

# A COMPARATIVE STUDY OF CHEBYSHEV ACCELERATION AND RESIDUE SMOOTHING IN THE SOLUTION OF NONLINEAR ELLIPTIC DIFFERENCE EQUATIONS

P.J. VAN DER HOUWEN & B.P. SOMMEIJER  
Centre for Mathematics and Computer Science  
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

G. PONTRELLI(\*)  
Istituto per le applicazioni del calcolo  
Viale del Policlinico 137, Rome, Italy 00161

**Abstract.** We compare the traditional and widely-used Chebyshev acceleration method with an acceleration technique based on residue smoothing. Both acceleration methods can be applied to a variety of function iteration methods and allow therefore a fair comparison. The effect of residue smoothing is that the spectral radius of the Jacobian matrix associated with the system of equations can be reduced substantially, so that the eigenvalues of the iteration matrix of the iteration method used are considerably decreased. Comparative experiments clearly indicate that residue smoothing is superior to Chebyshev acceleration. For a model problem we show that the rate of convergence of the smoothed Jacobi process is comparable with that of ADI methods.

The smoothing matrices by which the residue smoothing is achieved, allow for a very efficient implementation, thus hardly increasing the computational effort of the iteration process. Another feature of residue smoothing is that it is directly applicable to nonlinear problems without affecting the algorithmic complexity. Moreover, the simplicity of the method offers excellent prospects for execution on vector and parallel computers.

## 1. FUNCTION ITERATION METHODS

In [2] a Jacobi-type iteration method for solving nonlinear elliptic difference equations  $f(\mathbf{u})=0$  is described which is essentially based on function evaluation without requiring the solution of linear systems during the successive iterations. The

---

(\*) The investigations were supported by the National Research Council (CNR) of Italy.

function values to be evaluated are smoothed residue values  $Sf$ , where  $S$  is a smoothing matrix. This function iteration method (smoothed Jacobi iteration method) is extremely simple to implement on a computer and highly vectorizable on vector computers. The numerical experiments reported in [2] show that smoothed Jacobi iteration is many times faster than conventional Jacobi iteration, indicating that it may be a competitor to other, more sophisticated, function iteration methods for solving nonlinear elliptic difference equations. It is the purpose of this paper to show that smoothed Jacobi iteration is really faster than function iteration methods with a comparable algorithmic complexity. As a reference method we chose the Chebyshev acceleration method applied to Jacobi iteration with automatic estimation of the dominant eigenvalue in order to provide the eigenvalue interval of the Jacobian matrix  $\partial f/\partial u$  needed by the method. Like smoothed Jacobi iteration, Chebyshev-accelerated Jacobi iteration vectorizes well on vector computers. However, its implementation is more complicated and it turns out that smoothed Jacobi iteration is much faster both for linear and nonlinear problems.

We have tried to accelerate Chebyshev-accelerated Jacobi iteration by applying a technique for eliminating dominant eigenvectors from the iteration error. Since the dominant eigenvalue is automatically determined, such an elimination technique does not complicate the method further. Although we found a reduction of the number of iterations compared with the Chebyshev method without elimination, smoothed Jacobi iteration is still markedly faster.

We then tried to improve the smoothed Jacobi iteration method by applying Chebyshev acceleration, to obtain Chebyshev-accelerated smoothed Jacobi iteration. The results were disappointing. The generally small reduction of the number of iterations does not justify the increased implementational complexity.

Finally, we investigated whether it pays to replace Jacobi iteration by SSOR iteration, to obtain smoothed SSOR iteration. This method requires the evaluation of the Jacobian matrix and is therefore not a true function iteration method any more. Consequently, the convergence improvement should be sufficiently large in order to justify the increased complexity of the method. We found that, when compared with the smoothed Jacobi method, the smoothed SSOR iteration (provided with optimal relaxation parameters) is slightly faster; however, the price to be paid seems not worth the additional effort, and we refrained from a comparison of the Chebyshev-accelerated SSOR and the smoothed SSOR methods.

Our conclusion is that smoothed Jacobi iteration is an extremely attractive and efficient method, particularly on vector computers, and that nonlinearities in the system to be solved neither destroy the high rate of convergence, nor increase the algorithmic complexity. We do not claim that this method is faster than, e.g., multigrid methods. However, such methods, even when the underlying relaxation

method is based on function iteration, require considerably more implementational effort and are less vectorizable than smoothed Jacobi iteration. In a forthcoming paper we will report on a performance evaluation of smoothed Jacobi iteration on vector computers.

