Research Institute for Advanced Computer Science
NASA Ames Research Center

IN-61
43067
p.41

Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems

Peter Deuflhard, Roland Freund, and Artur Walter

RIACS Technical Report 90.31

July 1990

To appear in IMPACT of Computing in Science and Engineering

Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems

Peter Deuflhard, Roland Freund, and Artur Walter

The Research Institute of Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 311, Columbia, MD 244, (301)730-2656

Work reported herein was supported in part by DARPA via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA). Part of the work was performed at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035.

The authors would like to thank Mrs. Erlinda Cadano–Körnig for her excellent TEX–typing of the manuscript.

Peter Deuflhard Roland Freund Artur Walter

Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems

Abstract

A family of secant methods based on general rank–1 updates has been revisited in view of the construction of iterative solvers for large non–Hermitian linear systems. As it turns out, both Broyden's "good" and "bad" update techniques play a special role — but should be associated with two different line search principles. For Broyden's "bad" update technique, a minimum residual principle is natural — thus making it theoretically comparable with a series of well–known algorithms like GMRES. Broyden's "good" update technique, however, is shown to be naturally linked with a minimum "next correction" principle — which asymptotically mimics a minimum error principle. The two minimization principles differ significantly for sufficiently large system dimension. Numerical experiments on discretized PDE's of convection diffusion type in 2–D with internal layers give a first impression of the possible power of the derived "good" Broyden variant.

Key Words: nonsymmetric linear system, secant method, rank–1 update, Broyden's method, line search, GMRES.

AMS(MOS) Subject Classifications: 65F10, 65N20

Contents

1. Introduction

The solution of large sparse systems of linear equations

$$Ax = b \tag{1.1}$$

is one of the most frequently encountered tasks in numerical computations. In particular, such systems arise from finite difference or finite element approximations to partial differential equations (PDEs). For Hermitian positive definite coefficient matrices A, the classical conjugate gradient method (CG) of HESTENES/STIEFEL [11] is one of the most powerful iterative techniques for solving (1.1).

In recent years, a number of CG type methods for solving general non-Hermitian linear systems (1.1) have been proposed. The most widely used of these algorithms is GMRES due to SAAD/SCHULTZ [13]. However, solving non–Hermitian linear systems is, in general, by far more difficult than the case of Hermitian A, and the situation is still not very satisfactory. For instance, this is reflected in the fact that for methods such as GMRES work and storage per iteration grow linearly with the iteration number k. Consequently, in practice, one can not afford to run the full algorithm and restarted or truncated versions are used instead. Notice that, on the contrary, CG for Hermitian A is based on a three-term recursion and thus work and storage per iteration remain constant.

Non–Hermitian linear systems (1.1) are special cases of systems of *nonlinear* equations. For sufficiently good initial guesses, secant methods (see e.g. DENNIS/SCHNABEL [3]) based on Broyden's rank–1 updates are known to be quite efficient techniques for solving these more general problems. However, up to now, secant methods for solving linear systems have had a bad reputation.

The purpose of this paper is to take an unusual look at secant methods for non-Hermitian linear systems (1.1). In particular, as will be shown, combining Broyden's good and bad updates with different line search principles leads to iterative schemes which are competitive with GMRES. More than that, these secant methods typically exhibit a better reduction of the Euclidean error than GMRES. This is of particular importance for solving linear systems which arise in the context of multilevel discretizations of PDEs. There, linear systems are only solved to an accuracy corresponding to the discretization error on the respective level. In order to obtain such approximate solutions with as few iterations as possible, reduction of the Euclidean error is typically more crucial than minimizing the residual norm as GMRES does. For a description of such multilevel techniques, see the recent paper of DEUFLHARD/LEINEN/YSERENTANT [5].

It is well known (see e.g. FLETCHER [7, Chapter 3]) that CG for Hermitian positive definite A is intimately connected with minimization algorithms based on Broyden's family of rank–2 updates. In view of this result, the similar behavior of GMRES and secant methods based on rank–1 updates might not come as a surprise. Nevertheless, there appears to be no strict connection between the two techniques. Recently, however, EIROLA/NEVANLINNA [6] have established a connection between GMRES and a certain rank–1 update based on a nonstandard secant condition (cf. Remark 1 in Section 2.1).

The paper is organized as follows. In Section 2.1, we introduce a general family of secant methods. In Sections 2.2 and 2.3, special rank–1 updates and line search principles, respectively, are discussed. In Sections 3.1 and 3.2, we present convergence results for secant methods based on Broyden's bad and good updates. These results are then illustrated for a linear system arising from a simple 1–D boundary value problem in Section 3.3. Next, we discuss actual implementations of the proposed secant methods in Section 4. Typical numerical experiments are reported in Section 5. Finally, we make some concluding remarks.

Throughout this paper, all vectors and matrices are assumed to be complex. As usual, $M^* = (\overline{m_{kj}})$ denotes the conjugate transpose of the matrix $M = (m_{jk})$. The vector norm $\|x\| = \sqrt{x^*x}$ is always the Euclidean norm and $\|M\| = \sup_{\|x\|=1} \|Mx\|$ the corresponding matrix norm. Occasionally, the Frobenius norm $\|M\|_F = (\sum_{j,k} |m_{jk}|^2)^{1/2}$ will be used.

2. A Family of Secant Methods

The paper deals with the solution of linear systems (1.1) where A is a non-Hermitian $n \times n$ matrix and $b \in \mathbf{C}^n$. From now on, it is always assumed that A is nonsingular, and $x := A^{-1}b$ denotes the exact solution of (1.1).

The methods studied in this paper are iterative schemes. For any given starting vector $x_0 \in \mathbf{C}^n$, a sequence of approximations x_k, $k = 1, 2, \ldots$, to x is computed. Furthermore, in each step an $n \times n$ matrix H_k which approximates A^{-1} is generated. Here H_0 is a given nonsingular initial approximation of A^{-1}.

In the sequel,

$$e_k := x - x_k \quad \text{and} \quad r_k := b - Ax_k$$

always denote the *error vector* and *residual vector*, respectively, corresponding to the iterate x_k. Moreover,

$$E_k := I - H_k A$$

is the *error matrix* associated with the "preconditioning" matrix H_k and

$$\Delta_k := H_k r_k$$

is the "preconditioned" residual vector. Finally, for nonsingular H_k, we denote by

$$B_k := H_k^{-1}$$

the approximations of A.

2.1 The General Algorithm

The approximation H_{k+1} of A is obtained from the one of the previous iteration, H_k, by adding a rank–1 correction. In conjunction with the requirement that the following *secant condition* (or *quasi-Newton condition*)

$$H_{k+1} A \Delta_k = \Delta_k \tag{2.1}$$

holds, this leads (see e.g. [3, Chapter 8]) to the *general update*

$$H_{k+1} = H_k + (I - H_k A) \frac{\Delta_k v_k^*}{v_k^* H_k A \Delta_k} H_k \tag{2.2}$$

due to BROYDEN [1]. Here, $v_k \in \mathbf{C}^n$ is any vector such that $v_k^* H_k A \Delta_k \neq 0$. By applying the Sherman-Morrison formula to (2.2), one readily verifies that H_{k+1} is nonsingular with inverse

$$B_{k+1} = B_k + (A - B_k) \frac{\Delta_k v_k^*}{v_k^* \Delta_k}\ , \tag{2.3}$$

as long as H_k is nonsingular and $v_k^* \Delta_k \neq 0$.

Remark 1. EIROLA/NEVANLINNA [6] study secant methods which are based on the "conjugate transposed" secant condition

$$H_{k+1}^* A^* c_k = c_k \tag{2.4}$$

instead of (2.1). For the special choice $c_k = A\Delta_k$ in (2.4), the resulting algorithm ([6], see also [14]) is mathematically equivalent to GMRES.

In each iteration, the new approximation x_{k+1} to x is obtained by correcting the previous iterate x_k along the preconditioned residual Δ_k. In combination with the update (2.2), this leads to the following informal algorithm.

Algorithm 2.1

$$\begin{aligned}
&\text{Start:} \quad \text{a)} \quad r_0 := b - Ax_0 \\
&\text{Iteration loop:} \quad k = 0, 1, \ldots : \\
&\qquad \text{b)} \quad \Delta_k := H_k r_k \\
&\qquad\qquad q_k := A\Delta_k \\
&\qquad\qquad z_k := H_k q_k \\
&\qquad \text{c)} \quad x_{k+1} := x_k + t_k \Delta_k \\
&\qquad\qquad r_{k+1} := r_k - t_k q_k \\
&\quad \text{Update:} \\
&\qquad \text{d)} \quad H_{k+1} := H_k + (\Delta_k - z_k)\frac{v_k^* H_k}{v_k^* z_k}
\end{aligned}$$

Notice that Algorithm 2.1 describes a whole family of secant methods which still depend on the choices of v_k in the update d) and the step length t_k in c). Strategies for the selection of these parameters will be discussed in Sections 2.2 and 2.3.

In the following lemma, we collect some simple recursions that are valid for all choices of v_k and t_k. Here and in the sequel, the notations

$$\tau_k := \frac{v_k^* \Delta_k}{v_k^* z_k}\,, \quad \tilde{z}_i = \tilde{z}_i^{(k)} := H_i A \Delta_k\,, \quad \text{and} \quad \gamma_i := \frac{v_i^* \tilde{z}_i}{v_i^* z_i} \tag{2.5}$$

are used.

