minimal residual squared algorithm for non-Hermitian linear systems* Technical Report, RIACS, NASA Ames Res. Center, (1991).

[21] R. Freund, G. Golub, N. Nachtigal, *Iterative solution of linear systems*, Acta Numerica, (1992) 57-100.

[22] K. Gallivan, A. Sameh, Z. Zlatev, *A parallel hybrid sparse linear system solver*, Computing Systems in Engineering, 1(1990) 183-195.

[23] D. Gay, R. Schnabel, *Solving systems of nonlinear equations by Broyden's method with projected updates*, in O. L. Mangasarian, R. Meyer, S. Robinson (eds.), Nonlinear Programming 3, pp. 245-281, Academic Press, New York, 1978.

[24] D. Gay, *Some convergence properties of Broyden's method*, SIAM J. Numer. Anal. 16(1979), 623-630.

[25] R. Horn, C. Johnson, *Topics in matrix analysis*, Cambridge University Press, 1991.

[26] D. Liu, J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming 45(1989), 503-528.

[27] D. Luenberger, *Linear and nonlinear programming*, Addison-Wesley, Reading, MA, 1984.

[28] U. Meier Yang, *Preconditioned conjugate gradient-like methods for nonsymmetric linear systems*, CSRD Tech. Report 1210, Center for Research and Development, University of Illinois at Urbana-Champaign, 1992.

[29] J. Meijerink, H. van der Vorst, *An iterative solution method for linear systems for which the coefficient matrix is a symmetric M-matrix*, Math. Comp. 31(1977) 148-162.

[30] J. More, J. Traugenstein, *On the global convergence of Broyden's method*, Mathematics of Computation 30(1976) 523-540.

[31] J. Nocedal, *Updating Quasi-Newton matrices with limited storage*, Mathematics of Computation 35(1980) 773-782.

[32] D. O'Leary, *Why Broyden's nonsymmetric method terminates on linear equations*, Tech. Report CS-TR-3045, University of Maryland, 1993, to appear in SIAM J. on Optimization.

[33] J. Ortega, W. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York, 1970.

[34] J. Paloschi, J. Perkins, *Scale invariant Quasi-Newton methods for the solution of nonlinear equations*, Comput. Chem. Engineering 12(1988) 91-97.

[35] V. Pan, R. Schreiber, *An Improved Newton Iteration for the Generalized Inverse of a Matrix, with Applications*, Tech. Report.

[36] H. Rutishauser, *Theory of gradient methods*, Mitteilungen aus dem Institut fuer Angewandte Mathematik 8(1959) 24-29.

[37] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, Tech. Report, Computer Science Department and MSI, University of Minnesota, 1991.

[38] Y. Saad, M. Schultz, *Conjugate gradient-like algorithms for solving nonsymmetric linear systems*, Math. Comp, 44(1985) 417-424.

[39] Y. Saad, M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7(1986) 856-869.

[40] G. Schultz, *Iterative Berechnung der reziproken Matrix*, Z. Angew. Math. Mech. 13(1933) 57-59.

[41] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 10(1989) 36-52.

[42] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[43] E. de Sturler, D. Fokkema, *Nested Krylov methods and preserving the orthogonality*, in T. Manteuffel and S. McCormick, eds., Proceedings of the Sixth Copper Mountain Multigrid Conference on Multigrid Methods, VA, 1993.

[44] E. de Sturler, *Nested Krylov methods based on GCR*, Technical Report 93-50, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, Holland, 1993, to appear in Journal of Computational and Applied Mathematics.

[45] C. Tong, *A comparative study of preconditioned Lanczos methods for nonsymmetric linear systems*, Tech. Report, Center for Computational Engineering, Sandia National Laboratories, Livermore, CA, 1992.

[46] P. Vinsome, *Orthomin, an iterative method for solving sparse sets of simultaneous linear equations*, in Proc. Fourth Symp. on Reservoir Simulation, Society of Petr. Eng. of AIME, 1976.

[47] H. van der Vorst, *BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp. 13 (1992) 631-644.

[48] H. van der Vorst, C. Vuik, *GMRESR: A family of nested GMRES methods*, to appear in Numerical Linear Algebra and its Applications.

[49] V. Voevodin, *The problem of a non-selfadjoint generalization of the conjugate gradient method has been closed*, USSR Comput. Math. and Math. Phys. 23 (1983) 143-144.

[50] C. Vuik, *Further experiences with GMRESR*, Technical Report 92-12, Delft University of Technology, 1992.

[51] C. Vuik, H. van der Vorst, *A comparison of some GMRES-like methods*, Linear Algebra and its Applications, 160 (1992) 131-162.

[52] Z. Zlatev, *Computational methods for general sparse matrices*, Kluwer Academic Publishers, Dordrecht, Holland, 1991.

VITA

Ulrike Meier Yang was born on February 5, 1959 in Darmstadt, Germany. She graduated with a Diplom in Mathematics from Ruhruniversität Bochum in Bochum, Germany, in 1983. She worked as a researcher at the Center for Applied Mathematics (ZAM) of the KFA Jülich GmbH, a research center in Jülich Germany, from 1983 to 1985. She has been employed at the Center of Supercomputing Research and Development of the University of Illinois at Urbana-Champaign as a Computer Scientist from 1985 to 1994 and as a Senior Computer Scientist since 1994.