In the remainder of this section we shall briefly describe Chebyshev acceleration with automatic eigenvalue estimation and elimination of dominant eigenvector components in the iteration error, and the idea of residue smoothing. In the Sections 2 and 3 we illustrate these techniques for a few linear and nonlinear examples.

### 1.1. Chebyshev acceleration

Consider a stationary, linearly convergent one-step iteration method

$$(1.1) \quad \mathbf{u}_{n+1} = \mathbf{F}(\mathbf{u}_n), \quad n \geq 0,$$

where  $\mathbf{F}(\mathbf{u}) = \mathbf{u}$ , for finding the solution  $\mathbf{u}$  of the equation

$$(1.2) \quad \mathbf{f}(\mathbf{u}) = \mathbf{0}.$$

It is explicitly assumed that the iteration function  $\mathbf{F}$  does not depend on  $n$ , and that  $\partial\mathbf{F}/\partial\mathbf{u}$  essentially has a real eigenvalue spectrum. By applying the well-known Chebyshev acceleration method to (1.1) we obtain the two-step semi-iterative method (cf. ,e.g., [1])

$$(1.3) \quad \mathbf{v}_{n+1} = p_n \mathbf{v}_n + q_n \mathbf{F}(\mathbf{v}_n) + r_n \mathbf{v}_{n-1}, \quad n \geq 0,$$

where the coefficients are defined by:

$$(1.4) \quad \begin{aligned} p_0 &= w_0 / (w_0 + w_1), \quad p_n = 2w_0 T_n(w_0+w_1) / T_{n+1}(w_0+w_1), \quad n \geq 1; \\ q_0 &= 1 - p_0, \quad q_n = 2w_1 T_n(w_0+w_1) / T_{n+1}(w_0+w_1), \quad n \geq 1; \\ r_0 &= 0, \quad r_n = 1 - p_n - q_n, \quad n \geq 1. \end{aligned}$$

Here,  $T_n$  denotes the first-kind Chebyshev polynomial of degree  $n$  and

$$(1.5) \quad w_0 := -(b + a) / (b - a), \quad w_1 := 2 / (b - a),$$

where, usually,  $[a, b]$  denotes the eigenvalue interval of the Jacobian matrix  $\partial\mathbf{F}/\partial\mathbf{u}$ . One of the end points, say  $a$ , of the eigenvalue interval corresponds to the spectral radius of  $\partial\mathbf{F}/\partial\mathbf{u}$ , and can be estimated by Gerschgorin's disk theorem. The estimation of  $b$  is discussed in the next subsection.

### 1.2. Dominant eigenvectors

In this subsection, we discuss the estimation of the eigenvalue corresponding to the dominant eigenvector. This dominant eigenvalue will provide us with an estimate of  $b$ . Furthermore, we consider the elimination of the dominant eigenvector in an attempt to speed up the Chebyshev acceleration process.

### 1.2.1. Estimation of dominant eigenvalues

Introducing the iteration error

$$(1.6) \quad \varepsilon_n = \mathbf{v}_n - \mathbf{u},$$

and substitution into (1.3) yields, in first approximation,

$$(1.7) \quad \varepsilon_n \approx P_n(\partial\mathbf{F}/\partial\mathbf{u}) \varepsilon_0, \quad P_n(x) := T_n(w_0 + w_1x) / T_n(w_0 + w_1),$$

where  $\partial\mathbf{F}/\partial\mathbf{u}$  denotes the Jacobian matrix of  $\mathbf{F}$  evaluated at the solution  $\mathbf{u}$ ; we shall assume that  $\partial\mathbf{F}/\partial\mathbf{u}$  exists in the neighbourhood of  $\mathbf{u}$  and does not vanish, and that the initial approximation is already close enough to the true solution, so that second order terms can be neglected.

Suppose that we choose the interval  $[a,b]$  such that all eigenvalues of  $\partial\mathbf{F}/\partial\mathbf{u}$  are in  $[a,b]$  except for one eigenvalue  $\lambda^* > b$ , with eigenvector  $\mathbf{e}^*$ . Then, for sufficiently large  $n$ , all eigenvector components occurring in the eigenvector expansion of the initial iteration error will be significantly reduced in magnitude, except for the eigenvector  $\mathbf{e}^*$ ; this eigenvector component will dominate the iteration error, i.e.,

$$(1.8) \quad \varepsilon_n \approx P_n(\lambda^*) \mathbf{e}^*.$$

We shall call the above procedure where all eigenvectors but one are reduced in magnitude the *reduction phase* of the iteration method.