Lemma 2.2 *Let $v_i^* z_i \neq 0$, $i = 0, \ldots, k-1$. Then:*

$$\begin{aligned} &a)\quad e_{k+1} = ((1-t_k)I + t_k E_k)\, e_k\,, \\ &b)\quad \Delta_{k+1} = (1 - t_k + \tau_k)\Delta_k - \tau_k z_k) = ((1-t_k)I + \tau_k E_k)\,\Delta_k\,, \\ &c)\quad \tilde{z}_{i+1} = \tilde{z}_i + \frac{\gamma_i}{\tau_i}\left(\Delta_{i+1} - (1-t_i)\Delta_i\right),\ i = 0, \ldots, k-1\,. \end{aligned} \tag{2.6}$$

Proof. Note that $e_{k+1} = e_k - t_k \Delta_k$ and $\Delta_k = (I - E_k)e_k$. Combining these two identities yields (2.6a).
Next, one easily verifies that

$$\Delta_{k+1} = H_{k+1} r_{k+1} = (1-t_k)\Delta_k + \tau_k(\Delta_k - z_k)\,. \tag{2.7}$$

Since $E_k \Delta_k = \Delta_k - z_k$, (2.7) immediately leads to (2.6b).
By using (2.5) and the update formula which connects H_{i+1} and H_i, one obtains

$$\tilde{z}_{i+1} = \tilde{z}_i + \gamma_i(\Delta_i - z_i)\,. \tag{2.8}$$

Finally, by rewriting the term $\Delta_i - z_i$ in (2.8) by means of the first identity (with $k = i$) in (2.6b), one arrives at (2.6c). ∎

2.2 Special Rank–1 Updates

First, note that, by (2.2), the error matrix associated with the preconditioner H_k satisfies the update formula

$$E_{k+1} = E_k\left(I - \frac{\Delta_k v_k^*}{v_k^* z_k} H_k A\right)\,. \tag{2.9}$$

Clearly, one would like to improve the preconditioner from step to step. Thus, v_k in (2.9) should be chosen such that a suitable norm of E_k is decreasing. In this section, three special choices of v_k are discussed.

(A) The first one is the so-called *Broyden's "good" update* [1]. Here, in each iteration, one sets

$$v_k := \Delta_k\,. \tag{2.10}$$

Assume that H_k is nonsingular and that $x_k \neq x$, which implies $\Delta_k \neq 0$. With (2.10), (2.9) can be rewritten as

$$E_{k+1} = E_k P_k \quad \text{where} \quad P_k := \left(I - \frac{\Delta_k \Delta_k^*}{\Delta_k^* H_k A \Delta_k} H_k A\right)\,. \tag{2.11}$$

Remark that, except for the trivial case $H_kA = I$, P_k in (2.11) is an oblique, non-orthogonal projection. Thus, one cannot guarantee that $\|E_k\|$ is decreasing. However, for the different error matrix

$$\tilde{E}_k := I - A^{-1}B_k \,,$$

one obtains such a reduction property:

$$\tilde{E}_{k+1} = \tilde{E}_k Q_k \quad \text{where} \quad Q_k := \left(I - \frac{\Delta_k \Delta_k^*}{\Delta_k^* \Delta_k} \right) . \tag{2.12}$$

Now, Q_k is an orthogonal projection. Consequently, (2.12) guarantees an improvement of the preconditioner in each step, in the sense that

$$\|\tilde{E}_{k+1}\| \le \|\tilde{E}_k\| \tag{2.13}$$

and

$$\|\tilde{E}_{k+1}\|_F^2 = \|\tilde{E}_k\|_F^2 - \frac{\|\tilde{E}_k \Delta_k\|^2}{\|\Delta_k\|^2} \,. \tag{2.14}$$

Obviously, in view of (2.2) and (2.10), Broyden's good update is only defined as long as

$$\Delta_k^* H_k A \Delta_k \ne 0 \tag{2.15}$$

which (cf. (2.3)) guarantees that with H_k also H_{k+1} is nonsingular.

In particular, the more restrictive condition

$$\Delta_k^* H_k A \Delta_k > 0 \tag{2.16}$$

certainly implies (2.15). Clearly, (2.16) can be rewritten as

$$\varepsilon_k := \frac{\Delta_k^* E_k \Delta_k}{\Delta_k^* \Delta_k} < 1 \,. \tag{2.17}$$

Since $\varepsilon_k \le \|E_k\|$, a sufficient condition for (2.17) is

$$\|E_k\| < 1 \,. \tag{2.18}$$

Now, it is easily verified that $\tilde{E}_k$ and E_k are connected by

$$E_k = -(I - \tilde{E}_k)^{-1} \tilde{E}_k \,,$$

and it follows that

$$\|E_k\| \le \frac{\|\tilde{E}_k\|}{1 - \|\tilde{E}_k\|} \,. \tag{2.19}$$

By (2.19), the condition

$$\|\tilde{E}_k\| < \tfrac{1}{2} \tag{2.20}$$

implies (2.18). If (2.20) is satisfied for $k = 0$, then (2.13) guarantees that (2.20) holds for *all* k. Finally, by (2.18), $H_k A$ and, since A is assumed to be nonsingular, H_k is nonsingular.

Therefore, we have proved the following

Lemma 2.3 *Let H_0 be a nonsingular $n \times n$ matrix such that $\|\tilde{E}_0\| < \frac{1}{2}$. Then, Broyden's good update* (2.2), *with v_k chosen as in* (2.10), *is well defined as long as $x_k \neq x$.*

(B) The so-called *Broyden's "bad" update* [1] is obtained by choosing v_k in (2.2) such that

$$H_k^* v_k = A\Delta_k = q_k$$

holds. Then, (2.9) reduces to

$$E_{k+1} = E_k \left(I - \frac{q_k q_k^*}{q_k^* q_k} A \right) . \tag{2.21}$$

Remark that Broyden's bad update is well defined as long as $\Delta_k \neq 0$. In particular, no additional restrictions for H_0 are needed.

For the special error matrix

$$\hat{E}_k := A E_k A^{-1} = I - A H_k , \tag{2.22}$$

(2.21) leads to the update formula

$$\hat{E}_{k+1} = \hat{E}_k \left(I - \frac{q_k q_k^*}{q_k^* q_k} \right) . \tag{2.23}$$

From (2.23), it follows that H_{k+1} is an improved preconditioner, in the sense that

$$\|\hat{E}_{k+1}\| \leq \|\hat{E}_k\| \tag{2.24}$$

and

$$\|\hat{E}_{k+1}\|_F^2 = \|\hat{E}_k\|_F^2 - \frac{\|\hat{E}_k q_k\|^2}{\|q_k\|^2} . \tag{2.25}$$

(C) A third obvious choice for v_k in (2.2) is

$$v_k := z_k .$$

The corresponding update (2.9) for the error matrix is

$$E_{k+1} = E_k \left(I - \frac{\Delta_k z_k^*}{z_k^* z_k} H_k A \right) . \tag{2.26}$$

Here, one needs to ensure $z_k \neq 0$. Obviously, this is guaranteed if H_k is nonsingular and $x_k \neq x$. If H_k is nonsingular, then (2.26) can be rewritten in terms of an orthogonal projection as follows:

$$E_{k+1} = E_k (H_k A)^{-1} \left(I - \frac{z_k z_k^*}{z_k^* z_k} \right) H_k A . \tag{2.27}$$

However, unlike as for updates (A) and (B), (2.27) does not imply a reduction property of some "natural" measure for the preconditioner H_k. This suggests that this type of update is not competitive with Broyden's good and bad ones. Indeed, this was confirmed by our numerical experiments.

2.3 Line Search Principles

In this section, the selection of the step length t_k in part c) of Algorithm 2.1 is discussed. Ideally, one would like to choose t_k such that

$$\|e_{k+1}(t_k)\| = \min_{t \in \mathbf{C}} \|e_{k+1}(t)\| \tag{2.28}$$

where

$$e_{k+1}(t) := e_k - t\Delta_k .$$

Unfortunately, since x and hence e_k is unavailable, the step length defined by (2.28) can not be computed. However, in view of

$$e_{k+1}(t) = A^{-1} r_{k+1}(t) \quad \text{where} \quad r_{k+1}(t) := r_k - t q_k , \tag{2.29}$$

(2.28) can be satisfied at least approximately by choosing t_k such that

$$\|C_k r_{k+1}(t_k)\| = \min_{t \in \mathbf{C}} \|C_k r_{k+1}(t)\| . \tag{2.30}$$

Here C_k is some approximate inverse of A. At iteration k of Algorithm 2.1, there are three natural choices for C_k, namely H_{k+1}, H_k, or simply $C_k = I$, which lead to the line search principles (a), (c), or (b), respectively. Next, these three strategies are discussed.

(a) With $C_k = H_{k+1}$ and $\Delta_{k+1} = H_{k+1}r_{k+1}$, (2.30) reads as follows:

$$\|\Delta_{k+1}(t_k)\| = \min_{t\in\mathbf{C}} \|\Delta_{k+1}(t)\| . \tag{2.31}$$

Using (2.6b) and the second relation in (2.29), one readily verifies that (2.31) is equivalent to

$$\Delta_k^*\Delta_{k+1} = 0 \quad \text{where} \quad \Delta_{k+1} = (1+\tau_k)\Delta_k - \tau_k z_k - t_k\Delta_k . \tag{2.32}$$

Recall that τ_k was defined in (2.5) and note that τ_k still depends on the particular choice of the rank–1 update (2.2).

