

Konrad-Zuse-Zentrum für Informationstechnik Berlin



Artur Walter

Improvement of Incomplete Factorizations
by a Sparse Secant Method

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Rabe KG Buch-und Offsetdruck Berlin

ISSN 0933-7911

Artur Walter

**Improvement of Incomplete Factorizations
by a Sparse Secant Method**

Abstract

In the present paper, the improvement of an incomplete factorization of a non-symmetric matrix A is discussed. Starting from the ideas of sparsity preserving quasi-Newton methods, an algorithm is developed which improves the approximation of A by the incomplete factorization maintaining the sparsity structure of the matrices. No renumbering of the unknowns or the admittance of additional fill-in is necessary. The linear convergence of the algorithm is proved under the assumption, that L and U^* have the same sparsity structure and an incomplete factorization with some reasonable approximation property exists. In combination with this algorithm, the method of incomplete factorization and its several modifications are applicable to a wider class of problems with improved convergence qualities. This is shown by a numerical example.

Key Words: non-symmetric linear system, sparse secant method, incomplete factorization.

AMS(MOS) Subject Classifications: 65F10, 65N20, 65N30

Contents

1. Introduction	1
2. The Incomplete Factorization	3
3. Broyden's Method for Linear Problems	8
4. Least Change Secant Updates for Matrix Products	12
4.1 The dense case	12
4.2 The sparse case	15
5. Convergence Analysis	19
6. Numerical Experiment	24
6.1 Implementation	24
6.2 Example	25
7. Conclusion	28
References	29

1. Introduction

The solution of large sparse systems of linear equations

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

is one of the most difficult tasks during the computation of finite difference or finite element approximations to partial differential equations. In the case of a selfadjoint differential operator a positive definite coefficient matrix A arises. For such matrices very powerful iterative techniques for solving (1.1) exist, e.g. conjugate gradient type methods and multi-grid methods.

However, solving non-selfadjoint problems, e.g. convection diffusion equations result in non-symmetric matrices A . For those matrices, it is, in general, by far more difficult to solve (1.1). In a recent paper [1] secant methods were proposed to treat these non-symmetric problems.

For positive definite coefficient matrices A incomplete factorizations introduced in [8] have been proved to be a good preconditioner for conjugate gradient type methods as well as smoother in multi-grid methods,[9], [10]. In the non-symmetric case the incomplete factorization was less successful, in convergence theory and in applications. So, many modification were introduced, [10]. Most of them try to order the unknowns in an appropriate manner, e.g. in stream-line direction for convection dominated problems. The major drawbacks are, that it seems difficult to find a renumber algorithm for general situations on anisotropic grids generated by adaptive solution algorithms, [6]. For machines with special hardware tailored to regular grids, as the Connection Machine, such a renumbering will slow down the overall performance dramatically.

With the approach introduced in [1], we try to modify the matrices L and U of the incomplete factorization during the iteration. In this way, the factorization is improved without altering the structure of the matrix A . The linear system (1.1) is looked at as a special case of a nonlinear system. For such systems with the sparse Jacobian A , sparsity preserving secant methods are known for long [4]. Starting from this basis we develop an algorithm for updating the product LU maintaining the sparsity structure imposed by the incomplete factorization on these matrices.

The paper is organized as follows. In Section 2, the incomplete factorization and the notation to describe sparse matrices is introduced. Some properties of this matrices are shown. In Sections 3, we repeat the sparse secant update. These results are applied in Section 4 to obtain a sparsity preserving update of the product LU . Convergence results for the new algorithm are presented

in Section 5. In Section 6 the implementation of the algorithm is shown and a numerical example which proves the advantage of the modified incomplete factorization is given. Finally, we make some concluding remarks.

Throughout this paper, $M^* = (m_{kj})$ denotes the transposed 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. In addition, the Frobenius norm $\|M\|_F = (\sum_{j,k} |m_{jk}|^2)^{1/2}$ will be used.

2. The Incomplete Factorization

For the matrix $A \in L(\mathbb{R}^n)$ nonsingular we assume throughout the paper that the factorization algorithm of Gauss without pivoting is possible. During this factorization process lower unit triangular matrices L_k and matrices $A^{(k)}$ are constructed such that

$$L_k \cdot A^{(k)} = A^{(k+1)} \quad (2.1)$$

and $A^{(n)} =: U$ is an upper triangular matrix. The matrices L_k are given by

$$L_k := I - l_k e_k^*$$

with

$$l_k := (0 \dots 0, l_{k+1,k}, \dots, l_{k,k})^* \text{ and } L_k^{-1} = I + l_k e_k^* .$$

At the end of the algorithm we obtain

$$\tilde{L} \cdot A = L_{n-1} \cdot L_{n-2} \cdot \dots \cdot L_1 \cdot A = U \quad (2.2)$$

and have to invert \tilde{L} . For the construction of $(\tilde{L})^{-1}$ we assume that

$$L_1^{-1} \cdot \dots \cdot L_k^{-1} = I + \sum_{i=1}^k l_i e_i^*$$

Since $e_i^* l_{k+1} = 0$ for $1 \leq i \leq k$ it is

$$L_1^{-1} \cdot \dots \cdot L_k^{-1} \cdot L_{k+1}^{-1} = I + \sum_{i=1}^{k+1} l_i e_i^* \quad (2.3)$$

and so

$$L := (\tilde{L})^{-1} = I + \sum_{i=1}^{n-1} l_i e_i^* . \quad (2.4)$$

This shows, that the factorization can be done row by row, with no coupling between the former modified rows. This is one reason which leads to the incomplete factorization introduced by MEIJERINK/VANDERVORST [8]. The idea

is, to construct matrices L_k and $A^{(k+1)}$ according to (2.1) but ignoring such elements $(L_{ij})_k$ and $(A_{ij})^{(k+1)}$ for which $A_{ij} = 0$. Of course, it is $LU \neq A$ in general for the matrices L and U constructed in this way. The advantage of this approximate factorization is, that there is no fill-in as with Gauss's algorithm. This is very important for sparse matrices arising in the context of finite element methods.

A representation similar to (2.4) is not possible for the matrix \tilde{L} in (2.2). If we start from (2.3)

$$L_k \cdot L_{k-1} \cdot \dots \cdot L_1 = I - \sum_{i=1}^k l_i e_i^*$$

we obtain for

$$\begin{aligned} L_{k+1} \cdot L_k \cdot \dots \cdot L_1 &= (I - l_{k+1} e_{k+1}^*) \left(I - \sum_{i=1}^k l_i e_i^* \right) \\ &= I - \sum_{i=1}^k l_i e_i^* - \sum_{i=1}^k l_{k+1} e_{k+1}^* l_i e_i^* . \end{aligned}$$

Since $e_{k+1}^* l_i \neq 0$ for $1 \leq i \leq k$ there is a coupling to all former rows. This means especially, that if we require a sparsity structure in row i this will influence all subsequent rows. The algorithm developed in the sequel operates on each row of a given pair of matrices L and U separately. Our main interest is to maintain a prescribed sparsity structure during the algorithm. With the above considerations it seems recommended to us to start from a factorization of the form $A = LU$ instead of $\tilde{L}A = U$.

We should mention here, that the most succesful sparse update algorithm for nonlinear problems, the algorithm of JOHNSON/AUSTRIA [5] and its sparse version recently analyzed by MARTINEZ [2] start from the representation (2.2). But for (2.3) we will focus on another algorithm.