From relation (1.8), an estimate for the eigenvalue  $\lambda^*$  can be derived (we shall call  $\lambda^*$  the *dominant eigenvalue*). In the following, division of vectors is always understood to be carried out componentwise.

**Theorem 1.1.** Let  $w_0, w_1$  be defined by (1.5), and let the vector  $\mathbf{R}$  be defined by

$$\mathbf{R} := (w_0 + w_1 + [(w_0 + w_1)^2 - 1]^{1/2}) \Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1},$$

where  $\Delta$  denotes the forward difference operator. Then, a vector of  $\lambda^*$  values is provided by

$$\lambda^* \approx [\mathbf{R}/2 + 1/(2\mathbf{R}) - w_0\mathbf{1}]/w_1,$$

where  $\mathbf{1}$  denotes the unit vector  $(1, 1, \dots, 1)^T$ .

**Proof.** From the definition of Chebyshev polynomials we derive, for  $\lambda^* > b$ ,

$$(1.9) \quad P_n(\lambda^*) \approx [W(\lambda^*)/W(1)]^n, \quad W(x) := w_0 + w_1x + [(w_0 + w_1x)^2 - 1]^{1/2}.$$

By rewriting (1.8) for  $n-1$  and  $n+1$ , forming the expression

$$[\varepsilon_{n+1} - \varepsilon_n] / [\varepsilon_n - \varepsilon_{n-1}],$$

and by using (1.9), we find that  $\lambda^*$  approximately satisfies the relation

$$(1.10) \quad W(\lambda^*) \mathbf{1} = W(1) \Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1} = \mathbf{R}.$$

Solving relation (1.10) for  $\lambda^*$  yields the estimate given in the theorem. ■

When this theorem is applied in actual computation, we obtain as many estimates to the dominant eigenvalue  $\lambda^*$  as there are equations. The spread  $\Delta\lambda^*$  of the interval  $[\lambda^* - \Delta\lambda^*, \lambda^* + \Delta\lambda^*]$  of these estimates can be used as an indication to what extent the iteration error is indeed dominated by  $\mathbf{e}^*$ . The actually used approximation to the dominant eigenvalue might be the arithmetic mean of the available estimates.

### 1.2.2. Elimination of dominant eigenvectors

Having found an approximation to the dominant eigenvalue during the reduction phase of the iteration method, we can proceed with the elimination of the corresponding eigenvector  $\mathbf{e}^*$  from the iteration error (we shall call this process the *elimination phase* of the iterative method). Below, we briefly discuss a few possibilities for eliminating dominant eigenvectors.

One possibility is to apply again the Chebyshev acceleration process (1.3) - (1.4) with the last computed iterate as new initial approximation and with modified values for the parameters  $w_0$  and  $w_1$ .

**Theorem 1.2.** *Let in (1.4) the parameters  $w_0$  and  $w_1$  be defined by*

(1.11)  $w_0 = [a \cos(\pi/(2n)) + \lambda^*] / [a - \lambda^*]$ ,  $w_1 = (\cos(\pi/(2n)) + 1) / (\lambda^* - a)$ ,  $a < \lambda^*$ ,  
*where  $\lambda^*$  is the eigenvalue of  $\partial\mathbf{F}/\partial\mathbf{u}$  corresponding to the eigenvector  $\mathbf{e}^*$  dominating the iteration error  $\boldsymbol{\varepsilon}_0$ .*

(a) *Then the Chebyshev method (1.3) - (1.4) eliminates  $\mathbf{e}^*$  from the iteration error after exactly  $n$  iterations.*

(b) *If the number of iterations is sufficiently large, i.e., if*

(1.12)  $n \geq \pi [2 \arccos([2\lambda^* - a - 1] / [1 - a])]^{-1}$ ,

*then the method is stable in the sense that no eigenvector components of the iteration error are amplified.*

**Proof.** The expressions (1.11) immediately follow from the conditions that the polynomial  $P_n(x)$  should satisfy the relations:

$$P_n(a) = \pm 1/T_n(w_0 + w_1), \quad P_n(\lambda^*) = 0.$$

From these requirements we deduce

$$w_0 + w_1 a = -1, \quad w_0 + w_1 \lambda^* = \cos(\pi/(2n)),$$

resulting in (1.11).

The stability condition (1.12) follows from the requirement  $w_0 + w_1 \geq 1$ . ■

A disadvantage of the above elimination procedure is the computational effort involved by forming a new set of iterates. This leads us to a procedure based on iterates already computed during the reduction phase of the iteration method.