Finally, from (2.32), it follows that the step length for the line search principle (2.31) is given by

$$t_k = \tilde{t}_k := 1 + \tau_k - \tau_k \frac{\Delta_k^* z_k}{\Delta_k^*\Delta_k} . \tag{2.33}$$

Note that for the special case, $v_k = \Delta_k$, of *Broyden's good update*, (2.33) leads to

$$\tilde{t}_k = \tau_k = \frac{\Delta_k^*\Delta_k}{\Delta_k^* z_k} . \tag{2.34}$$

(b) For $C_k = I$, (2.30) reduces to

$$\|r_{k+1}(t_k)\| = \min_{t\in\mathbf{C}} \|r_{k+1}(t)\| \tag{2.35}$$

or, equivalently,

$$q_k^* r_{k+1} = 0 \quad \text{where} \quad r_{k+1} = r_k - t_k q_k . \tag{2.36}$$

Hence, by (2.36), the minimization principle (2.35) leads to

$$t_k = \hat{t}_k := \frac{q_k^* r_k}{q_k^* q_k} . \tag{2.37}$$

Remark that for *Broyden's bad update*, (B), one has

$$\hat{t}_k = \tau_k . \tag{2.38}$$

(c) With $C_k = H_k$, (2.30) specifies to

$$\|H_k r_{k+1}(t_k)\| = \min_{t\in\mathbf{C}} \|H_k r_{k+1}(t)\| . \tag{2.39}$$

By rewriting (2.39) in the form

$$z_k^* H_k r_{k+1} = 0 \quad \text{where} \quad H_k r_{k+1} = \Delta_k - t_k z_k \ ,$$

it follows that

$$t_k = t_k^\circ := \frac{z_k^* \Delta_k}{z_k^* z_k} \ . \tag{2.40}$$

Here, for update (C), one has

$$t_k^\circ = \tau_k \ . \tag{2.41}$$

Notice that, in view of (2.34), (2.38), and (2.41), the choice $t_k = \tau_k$ for the step length leads to a natural coupling of the three special rank–1 updates (A), (B), and (C) with the line search principles (a), (b), and (c), respectively.
More general, for $t_k = \tau_k$, the following properties hold.

Lemma 2.4 *In Algorithm* 2.1, *let $t_k = \tau_k$ be chosen and assume that $v_k^* z_k \neq 0$. Then:*

a) The iterate x_{k+1} is uniquely defined by the Galerkin type condition

$$H_k r_{k+1} \perp v_k \quad \textit{and} \quad x_{k+1} \in x_k + \operatorname{span}\{\Delta_k\} \ , \tag{2.42}$$

b) $H_{k+1} r_{k+1} = H_k r_{k+1}$.

Proof. By the second condition in (2.42), $x_{k+1} = x_k + t\Delta_k$ and thus

$$H_k r_{k+1} = \Delta_k - t z_k$$

for some $t \in \mathbf{C}$. Together with the definition of τ_k in (2.5), it follows that

$$v_k^* H_k r_{k+1} = v_k^* \Delta_k - t v_k^* z_k = 0 \Leftrightarrow t = \frac{v_k^* \Delta_k}{v_k^* z_k} = \tau_k \ ,$$

and this concludes the proof of a).
Next, one easily verifies that

$$H_k r_{k+1} = (1 - t_k)\Delta_k + t_k E_k \Delta_k \ . \tag{2.43}$$

By comparing (2.6b) and (2.43), one obtains the relation stated in b). ■

Remark that the classical step length used in combination with Broyden's update (2.2) is $t_k \equiv 1$. Somewhat surprisingly, this choice guarantees that the resulting method — at least in theory — terminates after at most $2n$ steps with the exact solution of (1.1), as was shown by GAY [9] (cf. also [10]). Obviously, this finite termination property is not of practical importance for large sparse linear systems. Here, we take another look at the choice $t_k \equiv 1$.

Lemma 2.5 *In Algorithm 2.1, assume that $v_k^* z_k \neq 0$. Let x_k and x_{k+2} be the iterates generated by two successive steps of Algorithm 2.1 with step length $t_k = t_{k+1} = 1$. Then:*

$$x_{k+2} = \hat{x}_{k+1} + H_k \hat{r}_{k+1} \, , \quad r_{k+2} = (I - AH_k)\hat{r}_{k+1} \, , \tag{2.44}$$

where

$$\hat{x}_{k+1} = x_k + \tau_k \Delta_k \, , \quad \hat{r}_{k+1} = (I - \tau_k AH_k) r_k \, , \tag{2.45}$$

and

$$\tau_k = \frac{v_k^* \Delta_k}{v_k^* z_k} = \frac{(H_k^* v_k)^* r_k}{(H_k^* v_k)^* q_k} \, .$$

Proof. Since $t_k = t_{k+1} = 1$, we have

$$x_{k+2} = x_k + \Delta_k + \Delta_{k+1} \, . \tag{2.46}$$

For $t_k = 1$, the first identity in (2.6b) reduces to $\Delta_{k+1} = \tau_k(I - H_k A)\Delta_k$ and, thus, (2.46) can be rewritten as

$$x_{k+2} = x_k + \tau_k \Delta_k + H_k(I - \tau_k AH_k) r_k \, . \tag{2.47}$$

Now, (2.44) and (2.45) readily follow from (2.47). ∎

Note that, in view of part a) of Lemma 2.4, the intermediate quantity $\hat{x}_{k+1}$, (2.45), is just the Galerkin iterate in the sense of (2.42).

Therefore, Lemma 2.5 shows that, by combining two successive steps, Algorithm 2.1 with $t_k \equiv 1$ can be interpreted as follows. At the beginning of step k, the approximate solution x_k and the preconditioner H_k are available. From these quantities, the iterate x_{k+2} of step $k+2$ is obtained by applying *one Galerkin step*, namely (2.45), *followed by one step of Richardson iteration*, namely (2.44), to the preconditioned linear system

$$H_k A x = H_k b \, .$$

In general, the "virtual" iterate $\hat{x}_{k+1}$ and the actual iterate x_{k+1} are different.

Note that (2.44) is a Richardson step *without* line search. In particular, if H_k — as is to be expected in the early stage of the iteration — is not yet a good approximation to A^{-1}, then (2.44) will lead to an increase rather than a decrease of $\|r_{k+2}\|$. In order to prevent such undesirable effects, it appears preferable to combine Broyden's update with the line search principles (a), (b), or (c), instead of using $t_k \equiv 1$.

3. Convergence Analysis

In principle, Algorithm 2.1 could be implemented with any of the 9 combinations (Aa), ..., (Cc) of rank-1 updates (A), (B), and (C) with line search strategies (a), (b), and (c). As already mentioned in Section 2.2, the update (C) is not competitive with (A) and (B), and, therefore, (C) is dropped here. Among the remaining 6 combinations, only the pairs (Aa), (Bb), and (Ac) will be considered.

As a first step, the following auxiliary result for the case of the line search principle (b) is established.

Lemma 3.1 *In the general Algorithm* 2.1, *let the step length* (2.37), $t_k = \widehat{t}_k$, *be chosen. Then,*

$$\frac{\|r_{k+1}\|}{\|r_k\|} \leq \frac{\|\widehat{E}_k r_k\|}{\|r_k\|} \leq \|\widehat{E}_k\| \,, \tag{3.1}$$

with $\widehat{E}_k = I - AH_k$ *defined as in* (2.22).

Proof. From part c) of Algorithm 2.1 and (2.37), one obtains

$$r_{k+1} = r_k - \widehat{t}_k q_k = r_k - \frac{q_k^* r_k}{q_k^* q_k} q_k = \left(I - \frac{q_k q_k^*}{q_k^* q_k}\right) r_k = \left(I - \frac{q_k q_k^*}{q_k^* q_k}\right)(r_k - q_k) \,.$$

Since

$$r_k - q_k = \widehat{E}_k r_k \,, \tag{3.2}$$

it follows that

$$\|r_{k+1}\| = \left\| \left(I - \frac{q_k q_k^*}{q_k^* q_k}\right) \widehat{E}_k r_k \right\| \leq \|\widehat{E}_k r_k\| \leq \|\widehat{E}_k\| \cdot \|r_k\| \,,$$

and thus (3.1) holds. ■

Recall from Section 2.2 that the error matrix $\widehat{E}_k$ is closely connected with Broyden's bad update (B), cf. (2.23)–(2.25). In the following section, Lemma 3.1 will be used to obtain a convergence result for update (B).

3.1 Broyden's Bad Update

Theorem 3.2 *(B–update)*
Consider Algorithm 2.1 *with update (B) and step length* $t_k = \widehat{t}_k = \tau_k$, (2.37), *or* $t_k = 1$. *Assume that*

$$\|\widehat{E}_0\| \leq \widehat{\delta}_0 < 1 \,.$$

Then, the iteration converges globally satisfying

$$\frac{\|Ae_{k+1}\|}{\|Ae_k\|} = \frac{\|r_{k+1}\|}{\|r_k\|} \le \frac{\|\widehat{E}_k r_k\|}{\|r_k\|} \le \widehat{\delta}_0 < 1 \tag{3.3}$$

and

$$\tau_k > 0 \ \ for \ \ r_k \neq 0 \, . \tag{3.4}$$

Moreover, if $r_k \neq 0$ for all $k = 0, 1, \ldots$, then,

$$\lim_{k\to\infty} t_k = 1 \, , \tag{3.5}$$

and the convergence is superlinear in the sense that

$$\lim_{k\to\infty} \frac{\|Ae_{k+1}\|}{\|Ae_k\|} = 0 \, . \tag{3.6}$$

Proof. First, global convergence is shown for $t_k = \tau_k$. By Lemma 3.1,

$$\frac{\|r_{k+1}\|}{\|r_k\|} \le \frac{\|\widehat{E}_k r_k\|}{\|r_k\|} \le \|\widehat{E}_k\| \, . \tag{3.7}$$

In view of (2.24), (3.7) implies

$$\frac{\|r_{k+1}\|}{\|r_k\|} \le \|\widehat{E}_k\| \le \|\widehat{E}_0\| \le \widehat{\delta}_0 < 1 \, . \tag{3.8}$$

By combining (3.7) and (3.8), the statement in (3.3) follows. By (2.37), (2.38), and (3.2), the step length $\widehat{t}_k = \tau_k$ satisfies

$$\tau_k = \frac{q_k^* r_k}{q_k^* q_k} = \frac{\|r_k\|^2}{\|q_k\|^2} - \frac{r_k^* \widehat{E}_k^* r_k}{\|q_k\|^2} \, . \tag{3.9}$$

Using (3.7), one deduces from (3.9) that

$$\tau_k \ge \frac{\|r_k\|^2}{\|q_k\|^2} - \frac{|r_k^* \widehat{E}_k r_k|}{\|q_k\|^2} \ge \left(1 - \|\widehat{E}_k\|\right) \frac{\|r_k\|^2}{\|q_k\|^2} > 0 \ \text{ for } \ r_k \neq 0 \, ,$$

and (3.4) holds true.