For a description of the sparsity structure of a matrix we need some notations and definitions. The sparsity structure of a row of A is

$$V_i^A := \{v \in \mathbb{R}^n : e_j^* v = 0 \forall j \text{ such that } e_i^* A e_j = 0\} .$$

If X is a subspace of \mathbb{R}^n , we define by S^X the orthogonal projection on this subspace. The prescription of the sparsity pattern of a vector $v \in V_i^A$ on an arbitrary vector $s \in \mathbb{R}^n$ is

$$S^{V_i^A}(s) := \begin{cases} s_j & \text{if } v_j \neq 0 \\ 0 & \text{if } v_j = 0. \end{cases}$$

The indices of nonzero elements of a vector $v \in V_i^A$ is

$$J(V_i^A) := \{1 \leq i \leq n : \exists v \in V_i^A : v_i \neq 0\}. \quad (2.5)$$

It is always assumed, that the indices in (2.5) are ordered. This means, if

$$J(V_i^A) = \{i_1, i_2, \dots, i_m\} \quad \text{with } m = \dim(V_i^A) \quad (2.6)$$

it is $i_1 < i_2 < \dots < i_m$. With this notation, we define a restriction operator

$$\begin{aligned} \bar{S}^{V_i^A} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ v &\rightarrow (v_{i_1}, v_{i_2}, \dots, v_{i_m}). \end{aligned} \quad (2.7)$$

In our considerations often matrix–vector–products occur and we need the same operators for matrices. The cancellation of rows of a matrix $M \in \mathbb{R}^{n \times n}$ with respect to V_i^A is

$$\begin{aligned} T^{V_i^A} : \mathbb{R}^{n \times n} &\rightarrow \mathbb{R}^{n \times n} \\ m_{ij} &\mapsto \begin{cases} m_{ij} & \text{if } i \in J(V_i^A) \\ 0 & \text{otherwise} \end{cases} \quad 1 \leq i, j \leq n. \end{aligned} \quad (2.8)$$

The corresponding columns are set to zero by

$$\begin{aligned} \tilde{T}^{V_i^A} : \mathbb{R}^{n \times n} &\rightarrow \mathbb{R}^{n \times n} \\ m_{ij} &\mapsto \begin{cases} m_{ij} & \text{if } i \text{ and } j \in J(V_i^A) \\ 0 & \text{otherwise.} \end{cases} \quad 1 \leq i, j \leq n. \end{aligned} \quad (2.9)$$

The analogue of (2.7) for matrices is

$$\begin{aligned} \bar{T}^{V_i^A} : \mathbb{R}^{n \times n} &\rightarrow \mathbb{R}^{m \times m} \\ M &\mapsto \begin{pmatrix} m_{i_1, i_1} & \dots & m_{i_1, i_n} \\ \vdots & & \vdots \\ m_{i_m, i_1} & \dots & m_{i_m, i_m} \end{pmatrix}. \end{aligned} \quad (2.10)$$

where the indices are given by (2.6).

In the development of the algorithm below, often a special matrix occurs which is defined here:

$$M^{(i)} := \begin{pmatrix} m_{1,1} & \dots & \dots & \dots & \dots & m_{1,n} \\ & m_{2,2} & & & & \vdots \\ & & \ddots & & & \vdots \\ & & & m_{i-1,i-1} & \dots & m_{i-1,n} \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix}. \quad (2.11)$$

Remark. In our applications, the matrix A stems from a finite element discretization of a non-selfadjoint partial differential equation, KORNHUBER / ROITZSCH [6]. The resulting matrix A is in general non-symmetric but has a symmetric sparsity structure. This means, that A and A^* have the same position of non-zero elements. The structure is maintained during the incomplete factorization process and so L and U^* have the same sparsity pattern. This fact is very important for the subsequent considerations and *we will restrict ourselves to such matrices A* . The special structure of the matrix (2.11) is analyzed in the following

Lemma 1 *Assume that U is given by an incomplete LU-factorization process of A and $x \in \mathbb{R}^n$. Then it is*

$$S^{V_i^A}(U^{(i)}x) = \tilde{T}^{V_i^A}(U^{(i)})S^{V_i^A}(x) \quad (2.12)$$

$$\bar{S}^{V_i^A}(U^{(i)}x) = \bar{T}^{V_i^A}(U^{(i)})\bar{S}^{V_i^A}(x). \quad (2.13)$$

Proof. From (2.8) we get

$$S^{V_i^A}(U^{(i)}x) = T^{V_i^A}(U^{(i)})x.$$

The fact that L^* and U have the same sparsity structure yields

$$\tilde{T}^{V_i^A}(U^{(i)}) = T^{V_i^A}(U^{(i)}) .$$

From this we obtain (2.12)

$$S^{V_i^A}(U^{(i)}x) = \tilde{T}^{V_i^A}(U^{(i)})x = \tilde{T}^{V_i^A}(U^{(i)})S^{V_i^A}(x) .$$

The elimination of zero rows and columns j for $j \notin J(V_i^A)$ with the operator (2.10) gives (2.13)

$$\bar{S}^{V_i^A}(U^{(i)}x) = \bar{T}^{V_i^A}(U^{(i)})\bar{S}^{V_i^A}(x) .$$

■

Remark. The proof of Lemma (1) shows, that it is enough to assume, that L^* and U have the same sparsity structure. This fact may be advantageous in some circumstances. All considerations here are also applicable to variants of incomplete factorizations introduced in [10].

3. Broyden's Method for Linear Problems

The application of Broyden's good method , see e.g. [7] to a linear problem

$$Ax = b \tag{3.1}$$

with $A \in L(\mathbb{R}^n)$ nonsingular and $b \in \mathbb{R}^n$ results in the following algorithm for $k = 1, \dots, k_{max}$:

$$\begin{aligned} \text{a) solution step} & : B_k s_k = r_k \\ \text{b) update step} & : x_{k+1} = x_k + t_k s_k \\ \text{c)} & : B_{k+1} = B_k + (y_k - B_k s_k) \frac{s_k^*}{s_k^* s_k} . \end{aligned} \tag{3.2}$$

Here the notation

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

is used. The line search parameter t_k can be calculated due to [1] or set to one, as it is done in the following. An initial approximation x_0 of the solution and B_0 of A is needed to start the algorithm. The update formula (3.2.c) for the matrix B_k is constructed by satisfying the two demands

$$B_{k+1} s_k = y_k \quad , \tag{3.3}$$

the secant condition and the least change condition

$$\min \|B_{k+1} - B_k\|_F \quad \text{subject to} \quad B_{k+1} \in Q(y, s) \tag{3.4}$$

where $Q(y, s) := \{M \in L(\mathbb{R}^n) : Ms = y\}$.

The initial approximation B_0 of A should have the same sparsity structure as A . With the update (3.2.c), this structure is destroyed. So, one looks for sparsity preserving updates, as was first done by SCHUBERT [4]. It seems worthwhile to repeat the derivation to illuminate the problems in the derivation of the new algorithm.

The set of all matrices with the same sparsity structure of A is

$$SP(A) := \{M \in L(\mathbb{R}^n) : M^* e_i \in V_i^A, \quad 1 \leq i \leq n\}$$

The sparse update is derived by obeying (3.3) and (3.4) at each row of B_{k+1} separately, motivated by the fact that

$$\|B_{k+1} - B_k\|_F^2 = \sum_{i=1}^n \|e_i^*(B_{k+1} - B_k)\|^2. \quad (3.5)$$

Lemma 2 *Let $\alpha \in \mathbb{R}$, $v \in \mathbb{R}^n \setminus \{0\}$ and $I \subseteq \{1, \dots, n\}$ be given such that $\sum_{i \in I} v_i^2 \neq 0$. Then the unique solution to*

$$\min_{x \in \mathbb{R}^n} \|x\| \quad \text{subject to } v^* x = \alpha$$

is

$$x = \alpha v / \sum_{i \in I} v_i^2. \quad (3.6)$$

Proof. If $\alpha = 0$ the lemma is true. Otherwise, we introduce on the space $Z := \{x \in \mathbb{R}^n : x_i = 0 \ \forall i \notin I\}$ the scalar product

$$\langle x, y \rangle := \sum_{i \in J(Z)=I} x_i y_i. \quad (3.7)$$

Now (3.6) follows from the fact that

$$[v^*]^+ = \frac{v}{\langle u, v \rangle},$$

where $[\]^+$ denotes the Moore–Penrose pseudoinverse with respect to the norm introduced by the scalar product (3.7). ■

With this Lemma, it is possible to derive a sparsity preserving update.