Consider the  $k+1$  iterates  $\mathbf{v}_{n-j+1}$ ,  $j = 0, \dots, k$  computed by (1.3), and define

$$(1.13) \quad \mathbf{v}^* = Q(E) \mathbf{v}_{n-k+1},$$

where  $E$  is the forward shift operator and  $Q$  is a polynomial of degree  $k$  satisfying the condition  $Q(1) = 1$ . For this  $k$ -step extrapolation formula the following theorem holds:

**Theorem 1.3.** *Let  $A$  be the matrix defined by  $A := \text{diag}(\Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1})$ , and define  $\boldsymbol{\varepsilon}^* := \mathbf{v}^* - \mathbf{u}$ ; then*

$$\|\boldsymbol{\varepsilon}^*\|_2 \leq \rho(Q(A)) \|\boldsymbol{\varepsilon}_{n-k+1}\|_2,$$

where  $\|\cdot\|_2$  denotes the spectral norm and  $\rho$  the spectral radius.

**Proof.** Using  $Q(1)=1$  we find from (1.13) that

$$(1.14) \quad \boldsymbol{\varepsilon}^* = \mathbf{v}^* - \mathbf{u} = Q(E) \mathbf{v}_{n-k+1} - \mathbf{u} = Q(E) \mathbf{v}_{n-k+1} - Q(E) \mathbf{u} = Q(E) \boldsymbol{\varepsilon}_{n-k+1},$$

and using (1.7),

$$(1.15) \quad \boldsymbol{\varepsilon}^* \approx Q(E) P_{n-k+1}(\partial F / \partial \mathbf{u}) \boldsymbol{\varepsilon}_0.$$

If  $\mathbf{e}^*$  dominates the iteration error, then it follows from (1.8) that

$$\boldsymbol{\varepsilon}^* \approx Q(E) P_{n-k+1}(\lambda^*) \mathbf{e}^*,$$

so that, by virtue of (1.9), we obtain

$$\boldsymbol{\varepsilon}^* \approx [W(\lambda^*)/W(1)]^{n-k+1} Q(W(\lambda^*)/W(1)) \mathbf{e}^* \approx P_{n-k+1}(\lambda^*) Q(W(\lambda^*)/W(1)) \mathbf{e}^*.$$

Again using (1.8), and replacing  $W(\lambda^*)/W(1)$  by a diagonal matrix  $A$  with elements defined by the components of the vector  $\Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1}$ , we arrive at the relation

$$\boldsymbol{\varepsilon}^* \approx Q(A) \boldsymbol{\varepsilon}_{n-k+1},$$

The assertion of the theorem is now immediate. ■

This theorem suggests that we should choose  $Q$  such that its magnitude is small in the interval

$$(1.16) \quad [\alpha^* - \Delta\alpha^*, \alpha^* + \Delta\alpha^*] := [\min\{\Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1}\}, \max\{\Delta \mathbf{v}_n / \Delta \mathbf{v}_{n-1}\}].$$

We remark that, if  $k=1$ , then, by requiring  $Q(\alpha^*)=0$ , we obtain the famous one-step extrapolation formula of Lyusternik [4].

It follows from the recursion (1.14) that the extrapolation formula (1.13) is stable if the characteristic polynomial  $x^{k+1} - Q(x)$  has its roots on the unit disk, those on the unit circle being simple. It can be shown that, for formulas based on only a few back iterates and for  $\alpha^*$  close to 1, this requirement is easily violated if we require at the same time that  $Q$  is small in magnitude in the interval (1.16). More stable formulas can be constructed by increasing  $k$ . However, this means that more storage is needed to store the necessary iterates.

In actual computation, instabilities introduced by a possible unstable extrapolation formula (1.13) are usually compensated by "overstability" of the

reduction phase. To be more precise, we consider the eigenvalues of the amplification matrix occurring in (1.15), i.e. the matrix

$$Q(E)P_{n-k+1}(\partial F/\partial \mathbf{u}) = Q(E)[T_{n-k+1}(w_0 + w_1 \partial F/\partial \mathbf{u})/T_{n-k+1}(w_0 + w_1)].$$

Let  $Q^*$  be the polynomial obtained from  $Q$  by replacing all the coefficients of  $Q$  by their absolute values. Then, all eigenvalues  $\alpha$  of this amplification matrix corresponding to eigenvalues of  $\partial F/\partial \mathbf{u}$  lying in  $[a, b]$  satisfy the inequality

$$|\alpha| \leq Q^*(E)[1/T_{n-k+1}(w_0 + w_1)] \approx 2 w^{-(n-k+1)} Q^*(1/w), \quad w := W(1),$$

where  $W(1)$  is defined as in (1.9). From this inequality we deduce that the eigenvector components of the initial error are certainly not amplified at the end of the reduction/elimination phase if

$$(1.17) \quad n \geq k - 1 + \log_w(2 Q^*(1/w)).$$

The following theorem presents the lower bound on  $n$  obtained when this result is applied to the case where  $Q$  has all its zeros at  $\alpha^*$ .