Next, based on the Frobenius norm property (2.25), superlinear convergence is shown. Following the proof technique of BROYDEN/DENNIS/MORÉ [2], one obtains

$$\lim_{k\to\infty} \frac{\|\widehat{E}_k q_k\|}{\|q_k\|} = 0 \, . \tag{3.10}$$

By (3.8), the assumption $\hat{\delta}_0 < 1$ guarantees that the matrix $(I - \hat{E}_k)$ is nonsingular. Thus the relation (3.2) can be rewritten as

$$r_k = (I - \hat{E}_k)^{-1} q_k \,. \tag{3.11}$$

Using (3.11) and (3.8), one readily verifies that

$$\frac{\|\hat{E}_k r_k\|}{\|r_k\|} = \frac{\|\hat{E}_k (I - \hat{E}_k)^{-1} q_k\|}{\|(I - \hat{E}_k)^{-1} q_k\|} \leq \frac{1 + \|\hat{E}_k\|}{1 - \|\hat{E}_k\|} \cdot \frac{\|\hat{E}_k q_k\|}{\|q_k\|} \leq \frac{1 + \hat{\delta}_0}{1 - \hat{\delta}_0} \cdot \frac{\|\hat{E}_k q_k\|}{\|q_k\|} \,.$$

Therefore, by means of (3.7) and (3.10), one concludes that

$$\lim_{k \to \infty} \frac{\|r_{k+1}\|}{\|r_k\|} \leq \frac{1 + \hat{\delta}_0}{1 - \hat{\delta}_0} \lim_{k \to \infty} \frac{\|\hat{E}_k q_k\|}{\|q_k\|} = 0 \,,$$

which immediately yields (3.6). Similarly, from

$$t_k = \frac{q_k^* r_k}{q_k^* q_k} = \frac{q_k^* (I - \hat{E}_k)^{-1} q_k}{q_k^* q_k} \,,$$

one deduces that

$$|t_k - 1| \leq \frac{1}{1 - \hat{\delta}_0} \cdot \frac{\|\hat{E}_k q_k\|}{\|q_k\|} \,.$$

Therefore, (3.10) implies

$$\lim_{k \to \infty} |t_k - 1| = 0$$

which confirms (3.5).

For the case $t_k = 1$, the relation (2.6a) reduces to

$$e_{k+1} = E_k e_k$$

which is equivalent to

$$r_{k+1} = \hat{E}_k r_k \,. \tag{3.12}$$

Now (3.12), also yields (3.7). The rest of the proof can just be copied. ■

3.2 Broyden's Good Update

Theorem 3.3 *(A–update)*
Consider Algorithm 2.1 *with update (A) and line search either (a) or (c). Assume that*

$$\|\tilde{E}_0\| \leq \tilde{\delta}_0 < \tfrac{1}{3} \,. \tag{3.13}$$

Then, the iteration converges globally satisfying

$$\frac{\|e_{k+1}\|}{\|e_k\|} \leq \frac{\dfrac{\|\widetilde{E}_k z_k\|}{\|z_k\|} + \dfrac{\|\widetilde{E}_k \Delta_k\|}{\|\Delta_k\|}}{1 - \dfrac{\|\widetilde{E}_k \Delta_k\|}{\|\Delta_k\|}} \leq \frac{2\widetilde{\delta}_0}{1-\widetilde{\delta}_0} < 1 \tag{3.14}$$

and

$$\tau_k > 0 \ \ \text{for} \ \ e_k \neq 0 \ . \tag{3.15}$$

Moreover, if $e_k \neq 0$ for all $k = 0, 1, \ldots$, then the convergence is superlinear with

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0 \tag{3.16}$$

and

$$\lim_{k\to\infty} t_k = 1 \ . \tag{3.17}$$

Proof. First, line search (a) with $t_k = \tau_k$ (cf. (2.34)) is considered. Rewriting (2.6a) in terms of $\widetilde{E}_k$ yields

$$e_{k+1} = (1-\tau_k)\Delta_k - \widetilde{E}_k \Delta_k \ . \tag{3.18}$$

By means of the relations

$$1 - \tau_k = 1 - \frac{\Delta_k^* \Delta_k}{\Delta_k^* z_k} = \frac{\Delta_k^*(z_k - \Delta_k)}{\Delta_k^* z_k}$$

and

$$\Delta_k = (I - \widetilde{E}_k) z_k \ , \tag{3.19}$$

one obtains from (3.18)

$$e_{k+1} = \frac{\Delta_k^* \widetilde{E}_k z_k}{\Delta_k^* z_k} \cdot \Delta_k - \widetilde{E}_k \Delta_k \ . \tag{3.20}$$

Moreover, using the formula (3.19) once more, one easily verifies that

$$\begin{aligned} \frac{|\Delta_k^* \widetilde{E}_k z_k|}{|\Delta_k^* z_k|} &= \frac{|(z_k - \widetilde{E}_k z_k)^* \widetilde{E}_k z_k|}{|(z_k - \widetilde{E}_k z_k)^* z_k|} \\ &= \frac{|z_k^* \widetilde{E}_k z_k|}{\|z_k\|^2} \cdot \frac{\left|1 - \dfrac{\|\widetilde{E}_k z_k\|^2}{z_k^* \widetilde{E}_k z_k}\right|}{\left|1 - \dfrac{z_k^* \widetilde{E}_k z_k}{\|z_k\|^2}\right|} \leq \widetilde{\varepsilon}_k \cdot \frac{1-\widetilde{\varepsilon}_k}{1-\widetilde{\varepsilon}_k} = \widetilde{\varepsilon}_k \ , \end{aligned}$$

where

$$\tilde{\varepsilon}_k := \frac{\|\tilde{E}_k z_k\|}{\|z_k\|} \le \|\tilde{E}_k\| \le \tilde{\delta}_0 \, .$$

With these inequalities, (3.20) leads to the estimates

$$\frac{\|e_{k+1}\|}{\|\Delta_k\|} \le \tilde{\varepsilon}_k + \frac{\|\tilde{E}_k \Delta_k\|}{\|\Delta_k\|} \le 2\tilde{\delta}_0 \, . \tag{3.21}$$

Finally, with

$$e_k = (I + \tilde{E}_k)\Delta_k \, ,$$

one obtains

$$\|e_k\| \ge \left(1 - \frac{\|\tilde{E}_k \Delta_k\|}{\|\Delta_k\|}\right) \|\Delta_k\| \ge (1 - \tilde{\delta}_0)\|\Delta_k\| \, . \tag{3.22}$$

By combining (3.21) and (3.22), one ends up with (3.14).
Similarly, one shows

$$\tau_k = 1 - \frac{\Delta_k^* \tilde{E}_k z_k}{\Delta_k^* z_k} \ge 1 - \tilde{\varepsilon}_k \, ,$$

which certainly confirms the assertion (3.15).
In order to prove *superlinear* convergence, first remark that the Frobenius norm result (2.14) implies

$$\lim_{k\to\infty} \frac{\|\tilde{E}_k \Delta_k\|}{\|\Delta_k\|} = 0 \, . \tag{3.23}$$

Along lines similar as in the proof of Theorem 3.2, one then verifies that

$$\lim_{k\to\infty} \tilde{\varepsilon}_k = 0 \, . \tag{3.24}$$

Now, by using (3.23), (3.24), and the estimates in (3.14) of this theorem, one obtains (3.16) and, with

$$|t_k - 1| \le \tilde{\varepsilon}_k \, ,$$

also (3.17). This completes the proof for line search (a).
Next, consider the line search principle (c) where, by (2.40),

$$t_k = t_k^\circ = \frac{z_k^* \Delta_k}{z_k^* z_k} \, .$$

As before, one starts with

$$e_{k+1} = (1 - t_k)\Delta_k - \tilde{E}_k \Delta_k \, ,$$

which now leads to

$$e_{k+1} = \frac{z_k^* \widetilde{E}_k^* z_k}{z_k^* z_k} \Delta_k - \widetilde{E}_k \Delta_k \; .$$

From this, one derives the estimate

$$\frac{\|e_{k+1}\|}{\|\Delta_k\|} \leq \widetilde{\varepsilon}_k + \frac{\|\widetilde{E}_k \Delta_k\|}{\|\Delta_k\|} \; ,$$

which is the same as (3.21). The rest of the proof can essentially be copied. ∎

For the choice $t_k \equiv 1$, Broyden's classical good method is obtained. The convergence behavior of this algorithm is studied in BROYDEN/DENNIS/MORÉ [2]. In this case, the assumption (3.13) can be relaxed to $\widetilde{\delta}_0 < \frac{1}{2}$. As already mentioned, the choice $t_k \equiv 1$ guarantees that Broyden's good method stops after at most $2n$ steps. A slight modification, the so-called projected Broyden's method, even terminates after at most n iterations. This algorithm is analyzed in GAY/SCHNABEL [8].