Theorem 3 *Let $B_k \in L(\mathbb{R}^n) \cap SP(A)$ be given. Then the unique solution to*

$$\min \|B_{k+1} - B_k\|_F \quad \text{subject to } B_{k+1} \in Q(y, s) \cap SP(A) \quad (3.8)$$

is

$$B_{k+1} = B_k + \sum_{i=1}^n [S^{V_i^A}(s)^* S^{V_i^A}(s)]^+ e_i e_i^* (y - B_k s) S^{V_i^A}(s)^*. \quad (3.9)$$

Proof. Define $C := B_{k+1} - B_k$. With (3.5), we can rewrite (3.8) as

$$\min_{C \in L(\mathbb{R}^n) \cap SP(A)} \sum_{i=1}^n \|C_i\|^2 \quad \text{subject to } C_i s = e_i^*(y - B_k s). \quad (3.10)$$

where $C_i := e_i^* C$. This is equivalent to n disjoint problems

$$\min_{C_i \in \mathbb{R}^n \cap V_i^A} \|C_i\|^2 \quad \text{subject to } C_i s = e_i^*(y - B_k s).$$

By the construction of (3.9) it is $C_i \in V_i^A$. This yields

$$\|C_i\|^2 = \sum_{j=1}^n C_{ij}^2 = \sum_{j \in J(V_i^A)} C_{ij}^2$$

and

$$C_i s = \sum_{j=1}^n C_{ij} s_j = \sum_{j \in J(V_i^A)} C_{ij} s_j = \sum_{j \in J(V_i^A)} C_{ij} S^{V_i^A}(s)_j.$$

In the case $S^{V_i^A}(s) \neq 0$ we can apply Lemma 2 and obtain

$$C_i = [S^{V_i^A}(s)^* S^{V_i^A}(s)]^+ e_i^*(y - B_k s) S^{V_i^A}(s)^*.$$

If $S^{V_i^A}(s) = 0$ no modification of B_k is possible since $[S^{V_i^A}(s)^* S^{V_i^A}(s)]^+ = 0$. The sparsity structure is kept and it remains to show that the secant condition is fulfilled for the i -th row of B_{k+1} .

$$e_i^* B_{k+1} s = (e_i^* B_{k+1}) S^{V_i^A}(s) = 0 = (e_i^* A) S^{V_i^A}(s) = e_i^* A s = y_i.$$

Note, that it is essential that the i -th row of A and B_{k+1} have the same sparsity structure to show the secant condition. ■

Remark. One can also fix any element of B_k . We only have to set the corresponding element of A to zero and incorporate it in the sparsity structure. This is done e.g. for the diagonal elements of L below.

In some circumstances it is appropriate to use a weighted Frobenius norm. The update corresponding to (3.2.c) is given by the

Corollary 4 *With the same notation as in Theorem 3 and the nonsingular matrices $M, N \in L(\mathbb{R}^n)$ the unique solution of*

$$\min \|M(B_{k+1} - B_k)N\|_F \quad \text{subject to } B_{k+1} \in Q(y, s) \quad (3.11)$$

is

$$B_{k+1} = B_k + (y - B_k s) \frac{(N^{(-1)}s)^* N^{-1}}{(N^{-1}s)^* N^{-1}s} . \quad (3.12)$$

Proof. Define $\bar{C} := MCN$ for $C \in L(\mathbb{R}^n)$. Then (3.11) can be rewritten as

$$\min \|\bar{B}_{k+1} - B_k\|_F \quad \text{subject to } \bar{B}_{k+1} \in Q(Ny, M^{-1}s) .$$

The solution is given by Theorem 3. ■

In a recent paper [11] another sparsity preserving update was proposed. There, the matrix A is used as a weight in (3.11). The zero elements of A annihilate the corresponding contributions in (3.12). The analysis of the resulting algorithm is possible in the same way as done below and will be given elsewhere.

4. Least Change Secant Updates for Matrix Products

4.1 The dense case

During the realization of Broyden's method it is necessary to solve the linear equation in (3.2). This should be easier than to solve (3.1) directly. There are, in principle, two possibilities. First, we can start with an approximation B_0 of A , e.g. the diagonal part of A . Then we can update $H_k := B_k^{-1}$ with the Sherman–Morrison–Lemma, as it is done in [1]. Here we want to start with an approximate factorization of A , the incomplete LU-factorization, introduced above. The question arises, how can such a factorization be updated in the framework presented in the previous chapter.

In this situation, the solution step of Broyden's method is

$$L_k U_k s_k = r_k .$$

We seek an updated of L_k and U_k subject to the secant condition

$$L_{k+1} U_{k+1} s_k = y_k \tag{4.1}$$

and the least change condition

$$\min \|L_{k+1} U_{k+1} - L_k U_k\|_F . \tag{4.2}$$

The matrices L and U don't have any prescribed sparsity structure yet. As a first step, we look at the "dense case". An additional unknown $w \in \mathbb{R}^n$ is introduced to make the problem more treatable. The approach is presented for different choices of w for completeness. All iteration indices k are dropped and the abbreviations $\bar{L} := L_{k+1}$, $\bar{U} := U_{k+1}$ are used. In this notation (4.1) is

$$\bar{L} w = y \quad \text{and} \quad \bar{U} s = w . \tag{4.3}$$

The least change condition (4.2) is split into

$$\min \|\bar{L} - L\|_F \quad \text{and} \quad \min \|\bar{U} - U\|_F \tag{4.4}$$

The combination of (4.3) and (4.4) provides an U -update of the form

$$\bar{U} = U + (w - Us) \frac{s^*}{s^*s} \quad (4.5)$$

and similar for the matrix L

$$\bar{L} = L + (y - Lw) \frac{w^*}{w^*w} . \quad (4.6)$$

For every w the secant condition (4.1) is guaranteed.

Now, we want to determine w so that (4.2) is fulfilled. From (4.5) and (4.6) we have for the product

$$\bar{L}\bar{U} = LU \left(I - \frac{ss^*}{s^*s} \right) + y \frac{s^*}{s^*s} + (y - Lw) \frac{w^*}{w^*w} U \left(I - \frac{ss^*}{s^*s} \right) . \quad (4.7)$$

With any matrix $M \in Q(y, s)$ the difference in (4.2) is

$$\|\bar{L}\bar{U} - LU\|_F = \left\| \left(M - LU + (y - Lw) \frac{w^*}{w^*w} U \right) \left(I - \frac{ss^*}{s^*s} \right) \right\|_F . \quad (4.8)$$

The first outstanding choice for w is

$$Lw = y . \quad (4.9)$$

The insertion in (4.8) gives $\|\bar{L}\bar{U} - LU\|_F \leq \|M - LU\|_F$ for all $M \in Q(y, s)$. This means with (4.6), that L is fixed. The resulting algorithm was proposed and analyzed by DENNIS/MARWIL [3]. Note, that from (4.7)

$$\|\bar{L}\bar{U} - A\|_F = \left\| \left(LU - A + (y - Lw) \frac{w^*}{w^*w} U \right) \left(I - \frac{ss^*}{s^*s} \right) \right\|_F$$

and with the choice (4.9)

$$\|\bar{L}\bar{U} - A\|_F \leq \|LU - A\|_F \quad (4.10)$$

a better approximation of the matrix A is obtained.