**Theorem 1.4.** *Let  $Q$  be given by*

$$Q(x) = [(x - \alpha^*)/(1 - \alpha^*)]^k,$$

where  $\alpha^*$  is defined in (1.16) and is assumed to be less than 1. Then, at the end of the reduction/elimination phase, no eigenvector components of the initial error are amplified if

$$n \geq k \log_w((1 + w|\alpha^*|)/(1 - \alpha^*)) + \log_w(2) - 1, \quad w := W(1),$$

where  $W(1)$  is defined in (1.9).

**Proof.** It follows from the definition of  $Q$  that

$$Q^*(x) = [(x + |\alpha^*|)/(1 - \alpha^*)]^k.$$

Substitution into (1.17) yields the lower bound on  $n$  stated in the theorem. ■

In our experiments, we employed two-step extrapolation formulas because we need already three iterates for estimating the dominant eigenvalue.

### 1.3. Residue smoothing

In [2] iteration methods employing residue smoothing have been analysed for solving nonlinear elliptic systems  $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ . Here, we apply the same technique to a more general class of iteration methods. Given a difference matrix  $D$ , a set of nonnegative integers  $r \in \mathbb{R}$ , a set of relaxation parameters  $\omega_r$ , and some basic iteration method with iteration function

$$\mathbf{G}(\mathbf{u}) := \mathbf{u} + M \mathbf{f}(\mathbf{u}),$$

where  $M$  is the characterizing matrix. Then, we define the class of smoothed iteration methods

$$(1.18a) \quad \mathbf{u}_{n+1} = \mathbf{F}(\mathbf{u}_n), \quad n \geq 0; \quad \mathbf{F} := \prod_{r \in \mathbb{R}} \mathbf{G}_r,$$

$$(1.18b) \quad \mathbf{G}_r(\mathbf{u}) := \mathbf{u} + \omega_r \mathbf{M} \mathbf{S}_r \mathbf{f}(\mathbf{u}),$$

$$(1.18c) \quad \mathbf{S}_r := 4^{-r} [\mathbf{T}_{2r}(\mathbf{I} + 2\mathbf{D}) - \mathbf{I}] (2\mathbf{D})^{-1}, \quad \mathbf{T}_m(x) := \cos(m \arccos(x)),$$

where  $\mathbf{S}_r$  is the so-called *smoothing matrix* (notice that  $\mathbf{S}_r$  reduces to the identity matrix if  $r=0$ ).

Thus, the iteration formula (1.18) may be interpreted as a cycle of smoothed basic iteration steps.

### 1.3.1. The smoothing matrix

The smoothing matrix  $\mathbf{S}_r$  is a *polynomial* in  $\mathbf{D}$  and completely defined as soon as  $\mathbf{D}$  is specified. We remark that  $\mathbf{D}$  is allowed to be a singular matrix because, in spite of our notation, its inverse does not need to exist. In order to simplify the notation we shall continue to write  $\mathbf{D}^{-1}$  in the various formulas without actually requiring that it exists.

The matrix  $\mathbf{D}$  is a difference matrix the eigenvalues of which are assumed to be in the interval  $[-1, 0]$ . As a consequence, the eigenvalues of the smoothing matrices are in the interval  $[0, 1]$ . In one-dimensional problems (two-point boundary value problems), one may consider the difference matrix

$$(1.19) \quad \mathbf{D} := (1/4) \begin{pmatrix} 0 & & & & & \\ 1 & -2 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ & & & & & 0 \end{pmatrix}$$

This matrix generates smoothing matrices which leave the first and last component of the vector to which they are applied unchanged. Therefore, they are suitable in cases where the first and last component of the vector to be smoothed should not change. For example, we mention the case of a residue vector of which the first and last component vanish, that is, the case where the first and last equation of the system  $\mathbf{f}(\mathbf{u}) = \mathbf{0}$  represent *Dirichlet boundary conditions* (we observe that in such situations the first element of the first row and the last element of the last row in  $\mathbf{D}$  may be replaced by any value, so that  $\mathbf{D}$  becomes nonsingular).