Conjecture. The authors were unable to get rid of the factor 2 in (3.14). If this factor drops, then only $\widetilde{\delta}_0 < \frac{1}{2}$ would be required — which seems to be more reasonable in view of (2.20).

3.3 An Illustrative Example

In this section, we discuss a simple illustrative example, namely a convection-diffusion problem in 1–D. Consider the ODE boundary value problem

$$\begin{aligned} &\text{a)} \quad -u'' + \beta u' = 0 \quad \text{on} \quad (0,1) \; , \\ &\text{b)} \quad u(0) = 1 \; , \quad u(1) = 0 \; . \end{aligned} \tag{3.25}$$

By using upwind discretization on a uniform grid with step size $h = 1/n$, (3.25) leads to a linear system $Ax = b$ with the diagonally dominant tridiagonal matrix

$$A := \begin{bmatrix} 2+\beta h & -1 & & \\ -(1+\beta h) & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -(1+\beta h) & 2+\beta h \end{bmatrix} . \tag{3.26}$$

Let $n = 50$ and set $b = (1, 0, \dots, 0)^T$. Moreover, choose x_0 as the prolongation obtained from the exact solution on the coarser grid $h = 1/25$. For H_0, we chose simple diagonal preconditioning as in (5.1). In this case, one is able to compute all quantities of interest directly and to compare the convergence theory of Sections 3.1 and 3.2 with the actual behavior of the algorithms — see Table 3.1.

	$\|\tilde{E}_0\|$	$\|\hat{E}_0\|$	$\|\hat{E}_0 r_0\|/\|r_0\|$	$\|\tilde{E}_0 z_0\|/\|z_0\|$	$\|\tilde{E}_0 \Delta_0\|/\|\Delta_0\|$
$\beta = 5$	415	0.99	0.53	0.43	2.72
$\beta = 100$	64	0.99	0.37	0.28	0.24

Table 3.1: Quantities used in convergence theory of Sections 3.1 and 3.2 for Example (3.25).

These results seem to justify the relaxation of the rather restrictive convergence criteria in Sections 3.1 and 3.2 — compare (2.16) and (2.17) in the light of (2.18), (2.20), and (3.13).

In Fig. 3.1 and Fig. 3.2, the convergence history of 3 codes (see Section 5 for a description of these codes) is compared — both in terms of the residual norms $\|r_k\|$ and the error norms $\|e_k\|$.

In this example, both GMRES and the "bad Broyden" code BB successively reduce the residual norm, whereas the "good Broyden" code GB reduces the error norm — a property that has been shown to hold at least asymptotically without a storage restriction: just compare the minimization property (2.31), $\|\Delta_{k+1}\| = \min$, for $t_k = \tau_k$ with the asymptotic property (see (4.2) below) $\|\Delta_{k+1}\| \doteq \|e_{k+1}\|$ for $\tau_k \doteq 1$. As an illustration, Fig. 3.3 gives a comparison of the true and estimated errors.

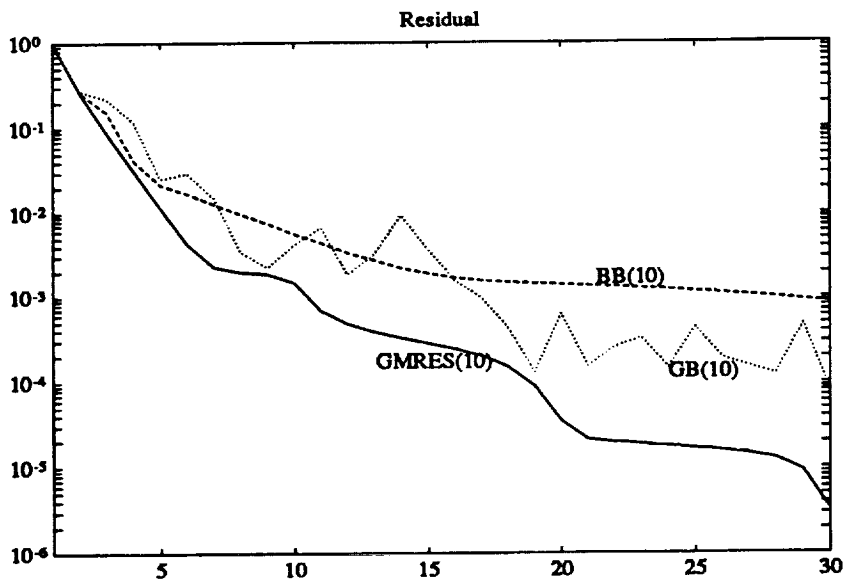

Figure 3.1: Comparative residual norms $\|r_k\|_2$ for 3 iterative solvers for Example (3.25).

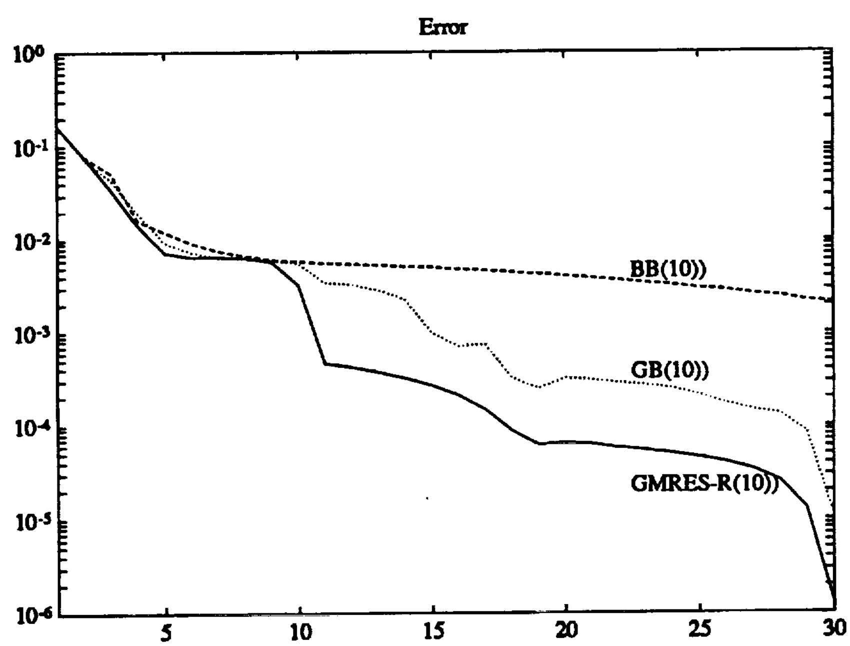

Figure 3.2: Comparative error norms $\|e_k\|_2$ for 3 iterative solvers with $k_{\max} = 10$ for Example (3.25).

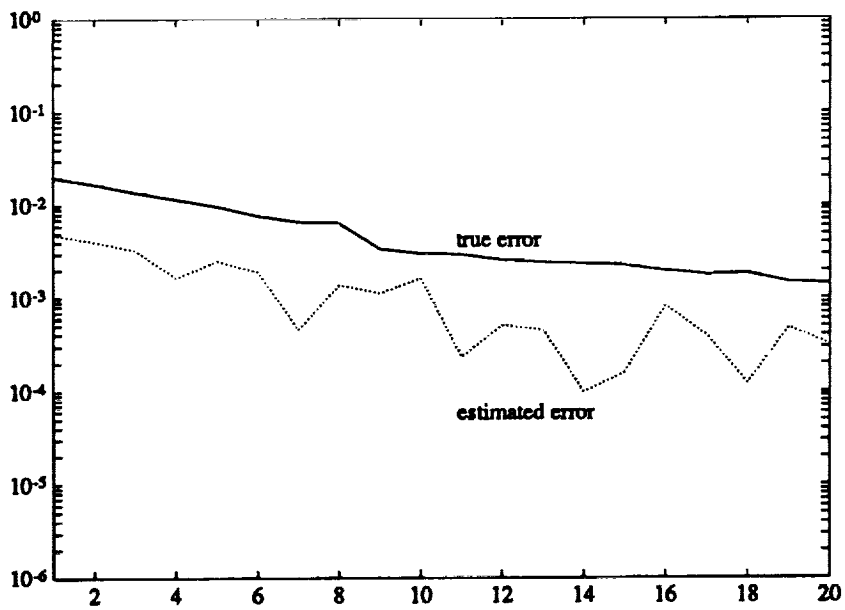

Figure 3.3: Iterative behavior of true errors $\|e_k\|$ and estimated errors $\|\Delta_k\|$ for GB(3) in Example (3.25).

4. Details of Realization

The secant methods based on Algorithm 2.1 with update either (A) or (B) are, of course, implemented in a storage saving compact form.