The second choice suggested by (4.8) is

$$Us = w . \quad (4.11)$$

With this setting (4.8) is

$$\begin{aligned}
\|\bar{L}\bar{U} - LU\|_F &= \left\| (M - LU) \left(I - \frac{sw^*U}{w^*w} \right) \left(I - \frac{ss^*}{s^*s} \right) \right\|_F \\
&= \left\| (M - LU)U^{-1} \left(I - \frac{ww^*}{w^*w} \right) U \right\|_F \\
&\leq \|M - LU\|_F \|U^{-1}\| \|U\|.
\end{aligned}$$

The analogue of (4.10) is

$$\|\bar{L}\bar{U} - A\|_F \leq \|LU - A\|_F \|U^{-1}\| \|U\|. \quad (4.12)$$

The choice (4.11) implies with (4.5) that no U -update is possible.

A third possibility, proposed by (4.8) is

$$U^*w = s \quad (4.13)$$

since $s^*(I - ss^*/s^*s) = 0$. This directly yields

$$\|\bar{L}\bar{U} - LU\|_F \leq \|M - LU\|_F$$

and

$$\|\bar{L}\bar{U} - A\|_F \leq \|LU - A\|_F,$$

which seems to be the best choice.

In multi-grid applications often the L - D - U -Gauss decomposition, where D is a diagonal matrix and $B = LDU$ is used. In this context all information should be contained in D . Therefor the matrices L and U are fixed. The secanat condition is

$$L\bar{D}Us = y.$$

The least change condition reads now

$$\min \|LDU - L\bar{D}U\|_F = \min \|L(D - \bar{D})U\|_F. \quad (4.14)$$

Corollary 4 gives the solution of (4.14):

$$\bar{D} = D + (L^{-1}y - DUs) \frac{s^*}{s^*s} U^{-1}.$$

It is worth noticing that the vector $w = (s^*U^{-1})^*$ from (4.13) also occurs here.

4.2 The sparse case

To take advantage of the different updates development above, it should be possible to construct sparsity preserving versions of these formulas which also satisfy the secant condition.

In Theorem 3 the intersection $SP(A) \cap Q(y, s)$ may be empty. In this case, we obtain from the update formula (3.9) a matrix out of $SP(A)$ which is closest to all matrices in $Q(y, s)$ in the Frobenius norm. This is reasonable in the context of optimization. Since our aim is to solve the linear system (3.1), we claim that the secant condition is more important. With this argument, formulas violating the secant condition, used in optimization, are rejected.

The update proposed in the previous section was

$$U^*w = s \quad , \quad \bar{U}s = w \quad , \quad \bar{L}w = y \quad . \quad (4.15)$$

$$\bar{U} = U + (w - Us) \frac{s^*}{s^*s} \quad , \quad \bar{L} = L + (y - Lw) \frac{w^*}{w^*w} \quad (4.16)$$

$$\bar{L}\bar{U} = LU + (y - LUs) \frac{s^*}{s^*s} \quad (4.17)$$

The application of Schubert's sparse update developed on the previous chapter provides the U -update

$$\bar{U} = U + \sum_{i=1}^n [S^{V_i^U}(s)^* S^{V_i^U}(s)]^+ e_i e_i^* (w - Us) S^{V_i^U}(s)^* \quad (4.18)$$

and in the same manner the L -update

$$\bar{L} = L + \sum_{i=1}^n [S^{V_i^L}(w)^* S^{V_i^L}(w)] e_i e_i^* (y - LUs) S^{V_i^L}(w)^* \quad (4.19)$$

This gives by construction (4.15), if

$$S^{V_i^U}(w) \neq 0 \quad \text{and} \quad S^{V_i^L}(s) \neq 0 \quad 1 \leq i \leq n \quad . \quad (4.20)$$

Any other case needs special consideration.

If w and s have at least one non-zero component, e.g. w at the position i , it is always possible to achieve (4.20) . Set

$$\bar{J}(w) := \{1 \leq j \leq n : w_j = 0\}$$

and construct a transformation matrix

$$N^{-1} := \prod_{j \in \bar{J}} (I + e_j e_j^*) .$$

The vector $v := N^{-1}w$ has no zero component and the update (4.18) is given by Corollary 4

$$\bar{U} = U + \sum_{i=1}^n [S^{V_i^U}(v)^* S^{V_i^U}(v)]^+ e_i e_i^* (w - Us) S^{V_i^U}(v)^* N^{-1}$$

with the weights $N = \prod_{j \in \bar{J}} (I - e_j e_j^*)$ and $M = I$. The same is possible for the update (4.19).

The difficulty steems from the fact that the product LU and A have different sparsity structures. Assume $S^{V_i^L}(w) = 0$ in equation (4.19) and hence for the i -th row

$$e_i^* \bar{L} w = e_i^* \bar{L} S^{V_i^L}(w) = 0 = e_i^* U^* S^{V_i^U}(w) . \quad (4.21)$$

It is $S^{V_i^U} \subset S^{V_i^A}$ and $S^{V_i^L} \subset S^{V_i^A}$, but there is no matrix at hand with the prescribed sparsity structure to show that (4.21) is equal y_i .

Our aim is to construct a method for updating the factorization LU in the way shown above and maintaining the sparsity structure. This is achieved by splitting (4.1) and (4.2) in n disjoint problems corresponding to the rows of LU . The i -th row of (4.1) is

$$e_i^* \bar{L} \bar{U} s = e_i^* \bar{R}_i \bar{U}^{(i)} s = e_i^* \bar{R}_i w^{(i)}$$

where

$$\bar{R}_i := (\bar{L}_{i,1}, \bar{L}_{i,2}, \dots, \bar{L}_{i,i-1}, \bar{U}_{i,i}, \dots, \bar{U}_{i,n})$$

and

$$w^{(i)} := \bar{U}^{(i)} s$$

with $\bar{U}^{(i)}$ defined above.

Since the update is row by row, the rows $1, 2, \dots, i-1$ are already updated. This allows to rewrite (4.2) as

$$\begin{aligned} \|\bar{L}\bar{U} - LU\|_F^2 &= \sum_{i=1}^n \|e_i^*(\bar{L}\bar{U} - LU)\|^2 \\ &= \sum_{i=1}^n \|\bar{R}_i\bar{U}^{(i)} - R_iU^{(i)}\|^2 \\ &= \sum_{i=1}^n \|(\bar{R}_i - R_i)U^{(i)}\|^2. \end{aligned} \quad (4.22)$$

The problem corresponding to (4.2) is now to find a solution of

$$\min_{R_i \in V_i^A} \|(\bar{R}_i - R_i)U^{(i)}\|^2 \quad \text{subject to} \quad (\bar{R}_i - R_i)U^{(i)}s = e_i^*(y - R_iw^{(i)}).$$

The solution in the case $S^{V_i^A}(w^{(i)}) \neq 0$ is given by Lemma 2

$$\bar{R}_i = R_i + [S^{V_i^A}(w^{(i)})^* S^{V_i^A}(w^{(i)})]^+(y_i - R_iw^{(i)})w^{(i)*}. \quad (4.23)$$

Broyden's good method (4.17) reads, with the introduced notation, for the i -th row

$$\bar{R}_i = R_i + (y_i - R_iU^{(i)}s) \frac{S^{V_i^A}(U^{(i)}s)^*}{S^{V_i^A}(U^{(i)}s)^* S^{V_i^A}(U^{(i)}s)}. \quad (4.24)$$

The equation (4.22) suggests to look for a least change condition in a weighted Frobenius norm. With the weight $N^{(i)} = (U^{(i)})^{-1}$, the update given by Corollary 4 is just (4.23). This shows that our proposed method is Broyden's good update with some special weighting functions. Summing up the above considerations, we have the following

Theorem 5 *The unique solution of*

$$\min_{R_i \in V_i^A} \|\bar{R}_i - R_i\| \quad \text{subject to} \quad \bar{R}_i w^{(i)} = y_i \quad (4.25)$$

is given by

$$\bar{R}_i = R_i + [S^{V_i^A}(w^{(i)})^* S^{V_i^A}(w^{(i)})]^+(y_i - R_iw^{(i)})S^{V_i^A}(w^{(i)})^*. \quad (4.26)$$

Proof. The case $S^{V_i^A}(w^{(i)}) \neq 0$ was discussed above. It remains to show that in the case $S^{V_i^A}(w^{(i)}) = 0$ the secant condition is fulfilled.