In actual computation, it is generally not feasible to precompute the smoothing matrix because of storage requirements. On the other hand, in the case where  $\mathbf{D}$  is defined by (1.19), the matrix  $\mathbf{S}_r$  exhibits a regular pattern which can be exploited for an efficient implementation. For example, the first three smoothing matrices are respectively given by  $\mathbf{S}_0 = \mathbf{I}$ ,



$$S_1 = (1/4) \begin{pmatrix} 4 & & & & \\ & 1 & 2 & 1 & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & & 1 & 2 & 1 \\ & & & & & & 4 \end{pmatrix}$$

and

$$S_2 = (1/16) \begin{pmatrix} 16 & & & & & & \\ & 9 & 2 & 2 & 2 & 1 & \\ & & 4 & 2 & 4 & 3 & 2 & 1 \\ & & & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ & & & & 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ & & & & & 0 & 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

These examples suggest the precomputation of the first few rows of the smoothing matrices until the pattern becomes regular. Alternatively, we can generate the smoothing matrices by exploiting factorization properties of Chebyshev polynomials. This approach seems to be more attractive. By using the identity

$$T_{2r+1}(z) \equiv T_2(T_{2r}(z)),$$

we derive from (1.18c) the recursion

$$S_{r+1} = (I + 4^r D S_r) S_r.$$

By writing  $S_{r+1} = F_{r+1} S_r$  we arrive at the following theorem which expresses the smoothing matrix  $S_r$  as a product of  $r$  factor matrices  $F_j$ :

**Theorem 1.5.** *Let  $D$  be any difference matrix and define the factor matrices  $F_1 := I + D$ ,  $F_{j+1} := (I - 2F_j)^2$ ,  $j \geq 1$ . Then  $S_r = F_1 \cdot F_2 \cdot \dots \cdot F_r$ . ■*

From this theorem we conclude that, if the factor matrices are precomputed, then the smoothing matrix can be generated by  $r$  matrix-vector multiplications. For a number of difference matrices  $D$  and for  $r$  not too large (in a typical case  $r$  should not exceed  $\log_2(1/\Delta x)$  where  $\Delta x$  is the mesh size), it turns out that the corresponding factor matrices are almost as sparse as  $D$  itself, so that the application of the smoothing matrix involves only  $r$  matrix-vector multiplications with matrices of similar complexity as  $D$ . For larger values of  $r$  one may proceed as follows. Let  $S_q$  be the smoothing matrix which can 'conveniently' be generated by means of Theorem 1.5. Then, by employing the factorization formula

$$T_{2q+j}(z) \equiv T_{2j}(T_{2q}(z)),$$

we find that

$$S_{q+j} := 4^{-q-j} [T_{2j}(I+2^{2q+1}D S_q) - I] (2D)^{-1}.$$

This formula expresses  $S_{q+j}$  in terms of a polynomial of degree at most  $2j$  of  $D$  and  $S_q$ . In this way it is in principle possible to generate smoothing matrices of higher indices.

The precomputation of the factor matrices is relatively easy in the case of *one-dimensional* problems and can often be done by hand. In higher dimensional elliptic problems with irregular geometries, this is less attractive. However, by considering the problem as a system of coupled two-point boundary-value problems, one may apply one-dimensional smoothing operators (e.g., based on (1.19)) to the successive problems (cf. [2]).

### 1.3.2. The matrix $M$

By means of the matrix  $M$ , several basic iteration methods can be selected. Let the matrix  $\partial f/\partial u$  be split according to

$$\partial f/\partial u = C + L + U \quad \text{or} \quad \partial f/\partial u = H + V,$$

where  $C$  is diagonal,  $L$  and  $U$  are lower and upper triangular, and  $H, V$  correspond to the ADI splitting of  $A$  (or any other 'convenient' splitting). In terms of these splitting matrices, various matrices  $M$  may be defined. In Table 1.1, a few examples are listed. Here,  $\omega$ ,  $\omega_H$  and  $\omega_V$  are parameters which depend on the spectrum of  $\partial f/\partial u$  (cf. [1]).

**Table 1.1.** Possible  $M$  matrices.

Jacobi-type:	$M :=$ diagonal, e.g., $M = -C^{-1}$
Gauss-Seidel:	$M := -(C + L)^{-1}$
SOR:	$M := -(C/\omega + L)^{-1}$
SSOR:	$M := -(2/\omega - 1) (C/\omega + U)^{-1} C (C/\omega + L)^{-1}$
ADI:	$M := -(\omega_H + \omega_V) (V + \omega_V D)^{-1} (H + \omega_H D)^{-1}$

We observe that instead of solving  $f(u) = 0$ , we can alternatively solve the preconditioned system  $Pf(u) = 0$ . If we replace the matrix  $MS_r$  by  $MS_r P$ , then the resulting iteration method is given by (1.18). In particular, we may set  $M = I$  and  $P$  equal to one of the matrices specified above.