Algorithm (A): "Good Broyden"

$$
\begin{aligned}
&\text{Start:} \quad \text{a)} \quad r_0 := b - Ax_0 \\
&\qquad\qquad\quad\; \Delta_0 := H_0 r_0 \\
&\qquad\qquad\quad\; \sigma_0 := \Delta_0^* \Delta_0 \\
&\text{Iteration loop:} \quad k = 0, 1, \dots : \\
&\qquad \text{b)} \quad q_k := A\Delta_k \\
&\qquad\qquad\; \tilde{z}_0 := H_0 q_k \\
&\qquad \text{Update loop:} \quad i = 0, \dots, k-1 \ (\text{for } k \geq 1) \\
&\qquad \text{c)} \quad \tilde{z}_{i+1} := \tilde{z}_i + \frac{\Delta_i^* \tilde{z}_i}{\gamma_i \tau_i} (\Delta_{i+1} - (1 - t_i)\Delta_i) \\
&\qquad \text{d)} \quad z_k := \tilde{z}_k \\
&\qquad\qquad\; \gamma_k := \Delta_k^* z_k \\
&\qquad\qquad\; \tau_k := \sigma_k / \gamma_k \\
&\qquad\qquad\; t_k := \tau_k \ \text{ or } \ t_k := \hat{t}_k = q_k^* r_k / q_k^* q_k \ \text{ or } \ t_k := 1 \\
&\qquad\qquad\; x_{k+1} := x_k + t_k \Delta_k \\
&\qquad\qquad\; r_{k+1} := r_k - t_k q_k \\
&\qquad\qquad\; \Delta_{k+1} := (1 - t_k + \tau_k)\Delta_k - \tau_k z_k \\
&\qquad\qquad\; \sigma_{k+1} := \Delta_{k+1}^* \Delta_{k+1}
\end{aligned}
$$

Array storage. The above implementation requires to store (up to iteration step k) the vectors

$$\Delta_0, \dots, \Delta_k, q, z = \tilde{z} \ ,$$

for step length $t_k = \tau_k$ or $t_k = 1$ and, in addition, $r = r_k$ in the case of step length $t_k = \tilde{t}_k$, which sum up to

$$(k+2)n \quad \text{or} \quad (k+3)n \tag{4.1}$$

storage places.

Operation count. Per iteration step k one needs 1 matrix-vector multiplication, 1 solution of a preconditioned system of the form $B_0 z = q$ in order to obtain $\tilde{z}_0$ in b), and $(2k+7)n$ multiplications. Obviously, the inner loop vectorizes.

Termination criteria. Under the assumptions of Theorem 3.3 (cf. (3.22)), one has

$$\left(1 - \frac{\|\tilde{E}_k \Delta_k\|}{\|\Delta_k\|}\right) \|\Delta_k\| \leq \|e_k\| \leq \left(1 + \frac{\|\tilde{E}_k \Delta_k\|}{\|\Delta_k\|}\right) \|\Delta_k\| ,$$

which means that, at least asymptotically, $\|\Delta_k\|$ is a reasonable computationally available estimate for $\|e_k\|$ to be written as

$$\|\Delta_k\| \doteq \|e_k\| . \tag{4.2}$$

This motivates the *convergence criterion*

$$\frac{\sqrt{\sigma_{k+1}}}{\|x_{k+1}\|} \leq \varepsilon \tag{4.3}$$

where ε is some relative accuracy parameter to be specified by the user.
In order to ensure that $H_{k+1}A$ is nonsingular, recall condition (2.17), which reads $\varepsilon_k < 1$ in the notation of Section 2.2. By replacing this condition by the stricter one

$$\varepsilon_k \leq 1 - \frac{1}{\tau_{\max}} ,$$

we arrive at a *restart condition*

$$\tau_k \leq 0 \quad \text{or} \quad \tau_k > \tau_{\max}$$

which can be easily monitored. Here, τ_{max} is some internal parameter, and we have chosen $\tau_{\max} = 10$ in all the numerical experiments described in this paper.

Remark 1. The convergence criterion (4.3) nicely agrees with requirements needed in the global inexact Newton algorithm for *nonlinear* problems as given by DEUFLHARD [4].

Remark 2. Clearly, the Euclidean inner product in Algorithm (A) can be replaced by any other inner product $(\cdot,\cdot)$ — possibly scaled and certainly depending on the problem to be solved.

Algorithm (B): "Bad Broyden"

$$
\begin{aligned}
&\text{Start:} \quad \text{a)} \quad r_0 := b - Ax_0 \\
&\qquad\qquad\qquad \Delta_0 := H_0 r_0 \\
&\text{Iteration loop:} \quad k = 0, 1, \ldots : \\
&\qquad\qquad \text{b)} \quad q_k := A\Delta_k \\
&\qquad\qquad\qquad \tilde{z}_0 := H_0 q_k \\
&\qquad \text{Update loop:} \quad i = 0, \ldots, k-1 \ (\text{for } \ k \geq 1) \\
&\qquad\qquad \text{c)} \quad \tilde{z}_{i+1} := \tilde{z}_i + \frac{q_i^* q_k}{\beta_i t_i}(\Delta_{i+1} - (1 - t_i)\Delta_i) \\
&\qquad\qquad \text{d)} \quad z_k := \tilde{z}_k \\
&\qquad\qquad\qquad \beta_k := q_k^* q_k \\
&\qquad\qquad\qquad t_k := \frac{r_k^* q_k}{\beta_k} \\
&\qquad\qquad\qquad x_{k+1} := x_k + t_k \Delta_k \\
&\qquad\qquad\qquad r_{k+1} := r_k - t_k q_k \\
&\qquad\qquad\qquad \Delta_{k+1} := \Delta_k - t_k z_k
\end{aligned}
$$

The version for $t_k = 1$ was ignored for obvious reasons.

Array Storage. The implementation of this algorithm requires to store (up to iteration step k) the vectors

$$\Delta_0, \ldots, \Delta_k, q_0, \ldots, q_k, z = \tilde{z}, r \ ,$$

which sums up to

$$(2k + 2)n$$

storage places — to be compared with (4.1).

Operation count. Per iterative step k one needs 1 matrix-vector multiplication, again solution of 1 linear preconditioned system with B_0 as coefficient matrix, and $(2k + 8)n$ multiplications. Once more, the inner loop easily vectorizes.

Termination criteria. Since update (B) is closely connected with minimization principle (2.35), the *convergence criterion* for Algorithm (B) will be based on the residual norm. In view of the property (for $t_k = \tau_k$)

$$r_{k+1}^* r_{k+1} = r_k^* r_k - t_k^2 q_k^* q_k \;,$$

Algorithm (B) is stopped as soon as

$$\|r_{k+1}\| \leq \varepsilon \|r_0\|$$

is reached. Again, ε is to be specified by the user. Moreover, the iteration is *restarted*, if

$$|t_k| \cdot \|q_k\| < \varepsilon \|r_0\| \;.$$

Note that in view of Theorem 3.2, the iteration would need to be just terminated, if

$$\frac{\|\hat{E}_k r_k\|}{\|r_k\|} = \frac{\|q_k - r_k\|}{\|r_k\|} > 1 \;,$$

which can be shown to be equivalent to the condition

$$t_k = \tau_k < \tfrac{1}{2} \;.$$

Restricted storage versions

For large n, one needs to restrict storage to some $m \cdot n$ such that

$$\begin{aligned} k_{\max} + 2 &= m \quad \text{for } (Aa) \;, \\ 2k_{\max} + 2 &= m \quad \text{for } (Bb) \;. \end{aligned}$$

Several options are possible to satisfy this restriction.

(I) Both Algorithms (A) and (B) can be just *restarted* after $k_{\max}$ iterations using $x_{k_{\max}}$ as the new starting guess x_0. Under the assumptions of the convergence theorems in Sections 3.1 and 3.2, these restricted variants can be shown to converge *linearly.*

(II) Both (A) and (B) can be modified by restricting the update loop to indices

$$i = 0, \ldots, k_{\max} - 1 \quad \text{(initial window)} \;.$$

This means a fixed preconditioning of the problem associated with $H_{k_{\max}}$ — with preconditioning from the right in (B) and from the left in (A).

Again, *linear* convergence can be shown under the assumptions made in Sections 3.1 and 3.2.

(III) Once $k > k_{\max}$ is reached, one may also consider restricting the update loop to indices

$$i = k - k_{\max}, \ldots, k \quad \text{(moving window)} .$$

For update (A), such a variant seems to be hard to interpret. For update (B), however, the update loop c) in Algorithm (B) can be solved to yield

$$\text{a)} \quad z_k = H_0 A \Delta_k + \sum_{i=0}^{k-1} \gamma_{ik} \cdot (\Delta_{i+1} - (1 - t_i)\Delta_i)$$

with factors

$$\text{b)} \quad \gamma_{ik} := \frac{q_i^* q_k}{t_i q_i^* q_i} .$$

Note that the corresponding factors γ_{ik} in Algorithm (A) would contain $\tilde{z}_i$. Obviously, the moving window variant in Algorithm (B) means replacing the above sum by its most recent iterative contributions. Such a variant might seem reasonable in view of the superlinear convergence properties of secant methods. However, it is unclear whether such a variant converges at all.

Each of the above restricted storage versions was implemented and tested on several examples. It turns out that all the window variants are *not* competitive with variant (I). Therefore, only (I) will be studied in Section 5.

5. Numerical Experiments

On the basis of the above derivation, the following storage restricted algorithms are compared here:

GB($k_{\max}$):	Update (A) with line search (a), Broyden's "good" method, restricted storage version (I).
BB($k_{\max}$):	Update (B) with line search (b), Broyden's "bad" method, restricted storage version (I).
GMRES–L($k_{\max}$):	Program GMRES(k) [13] with *left* preconditioning.
GMRES–R($k_{\max}$):	As above, but with the usual *right* preconditioning.

Any other variants of GB or BB are not included here, since their performance was not competitive with the two versions above. This excludes both *window* variants (II) and (III) of Section 4 and the different line searches $t_k \neq \tau_k$ for GB. The distinction of left and right preconditioning for GMRES has been made deliberately, since GB may be understood as some successively refined left preconditioner, whereas BB may be interpreted as some successively refined right preconditioner — which can be seen in the matrices $\tilde{E}_k$ for GB and $\hat{E}_k$ for BB.

Recall from Section 4 that BB($k_{\max}$) requires about *twice* the array storage as the other 3 codes. Moreover, the GB code and the GMRES codes supply the residual vector only, if explicitly wanted. If the successive iterates x_k are explicitly wanted (say, within an adaptive code or a nonlinear code [4]), then both GMRES codes need some modification, which in GMRES–R includes an additional preconditioned system solve per *each* iteration. Throughout the present section, only the rather simple preconditioning

$$H_0 = D^{-1} \ , \ \ D := \operatorname{diag}(a_{11}, \ldots, a_{nn}) \ , \tag{5.1}$$

is chosen. In a PDE context, this preconditioner takes care of the elliptic part (cf. [5]) — the rest must be taken care of by the rank–1 updates. A detailed study of different preconditioning techniques in a PDE setting will be given elsewhere.