Assume $S^{V_i^A}(w^{(i)}) = 0$. This implies

$$\bar{S}^{V_i^A}(w^{(i)}) = 0 = \bar{S}^{V_i^A}(U^{(i)}s) = \bar{T}^{V_i^A}(U^{(i)})\bar{S}^{V_i^A}(s)$$

by Lemma 2. Since $\bar{T}^{V_i^A}(U^{(i)})$ is an upper triangular matrix with non-zero diagonal elements, it follows that

$$\bar{S}^{V_i^A}(s) = 0 \quad \text{and} \quad S^{V_i^A}(s) = 0.$$

This gives for the secant condition

$$y_i = e_i^* A s = e_i^* A S^{V_i^A}(s) = 0 = \bar{R}_i w^{(i)}$$

and the Theorem is true. ■

5. Convergence Analysis

The interpretation of the algorithm (4.26) as a weighted update makes it necessary to control this variable weights during the iteration. Throughout this chapter it is assumed that

$$\|\bar{U}^{(i)} - \mathcal{U}^{(i)}\| \leq \alpha \quad \text{for } 1 \leq i \leq n, \quad (5.1)$$

with α determined below. The matrices \mathcal{L} and \mathcal{U} are given by a regular Gauss factorization process such that $\mathcal{L}\mathcal{U} = A$. Therefore $\mathcal{L} \notin SP(A)$ and $\mathcal{U} \notin SP(A)$. In the following the notations \mathcal{R} and \mathcal{M} correspond to these matrices in an obvious way.

The reduced matrix $\bar{T}^{V_i^A}(\mathcal{U}^{(i)})$ is nonsingular for all i . There are constants $\beta \geq 0$ and $\kappa \geq 0$ such that

$$\|[\bar{T}^{V_i^A}(\mathcal{U}^{(i)})]^{-1}\| \leq \beta \quad (5.2)$$

and

$$\|\mathcal{L}\| \leq \kappa, \quad \|\mathcal{U}\| \leq \kappa. \quad (5.3)$$

Lemma 6 *Under the assumptions (5.1) and (5.2) there is a constant $\gamma > 0$ with*

$$\gamma = \frac{\beta}{1 - \alpha\beta} \quad \text{if } \alpha\beta < 1 \quad (5.4)$$

such that

$$\gamma \|S^{V_i^A}(w^{(i)})\| \geq \|S^{V_i^A}(s)\|. \quad (5.5)$$

Proof. The assumption (5.1) implies

$$\|\bar{T}^{V_i^A}(\bar{U}^{(i)}) - \bar{T}^{V_i^A}(\mathcal{U}^{(i)})\| \leq \alpha, \quad 1 \leq i \leq n$$

and with the Banach perturbation lemma

$$\|[\bar{T}^{V_i^A}(\bar{U}^{(i)})]^{-1}\| \leq \frac{\beta}{1 - \alpha\beta} =: \gamma_i, \quad 1 \leq i \leq n.$$

Let $\gamma := \max_{1 \leq i \leq n} \{\gamma_i\}$. Lemma 1 yields

$$\bar{S}^{V_i^A}(w^{(i)}) = \bar{T}^{V_i^A}(\bar{U}^{(i)})\bar{S}^{V_i^A}(s)$$

This implies

$$\bar{S}^{V_i^A}(s) = [\bar{T}^{V_i^A}(\bar{U}^{(i)})]^{-1}\bar{S}^{V_i^A}(w^{(i)})$$

and

$$\|\bar{S}^{V_i^A}(s)\| \leq \gamma \|\bar{S}^{V_i^A}(w^{(i)})\|. \quad (5.6)$$

Since $\|\bar{S}^{V_i^A}(s)\| = \|S^{V_i^A}(s)\|$ equation (5.5) is true. \blacksquare

For the convergence proof we need another technical Lemma. Let the matrix $M \in \mathbb{R}^{n \times n}$ be defined by

$$M := \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} \quad (5.7)$$

Lemma 7 *Under the assumptions (5.1) to (5.4) there is a constant $c > 0$ depending on the dimension n of A such that*

$$\|\bar{M} - \mathcal{M}\|_F \leq c \|M - \mathcal{M}\|_F \quad (5.8)$$

Proof. For shortness of the presentation let $v^{(i)} := S^{V_i^A}(w^{(i)})$. In view of (3) we can restrict ourselves to the case $v^{(i)} \neq 0$. The update formula (4.26) gives

$$\bar{R}_i - \mathcal{R}_i = (R_i - \mathcal{R}_i) \left(I - \frac{v^{(i)}v^{(i)*}}{v^{(i)*}v^{(i)}} \right) + (y_i - \mathcal{R}_i v^{(i)}) \frac{v^{(i)*}}{v^{(i)*}v^{(i)}} \quad (5.9)$$

and by orthogonality of the vectors on the right-hand side

$$\|\bar{R}_i - \mathcal{R}_i\|^2 = \left\| (R_i - \mathcal{R}_i) \left(I - \frac{v^{(i)}v^{(i)*}}{v^{(i)*}v^{(i)}} \right) \right\|^2 + \left\| (y_i - \mathcal{R}_i v^{(i)}) \frac{v^{(i)*}}{v^{(i)*}v^{(i)}} \right\|^2.$$

The first term on the right-hand side of (5.9) is an orthogonal projection and so

$$\left\| (R_i - \mathcal{R}_i) \left(I - \frac{v^{(i)} v^{(i)*}}{v^{(i)*} v^{(i)}} \right) \right\| \leq \|R_i - \mathcal{R}_i\|. \quad (5.10)$$

Note that $A_i = \mathcal{R}_i \mathcal{U}^{(i)}$ and hence the second term on the right hand side of equation (5.9) is with Lemma 6

$$\begin{aligned} \left\| y_i - \mathcal{R}_i v^{(i)} \frac{v^{(i)*}}{v^{(i)*} v^{(i)}} \right\| &\leq \|A_i S^{V_i^A}(s) - \mathcal{L}_i \bar{U}^{(i)} S^{V_i^A}(s)\| \cdot (\|S^{V_i^A}(w^{(i)})\|)^{-1} \\ &\leq \gamma \|\mathcal{L}_i\| \|\mathcal{U}^{(i)} - \bar{U}^{(i)}\| \end{aligned}$$

Summing up these results provides

$$\begin{aligned} \|\bar{R}_i - \mathcal{R}_i\|^2 &\leq \|R_i - \mathcal{R}_i\|^2 + \gamma \|\mathcal{L}_j\|^2 \|\bar{U}^{(i)} - \mathcal{U}^{(j)}\|^2 \\ &\leq \|R_i - \mathcal{R}_i\|^2 + \gamma \kappa \alpha. \end{aligned} \quad (5.11)$$