### 1.3.3. The model situation

The choice of the relaxation parameters in (1.18) will be based on the model case where  $D$  equals the difference matrix defined by the normalized Jacobian of  $f$ :

$$(1.20) \quad D := \rho^{-1} \partial f / \partial \mathbf{u},$$

with  $\rho$  denoting the spectral radius of  $\partial f / \partial \mathbf{u}$ . However, we emphasize that the matrix  $D$  actually used in practice is a very rough approximation to this normalized Jacobian; for example, the matrix defined in (1.19) turned out to be rather effective in the case of smoothed Jacobi iteration of Dirichlet problems (cf. [2]). The damping of the iteration error, is largely determined by the Jacobian matrix  $\partial F / \partial \mathbf{u}$  in (1.18):

$$(1.21) \quad \partial F / \partial \mathbf{u} = \prod_r \partial G_r / \partial \mathbf{u} = \prod_r [I + \omega_r M S_r \partial f / \partial \mathbf{u}].$$

Substitution of (1.20) into (1.18c) and the resulting expression for  $S_r$  into (1.21) yields

$$(1.22) \quad \partial F / \partial \mathbf{u} = \prod_r [I + \rho \omega_r 4^{-r} M [T_{2r}(I+2D) - I] / 2].$$

Thus, given the matrices  $M$  and  $D$ , we are faced with the problem of choosing a set of relaxation parameters  $\{\omega_r\}$  such that the eigenvalues of  $\partial F / \partial \mathbf{u}$  are small in magnitude. These eigenvalues will be called *damping factors* of the iteration method. In Section 3 we will derive suitable relaxation parameters for the Jacobi case. The resulting iteration scheme belongs to the class of function iteration methods, which, essentially, only require the evaluation of values of  $f$ . In Section 4 we describe a numerical approach to obtain relaxation parameters for the SSOR method. Formally, these schemes do not belong to the class of function iteration methods, although the amount of linear algebra for these schemes is rather modest. Still further away from this class is the ADI case, which involves the solution of tridiagonal systems. We did not consider this method; an analysis of smoothed ADI iteration may be found in [5].

First, however, for the sake of comparison, we give results obtained by Chebyshev-accelerated Jacobi iteration which still belongs to the most efficient conventional function iteration methods available in the literature for solving elliptic equations.

## 2. CHEBYSHEV ACCELERATION OF JACOBI ITERATION

We shall present numerical experiments with the Chebyshev acceleration method of conventional Jacobi-type iteration with automatic estimation of the dominant eigenvalue. The most simple choice of the matrix  $M$  characterizing a Jacobi-type iteration method is  $M = 2I/\rho$ , where  $\rho$  is the spectral radius of  $\partial f / \partial \mathbf{u}$ .

Alternatively, one may choose  $M = - [\text{diag}(\partial f/\partial \mathbf{u})]^{-1} = - C^{-1}$ . We emphasize that, updating the matrix  $M$  during the iteration may result in an iteration function  $F$  which is  $n$ -dependent, contradicting our assumption that  $F$  is stationary (see Section 1.1). If no Chebyshev acceleration is applied, then, at the cost of some additional computational effort, this strategy may improve the convergence (see Section 3.2). However, if Chebyshev acceleration is used, then the matrix  $M$  should be evaluated in the first step of the reduction phase.

First, we apply the Chebyshev-accelerated Jacobi method without elimination of the dominant eigenvectors. In the next subsection we shall illustrate the effect of the elimination process.

**2.1. Chebyshev acceleration with automatic estimation of the dominant eigenvalue**

The following strategy was applied:

(i) *Initial approximation:* linear interpolation of the boundary values.

(ii) *Chebyshev reduction phase:* application of the Chebyshev acceleration process {(1.3) - (1.5)} where  $F$  is defined by (1.18) with  $\mathbb{R} = \{0\}$ ,  $\omega_0 = 1/2$ ,  $a = 0$ , and where the value of  $b$  occurring in {(1.3) - (1.5)} is such that the dominant eigenvalue  $\lambda^*$  is outside the interval  $[a, b]$  (observe that this choice of  $\mathbb{R}$  results in a conventional iteration method because the only smoothing matrix  $S_r = S_0 = I$ ). In our experiments we chose  $b = 0.95$ .