Our test examples arise from convection-diffusion problems in 2–D of the following type:

$$\text{a)} \quad -\varepsilon \Delta u + \beta \cdot \nabla u = f \quad \text{on} \ \ \Omega \subset \mathbb{R}^2 \ ,$$

$$\text{b)} \quad u|_{\Gamma_0} = u_0 \ , \ \ \left.\frac{\partial u}{\partial n}\right|_{\Gamma_1} = 0 \quad \text{on} \ \partial\Omega = \Gamma_0 \cup \Gamma_1 \ , \ \ \Gamma_0 \cap \Gamma_1 = \emptyset \ .$$

In order to solve this problem, streamline upwind discretization with anisotropic adaptive grid refinement due to Kornhuber/Roitzsch [12]) is used.

Example 1. Circular layer problem

As a first special case of (5.1), we study a problem with a circular layer. For this, we set $\varepsilon = 10^{-5}$, $f = 0$ and $\beta = (y, -x)$. The domain Ω is $(0,1) \times (0,1) \backslash \Gamma_0$ with $\Gamma_0 = \{(x,y) : x = 0.5,\ y \leq 0.5\}$. On the inflow boundary, we prescribe

$$u_0(x,y) = \begin{cases} 0 & \text{if } y > 0.3 \\ 1 & \text{if } y \leq 0.3 \end{cases} \quad , \quad (x,y) \in \Gamma_0 \, .$$

In Fig. 5.1, the underlying grid with $n = 4238$ is shown. Starting point x_0 is the interpolated solution on a coarser grid.

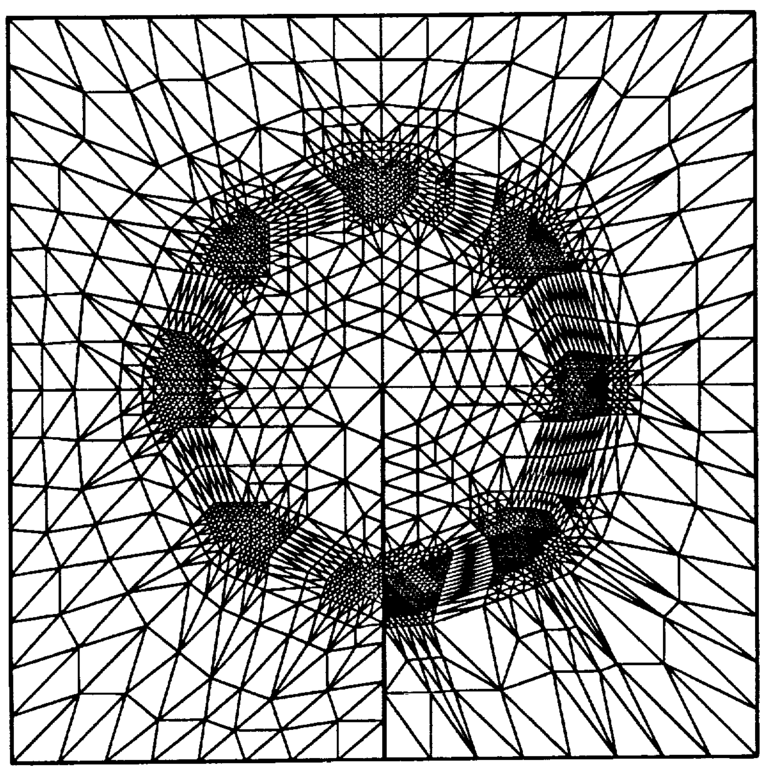

Figure 5.1: Anisotropic grid for Example 1, due to [12].

With only diagonal preconditioning, the BB code fails to solve the problem within n steps (nearly constant residual norm throughout the iteration). First, the behavior of GMRES with $k_{\max} \leq 10$ has been studied (Fig. 5.2 and 5.3), which led to the selection of GMRES–R(10) as best version. This version has been compared with GB(10) — see Fig. 5.4. To measure the error norms, the final iterate of a GB(10) run with required relative accuracy $\varepsilon = 10^{-8}$ in (4.3) has been taken as an estimate of the exact solution. Unlike the illustrative example in Section 3.3, the estimated error in GB(10) behaves only qualitatively as the true error — compare Fig. 5.5. Asymptotically, true and estimated error exhibit the same behavior, apart from oscillations caused by the k_{max}-restriction.

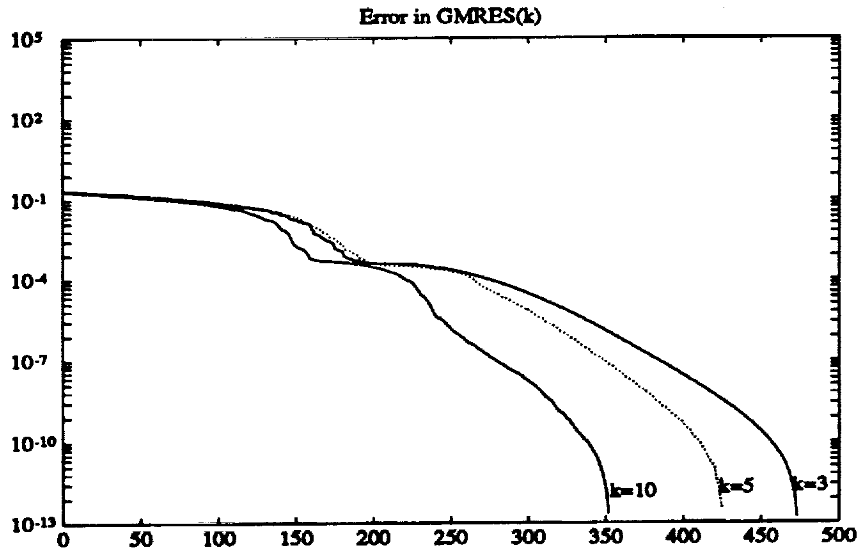

Figure 5.2: Comparison of 3 GMRES versions with right preconditioning in Example 1.

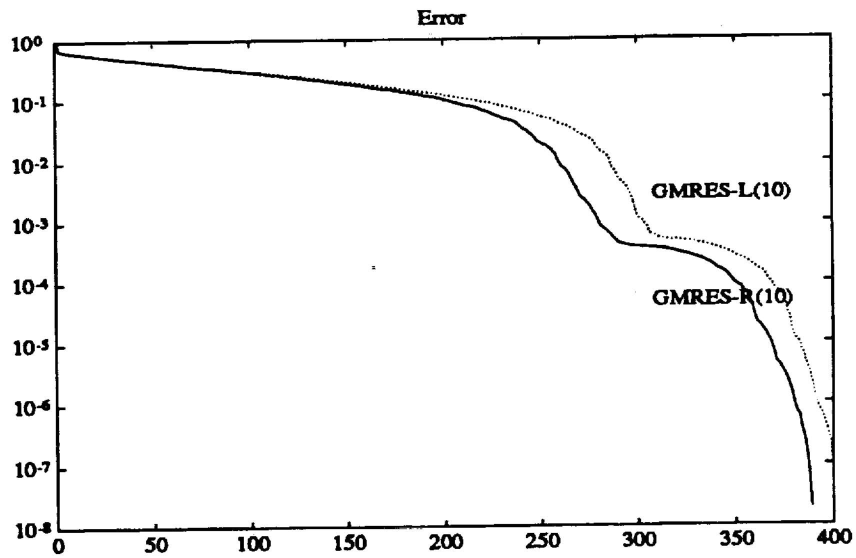

Figure 5.3: Comparison of left and right preconditioning in Example 1.

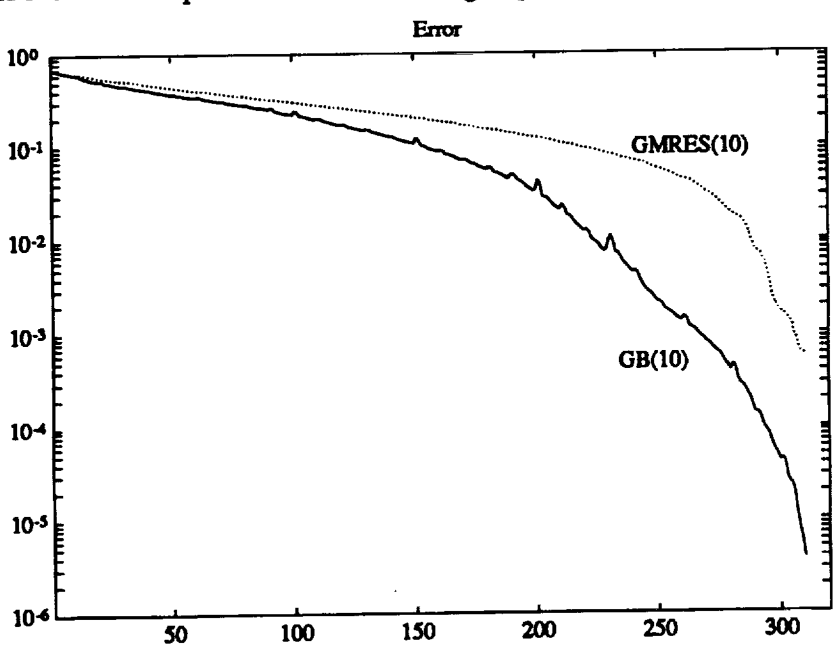

Figure 5.4: Comparison of error for GMRES and GB in Example 1.