For the above defined matrix M it is

$$\|\bar{M}_i - \mathcal{M}_i\|^2 = \|\bar{L}^{(i+1)} - \mathcal{L}^{(i+1)}\|^2 + \|\bar{U}^{(i+1)} - \mathcal{U}^{(i+1)}\|^2. \quad (5.12)$$

Let

$$\varepsilon_i := \sum_{j=1}^i \|\bar{R}_j - \mathcal{R}_j\|^2 = \|\bar{M}^{(i)} - \mathcal{M}^{(i)}\|_F^2 \quad (5.13)$$

The assumption (5.1) implies $\|\bar{U}^{(j)} - \mathcal{U}^{(j)}\| \leq \alpha$ for $j \leq i$. With the definition (5.13) and (5.12) it is for ε_{j-1}

$$\varepsilon_{j-1} = \|\bar{R}_{j-1} - \mathcal{R}_{j-1}\|^2 = \|\bar{L}^{(j)} - \mathcal{L}^{(j)}\|^2 + \|\bar{U}^{(j)} - \mathcal{U}^{(j)}\|^2 \geq \|\bar{U}^{(j)} - \mathcal{U}^{(j)}\|^2.$$

Hence from (5.11)

$$\varepsilon_j - \varepsilon_{j-1} \leq \delta_j + \gamma \kappa \varepsilon_{j-1}$$

where

$$\delta_j := \|R_j - \mathcal{R}_j\|^2.$$

This gives

$$\varepsilon_j \leq \delta_j + (1 + \gamma\kappa)\varepsilon_{j-1} \leq \sum_{i=0}^j (1 + \gamma\kappa)^i \delta_{j-i} \leq (1 + \gamma\kappa)^j \sum_{j=0}^i \delta_j$$

and

$$\|\bar{M} - \mathcal{M}\|_F^2 \leq (1 + \gamma\kappa)^n \sum_{j=0}^n \|R_j - \mathcal{R}_j\|^2 \leq (1 + \gamma\kappa)^n \|M - \mathcal{M}\|_F^2.$$

So, the Lemma is true. ■

Theorem 8 *With the assumptions of Lemma 7 and in addition α form (5.1) such that*

$$(2\alpha\kappa + \kappa)2\alpha\kappa \cdot \|A^{-1}\| < 1 \quad (5.14)$$

the algorithm is linear convergent, i.e. there is a constant $\tilde{c} \in (0, 1)$ such that

$$\|e_{k+1}\| \leq \tilde{c}\|e_k\| \quad \text{for all } k \quad (5.15)$$

Proof. Equation (5.15) is shown in the usual way by an induction over k . The update formula (4.26) gives for the error e_{k+1}

$$e_{k+1} = (I - (L_k U_k)^{-1} A) e_k = (L_k U_k)^{-1} (L_k U_k - A) e_k. \quad (5.16)$$

So, we have to control the two factors on the right-hand side.

It is

$$\begin{aligned} (L_k U_k - A) &= L_k(U_k - \mathcal{U}) + (L_k - \mathcal{L})\mathcal{U} \\ &= ((L_k - \mathcal{L}) + \mathcal{L})(U_k - \mathcal{U}) + (L_k - \mathcal{L})\mathcal{U} \end{aligned}$$

and

$$\|L_k U_k - A\| \leq (\|L_k - \mathcal{L}\| + \kappa)\|U_k - \mathcal{U}\| + \|L_k - \mathcal{L}\| \cdot \kappa. \quad (5.17)$$

From Lemma 7 it follows that

$$\begin{aligned}\|L_k - \mathcal{L}\|^2 + \|U_k - \mathcal{U}\|^2 &= \|M^k - \mathcal{M}\|^2 \leq c\|M^{k-1} - \mathcal{M}\|^2 \\ &= c(\|L^{k-1} - \mathcal{L}\|^2 + \|U^{k-1} - \mathcal{U}\|^2) \leq 2c\alpha\end{aligned}$$

from the induction hypothesis. This yields for (5.17)

$$\|L_k U_k - A\| \leq (2\alpha c + \kappa)2c\alpha =: \tilde{\alpha}.$$

With the assumption (5.14) on α we obtain from the Banach perturbation lemma

$$\|(L_k U_k)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\tilde{\alpha}}.$$

Inserting these results in (5.16) and taking norms yields

$$\|e_{k+1}\| \leq \frac{\|A^{-1}\|\tilde{\alpha}}{1 - \|A^{-1}\|\tilde{\alpha}}\|e_k\| =: \tilde{c}\|e_k\|$$

with $\tilde{c} \in (0, 1)$. This proves (5.15). ■

6. Numerical Experiment

6.1 Implementation

The starting point of the above developed algorithm is an incomplete factorization of A . So, the matrices L , U and an initial approximation x_0 of the solution x should be given. In the first step of the iteration loop the residual vector is calculated

$$r_k := b - Ax_k$$

Next, the linear system

$$L_k U_k s_k = r_k$$

is solved with one forward and one backward substitution and the solution is updated.

$$x_{k+1} := x_k + s_k \tag{6.18}$$

Now, the matrices L and U are modified according to (4.26). This is done in a loop over the rows i . To this end, we need a storage vector w and set

$$w := s_k$$

If $i > 1$ the vector $w^{(i)}$ in (4.26) is given by

$$\begin{aligned} & \text{sum} = 0 \\ & \text{for } k = i-1:n \\ & \quad \text{sum} = \text{sum} + U(i-1,k) * s(k) \\ & \text{end} \\ & w(i-1) = \text{sum} \end{aligned} \tag{6.19}$$

With this vector, the difference $(y_i - R_i w^{(i)})$ is

```

corr=0
for k=1:i-1
    corr=corr+L(i,k)*w(k)
end
for k=i:n
    corr=corr+U(i,k)*w(k)
end
corr=y(i)-corr

```

(6.20)

After imposing the sparsity pattern of row i of A the scalar product in (4.26) can be calculated and we set $corr = corr/scalp$ or $corr = 0$ according to the value of the scalar product $scalp$. Finally the i -th row of L and U is modified with respect to (4.26).

```

for k=1:i-1
    L(i,k)=L(i,k)+corr*w(k)
end
for k=i:n
    U(i,k)=U(i,k)+corr*w(k)
end

```

In many program packages the sparse matrix A is described by an array of pointers to the non-zero elements of A . The loops (6.19) and (6.20) have to be changed accordingly. This was not done here for clearness of presentation.

6.2 Example

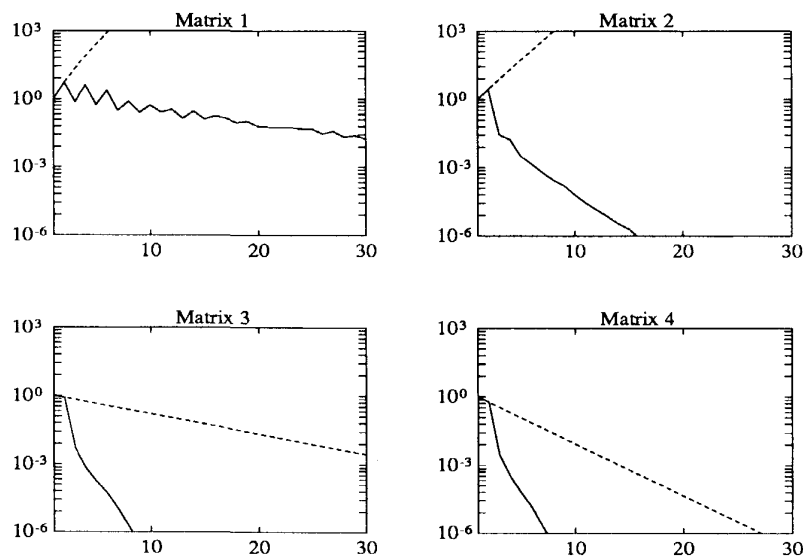
The algorithm was tested with a 100×100 regular matrix generated by random. The elements range from -1 to 1 and a symmetric sparsity structure was imposed also by random. The resulting matrix had 3906 non-zero elements and a condition number of approximately 3000. The incomplete factorization with Richardson iteration is known to work well in the case of symmetric diagonal dominant matrices. This iteration is algorithm (3.2) since both L and U are fixed. Since we want to compare this to algorithms, the matrix A was successively altered. The new matrices were obtained by adding 1, 3, 5 and 9 to all

diagonal elements of A . Then a new incomplete factorization was made and the two algorithms were applied again. The properties of the resulting matrices are given in table 1. There NDD denotes the number of lines which were not diagonal dominant. The greatest difference between the diagonal element and the sum of all other elements in this row was 9.3. The abbreviation $DIFF$ means the difference between the matrix A and the product LU in the matrix norm.

Matrix	NDD	DIFF	with update	without update
1	100	18	85	-
2	95	8.0	16	-
3	69	3.5	9	128
4	35	2.0	6	13
5	3	1.3	5	9

Table 6.1: Results of numerical example

The next two columns show the number of iterations needed to achieve a relative error of 10^{-6} for algorithm (4.26) and the Richardson iteration. The corresponding iteration history is shown in the subsequent figures. There, the number of iterations is plotted against the norm of the error.



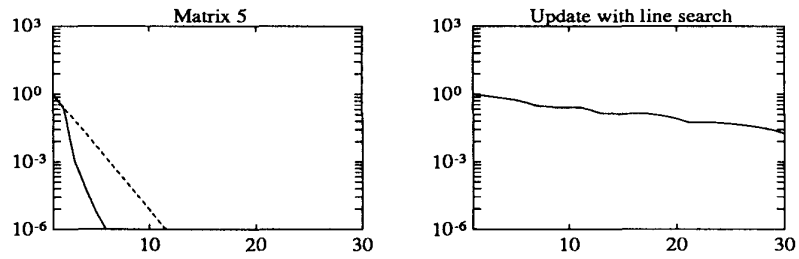


Figure 6.1: Convergence history for different matrices

It is possible to avoid the oscillations in the iteration history for the worst matrix 1 which was the starting matrix A if one uses the update (3.2) with the line search parameter proposed by [1]. The value of t_k is obtained by

$$t_k := \frac{s_k^* s_k}{s_k^* z} \quad \text{with} \quad LUz = As_k.$$

The result is shown in the last figure. Unfortunately, there is no sound theoretical basis for this line search for the algorithm presented here.

7. Conclusion

Starting from the assumption, that an incomplete factorization of a given matrix A is possible, a new algorithm is developed. It maintains the given sparsity structure and don't takes positive definiteness into account. This makes the algorithm appropriate as a solver for non-symmetric linear problems and as a smoother in multi-grid methods. The convergence analysis presented here seems to be a basis for further theoretical investigations of incomplete factorizations for non-symmetric matrices. With our approach, the applicability of the incomplete factorization method and its variants is spread to a wider class of problems, e.g. convection diffusion problems or problems arising in computational fluid mechanics. Especially the numerical results are convincing. This impression is backed not only by the given example, but also by further more extensive tests which will be published. Further testing and theoretical investigations should be done.

Acknowledgement I want to thank Prof. P. Deuffhard, who directed my attention to the sparse secant updates. During this work he encouraged me in several useful discussions.

References

- [1] P. Deuffhard, R. Freund, A. Walter: *Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems*. Preprint SC 90-5, Konrad-Zuse-Zentrum Berlin (1990) and IMPACT of Computing in Science and Engineering to appear.
- [2] J. M. Martinez: *A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations*. SIAM J. Numer. Anal., **27**, pp. 1034-1049 (1990).
- [3] J. E. Dennis and E. S. Marwil: *Direct secant updates of matrix factorizations*. Math. Comp. , **38**, pp. 549-560.
- [4] L. K. Schubert: *Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian*. Math. Comp. , **24**, pp. 27-30 (1970).
- [5] G. W. Johnson and N. H. Austria: *A quasi-Newton method employing direct secant updates of matrix factorizations*. SIAM J. Numer. Anal., **20**, pp. 315-325 (1983).
- [6] 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).
- [7] J. E. Dennis and R. B. Schnabel: *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Englewood Cliffs, NJ. (1983).
- [8] J. A. Meijerink, H. A. Van der Vorst: *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*. Math. Comp. , **31**, pp. 148-162 (1977).
- [9] O. Axelsson: *A survey of preconditioned iterative methods for linear systems of algebraic equations*. BIT , **25**, pp. 166-187 (1985).
- [10] G. Wittum: *On the robustness of ILU-smoothing*. SIAM J. Sci. Stat. Comput. , **10**, pp. 699-717 (1989).
- [11] I. D. L. Bogle and J. D. Perkins: *A new sparsity preserving quasi-Newton update for solving nonlinear equations*. SIAM J. Sci. Stat. Comput. , **11**, pp. 621-630 (1990).

Veröffentlichungen des Konrad-Zuse-Zentrum für Informationstechnik Berlin
Preprints
November 1990