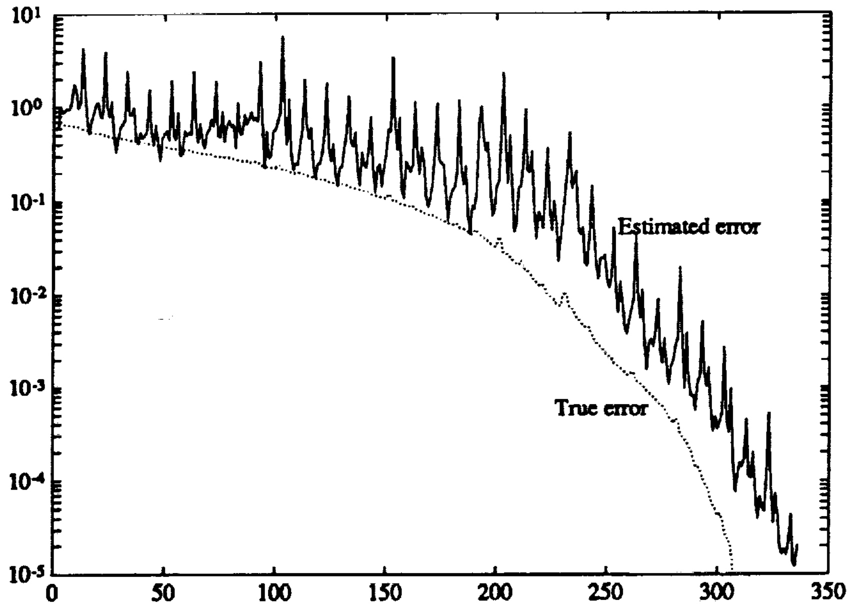

Figure 5.5: Comparison of true and estimated error in GB(10) in Example 1.

Example 2. Straight interior layer problem

The second test case was the convection–diffusion equation (5.1) on $\Omega = (0,1)\times(0,1)$ with a straight interior layer. To obtain this, we set $\varepsilon = 10^{-6}$, $f = 0$ and $\beta = (1.0;\ 0.5)$. The inflow boundary Γ_0 is given by $\Gamma_0 = \{(x,y) \in \partial\Omega : \max(x,y) < 1\}$. We prescribe the boundary condition

$$u_0(x,y) = \begin{cases} 0 & \text{if } y > 0.3 \\ 1 & \text{if } y \leq 0.3 \end{cases}, \quad (x,y) \in \Gamma_0 .$$

In Fig. 5.6, the final grid with $n = 2874$ is shown.

The behavior of the true error with diagonal preconditioning during the iteration is shown in Fig. 5.7. Once more, as in Example 1, GB appears to be the best solver. Note that the behavior in case $k_{\max} = 5$ is typical also for other choices of $k_{\max}$.

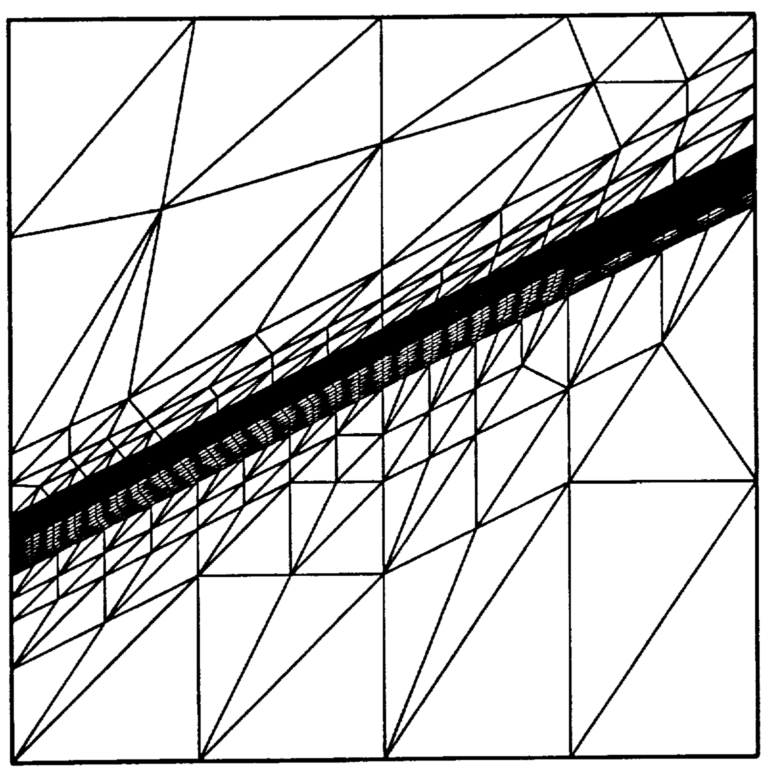

Figure 5.6: Anisotropic grid for Example 2, due to [12].

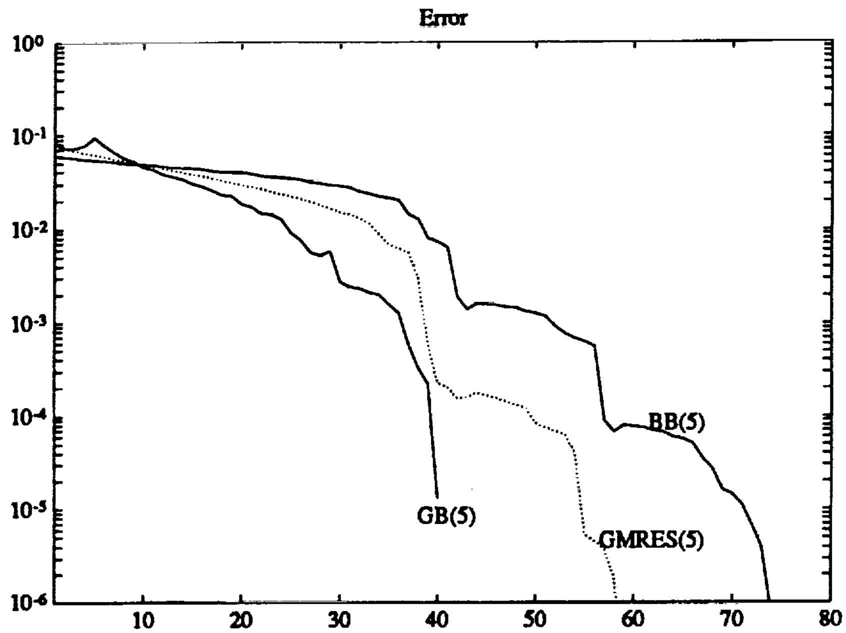

Figure 5.7: Comparison of error in GB(5), GMRES(5) and BB(5) in Example 2.

Conclusion

Two variants of secant methods based on Broyden's "good" and "bad" rank–1 updates have been studied. It turned out to be important that each update technique is combined with its associated line search. In comparison with GMRES, the up to now bad reputation of secant methods for linear problems is certainly not justified, if a reasonable preconditioning is at hand. Especially, the "good" Broyden variant appeared to be the more competitive, the larger the system dimension was. This observation is backed not only by the given examples, but also by further more extensive tests. In the context of multilevel discretizations of PDEs, the derived secant methods seem to have the structural advantage that the arising inner products can be especially adapted to the underlying PDE problem.

References

[1] C. G. Broyden: *A class of methods for solving nonlinear simultaneous equations.* Math. Comp., **19**, pp. 577–593 (1965).

[2] C. G. Broyden, J. E. Dennis, and J. J. Moré: *On the local and superlinear convergence of quasi-Newton methods.* J. Inst. Math. Appl., **12**, pp. 223–245 (1973).

[3] J. E. Dennis and R. B. Schnabel: *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, Englewood Cliffs, NJ. (1983).

[4] P. Deuflhard: *Global inexact Newton methods for very large scale nonlinear problems.* Preprint SC 90–2, Konrad-Zuse-Zentrum Berlin (1990) and SIAM J. Sci. Stat. Comput., to appear.

[5] P. Deuflhard, P. Leinen and H. Yserentant: *Concepts of an adaptive hierarchical finite element code.* IMPACT of Computing in Science and Engineering, **1**, pp. 3–35, (1989).

[6] T. Eirola and O. Nevanlinna: *Accelerating with rank-one updates.* Linear Algebra Appl., **121**, pp. 511–520 (1989).

[7] R. Fletcher: *Practical methods of optimization. Vol. 1: Unconstrained optimization.* John Wiley & Sons, New York (1980).

[8] D. M. Gay and R. B. Schnabel: *Solving systems of nonlinear equations by Broyden's method with projected updates.* In: Nonlinear Programming 3. O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), Academic Press, New York, pp. 245–281 (1978).

[9] D. M. Gay: *Some convergence properties of Broyden's method.* SIAM J. Numer. Anal., **16**, pp. 623–630 (1979).

[10] R. R. Gerber and F. T. Luk: *A generalized Broyden's method for solving simultaneous linear equations.* SIAM J. Numer. Anal., **18**, pp. 882–890 (1981).

[11] M. R. Hestenes and E. Stiefel: *Methods of conjugate gradients for solving linear systems.* J. Res. Natl. Bur. Stand., **49**, pp. 409–436 (1952).

[12] R. Kornhuber and R. Roitzsch: *On adaptive grid refinement in the presence of internal or boundary layers.* IMPACT of Computing in Science and Engineering, **2**, pp. 40–72 (1990).

[13] Y. Saad and M. H. Schultz: *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems.* SIAM J. Sci. Stat. Comput., **7**, pp. 856–869 (1986).

[14] C. Vuik: *A comparison of some GMRES-like methods.* Technical Report, Delft University of Technology, Delft (1990).