- SC 86-1. P. Deuffhard; U. Nowak. *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems.* (vergriffen) In: Ber. Bunsenges. Phys. Chem., vol. 90, 1986, 940-946
- SC 86-2. H. Melenk; W. Neun. *Portable Standard LISP for CRAY X-MP Computers.*
- SC 87-1. J. Anderson; W. Galway; R. Kessler; H. Melenk; W. Neun. *The Implementation and Optimization of Portable Standard LISP for the CRAY.*
- SC 87-2. R. E. Bank; T. F. Dupont; H. Yserentant. *The Hierarchical Basis Multigrid Method.* (vergriffen) In: Numerische Mathematik, 52, 1988, 427-458.
- SC 87-3. P. Deuffhard. *Uniqueness Theorems for Stiff ODE Initial Value Problems.*
- SC 87-4. R. Buhtz. *CGM-Concepts and their Realizations.*
- SC 87-5. P. Deuffhard. *A Note on Extrapolation Methods for Second Order ODE Systems.*
- SC 87-6. H. Yserentant. *Preconditioning Indefinite Discretization Matrices.*
- SC 88-1. W. Neun; H. Melenk. *Implementation of the LISP-Arbitrary Precision Arithmetic for a Vector Processor.*
- SC 88-2. H. Melenk; H. M. Möller; W. Neun. *On Gröbner Bases Computation on a Supercomputer Using REDUCE.* (vergriffen)
- SC 88-3. J. C. Alexander; B. Fiedler. *Global Decoupling of Coupled Symmetric Oscillators.*
- SC 88-4. H. Melenk; W. Neun. *Parallel Polynomial Operations in the Buchberger Algorithm.*
- SC 88-5. P. Deuffhard; P. Leinen; H. Yserentant. *Concepts of an Adaptive Hierarchical Finite Element Code.*
- SC 88-6. P. Deuffhard; M. Wulkow. *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable.* (vergriffen) In: IMPACT, 1, 1989, 269-301.
- SC 88-7. H. Melenk; H. M. Möller; W. Neun. *Symbolic Solution of Large Stationary Chemical Kinetics Problems.*
- SC 88-8. R. H. W. Hoppe; R. Kornhuber. *Multi-Grid Solution of Two Coupled Stefan Equations Arising in Induction Heating of Large Steel Slabs.*
- SC 88-9. R. Kornhuber; R. Roitzsch. *Adaptive Finite-Element-Methoden für konvektions-dominierte Randwertprobleme bei partiellen Differentialgleichungen.*
- SC 88-10. S.-N. Chow; B. Deng; B. Fiedler. *Homoclinic Bifurcation at Resonant Eigenvalues.*
- SC 89-1. H. Zha. *A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets.*
- SC 89-2. H. Zha. *Restricted Singular Value Decomposition of Matrix Triplets.*
- SC 89-3. W. Huamo. *On the Possible Accuracy of TVD Schemes.*
- SC 89-4. H. M. Möller. *Multivariate Rational Interpolation: Reconstruction of Rational Functions.*
- SC 89-5. R. Kornhuber; R. Roitzsch. *On Adaptive Grid Refinement in the Presence of Internal or Boundary Layers.*
- SC 89-6. W. Huamo; Y. Shuli. *MmB-A New Class of Accurate High Resolution Schemes for Conservation Laws in Two Dimensions.*
- SC 89-7. U. Budde; M. Wulkow. *Computation of Molecular Weight Distributions for Free Radical Polymerization Systems.*
- SC 89-8. G. Maierhöfer. *Ein paralleler adaptiver Algorithmus für die numerische Integration.*
- SC 89-9. H. Yserentant. *Two Preconditioners Based on the Multi-Level Splitting of Finite Element Spaces.*
- SC 89-10. R. H. W. Hoppe. *Numerical Solution of Multicomponent Alloy Solidification by Multi-Grid Techniques.*
- SC 90-1. M. Wulkow; P. Deuffhard. *Towards an Efficient Computational Treatment of Heterogeneous Polymer Reactions.*
- SC 90-2. P. Deuffhard. *Global Inexact Newton Methods for Very Large Scale Nonlinear Problems.*
- SC 90-3. K. Gatermann. *Symbolic solution of polynomial equation systems with symmetry.*
- SC 90-4. F. A. Bornemann. *An Adaptive Multilevel Approach to Parabolic Equations I. General Theory & 1D-Implementation.*
- SC 90-5. P. Deuffhard; R. Freund; A. Walter. *Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems.*
- SC 90-6. D. Wang. *On Symplectic Difference Schemes for Hamiltonian Systems.*
- SC 90-7. P. Deuffhard; U. Nowak; M. Wulkow. *Recent Developments in Chemical Computing.*
- SC 90-8. C. Chevalier; H. Melenk; J. Warnatz. *Automatic Generation of Reaction Mechanisms for Description of Oxidation of Higher Hydrocarbons.*
- SC 90-9. P. Deuffhard; F. A. Potra. *Asymptotic Mesh Independence of Newton-Galerkin Methods via a Refined Mysovskii Theorem.*
- SC 90-10. R. Kornhuber; R. Roitzsch. *Self Adaptive FEM Simulation of Reverse Biased p-n Junctions.*
- SC 90-11. K. Gatermann; A. Hohmann. *Symbolic Exploitation of Symmetry in Numerical Pathfollowing.*
- SC 90-12. A. Walter. *Improvement of Incomplete Factorizations by a Sparse Secant Method.*
- SC 90-13. F. A. Bornemann. *An Adaptive Multilevel Approach to Parabolic Equations II.*

Veröffentlichungen des Konrad-Zuse-Zentrum für Informationstechnik Berlin
Technical Reports November 1990

TR 86-1. H. J. Schuster. *Tätigkeitsbericht (vergriffen)*

TR 87-1. H. Busch; U. Pöhle; W. Stech. *CRAY-Handbuch. - Einführung in die Benutzung der CRAY.*

TR 87-2. H. Melenk; W. Neun. *Portable Standard LISP Implementation for CRAY X-MP Computers. Release of PSL 3.4 for COS.*

TR 87-3. H. Melenk; W. Neun. *Portable Common LISP Subset Implementation for CRAY X-MP Computers.*

TR 87-4. H. Melenk; W. Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running COS Version 3.3*

TR 87-5. H. Melenk; W. Neun. *REDUCE Users Guide for the CRAY 1 / X-MP Series Running COS. Version 3.3*

TR 87-6. R. Buhtz; J. Langendorf; O. Paetsch; D. A. Buhtz. *ZUGRIFF - Eine vereinheitlichte Datenspezifikation für graphische Darstellungen und ihre graphische Aufbereitung.*

TR 87-7. J. Langendorf; O. Paetsch. *GRAZIL (Graphical ZIB Language).*

TR 88-1. R. Buhtz; D. A. Buhtz. *TDLG 3.1 - Ein interaktives Programm zur Darstellung dreidimensionaler Modelle auf Rastergraphikgeräten.*

TR 88-2. H. Melenk; W. Neun. *REDUCE User's Guide for the CRAY 1 / CRAY X-MP Series Running UNICOS. Version 3.3.*

TR 88-3. H. Melenk; W. Neun. *REDUCE Installation Guide for CRAY 1 / CRAY X-MP Systems Running UNICOS. Version 3.3.*

TR 88-4. D. A. Buhtz; J. Langendorf; O. Paetsch. *GRAZIL-3D. Ein graphisches Anwendungsprogramm zur Darstellung von Kurven- und Funktionsverläufen im räumlichen Koordinatensystem.*

TR 88-5. G. Maierhöfer; G. Skorobohatyj. *Parallel-TRAPEX. Ein paralleler, adaptiver Algorithmus zur numerischen Integration ; seine Implementierung für SUPRENUM-artige Architekturen mit SUSI.*

TR 89-1. *CRAY-HANDBUCH. Einführung in die Benutzung der CRAY X-MP unter UNICOS.*

TR 89-2. P. Deuffhard. *Numerik von Anfangswertmethoden für gewöhnliche Differentialgleichungen.*

TR 89-3. A. R. Walter. *Ein Finite-Element-Verfahren zur numerischen Lösung von Erhaltungsgleichungen.*

TR 89-4. R. Roitzsch. *KASKADE User's Manual.*

TR 89-5. R. Roitzsch. *KASKADE Programmer's Manual.*

TR 89-6. H. Melenk; W. Neun. *Implementation of Portable Standard LISP for the SPARC Processor.*

TR 89-7. F. A. Bornemann. *Adaptive multilevel discretization in time and space for parabolic partial differential equations.*

TR 89-8. G. Maierhöfer; G. Skorobohatyj. *Implementierung des parallelen TRAPEX auf Transputern.*

TR 90-1. K. Gatermann. *Gruppentheoretische Konstruktion von symmetrischen Kubaturformeln.*

TR 90-2. G. Maierhöfer; G. Skorobohatyj. *Implementierung von parallelen Versionen der Gleichungslöser EULEX und EULSIM auf Transputern.*

TR 90-3. *CRAY-Handbuch. Einführung in die Benutzung der CRAY X-MP unter UNICOS 5.1*

TR 90-4. H.-C. Hege. *Datenabhängigkeitsanalyse und Programmtransformationen auf CRAY-Rechnern mit dem Fortran-Präprozessor fpp.*

TR 90-5. M. Grammel; G. Maierhöfer; G. Skorobohatyj. *Trapex in POOL; Implementierung eines numerischen Algorithmus in einer parallelen objektorientierten Sprache.*

TR 90-6. P. Deuffhard; A. Hohmann. *Einführung in die Numerische Mathematik.*

TR 90-7. P. Deuffhard. *Zuses Werk weiterdenken.*