(iii) *Restart criterion:* restart of the reduction phase with adjusted value of  $b$  as soon as the  $\lambda^*$ -estimates obtained in two successive iterations (cf. Theorem 1.1 and the discussion following this theorem) satisfy the condition  $\|\lambda^*_n - \lambda^*_{n-1}\| < \eta \lambda^*_n$ . The new value of  $b$  is defined by  $b = \lambda^*_n + \delta(1 - \lambda^*_n)$ . The strategy parameters  $\eta$  and  $\delta$  are specified in the tables of results.

(iv) *Stopping criterion:* termination of the iteration process as soon as the residue satisfies the condition  $\|\mathbf{f}(\mathbf{u}_n)\|_\infty \leq 10^{-2} (\Delta x)^2$ , where  $\Delta x$  denotes the mesh size of the grid defining the elliptic difference equations.

Consider the model problem  $u_{xx} = g(x)$  with Dirichlet boundary conditions at  $x=0$  and  $x=1$ , that is the system

$$(2.1) \quad \begin{pmatrix} 1 & & & & & & \\ 1 & -2 & 1 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & -2 & 1 & \\ & & & & & & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \cdot \\ \cdot \\ u_m \\ u_{m+1} \end{pmatrix} = \begin{pmatrix} u(0) \\ (\Delta x^2)g_1 \\ \cdot \\ \cdot \\ (\Delta x^2)g_m \\ u(1) \end{pmatrix}$$

where  $\Delta x := 1/(m+1)$ ,  $g_j := g(j\Delta x)$ , and where  $u(0)$  and  $u(1)$  are prescribed boundary values.

We start with this model problem where

$$(2.2) \quad g(x) = 6x, u(0) = 0 \text{ and } u(1) = 1.$$

For future reference, we first give the results of the conventional Jacobi method for a few values of  $\Delta x$ . The numbers of iterations needed to satisfy the stopping criterion (iv) are given in Table 2.1.

**Table 2.1.** Conventional Jacobi method for the model problem with  $M = (\Delta x)^2 I/2$ .

$\Delta x=1/16$	$\Delta x=1/32$	$\Delta x=1/64$
1190	5342	23675

Next, we apply the Chebyshev acceleration, following the above mentioned strategy. For a few values of  $\Delta x$ ,  $\eta$  and  $\delta$ , the numbers of iterations are listed in Table 2.2. The smallest number on each grid is printed in bold type.

**Table 2.2.** Chebyshev-accelerated Jacobi method for the model problem {(2.1)-(2.2)} with  $M = (\Delta x)^2 I/2$

$\eta$	$\Delta x=1/16$			$\Delta x=1/32$			$\Delta x=1/64$		
	$\delta=0$	$\delta=.1$	$\delta=.25$	$\delta=0$	$\delta=.1$	$\delta=.25$	$\delta=0$	$\delta=.1$	$\delta=.25$
$10^{-2}$	159	148	131	658					
$10^{-3}$	72	67	78	265	242	199	983		
$10^{-4}$	74	<b>66</b>	79	198	160	163	694	643	555
$10^{-5}$		79	66	166	<b>158</b>	175	554	501	<b>388</b>
$10^{-6}$			71	173	161	180	402	394	417

The second example is a nonmodel problem originating from the nonlinear problem

$$(2.3) \quad (\exp(u))_{xx} - 5x^3(4 + 5u)\exp(u) = 0, u(0) = 0, u(1) = 1,$$

the exact solution of which is given by  $u(x)=x^5$ . This problem is discretized on the grid points  $\{j\Delta x\}$  using symmetric differences.

Again, we start with the results obtained by the conventional Jacobi method:

**Table 2.3.** Conventional Jacobi method for problem (2.3) with  $M = (\Delta x)^2 I / (2e)$ .

$\Delta x=1/16$	$\Delta x=1/64$
1925	38117

The effect of the Chebyshev acceleration on this problem is shown in Table 2.4.

**Table 2.4.** Chebyshev-accelerated Jacobi method for problem (2.3) with  $M = (\Delta x)^2 I / (2e)$ .

$\eta$	$\Delta x=1/16$			$\Delta x=1/32$			$\Delta x=1/64$		
	$\delta=0$	$\delta=.1$	$\delta=.25$	$\delta=0$	$\delta=.1$	$\delta=.25$	$\delta=0$	$\delta=.1$	$\delta=.25$
$10^{-2}$	415			1417					
$10^{-3}$	415	392	356	1222					
$10^{-4}$	151	137	<b>98</b>	1235	1169	1064	2917	2763	2513
$10^{-5}$	151	137	<b>98</b>	1235	1169	1064	2417	2289	2081
$10^{-6}$	212	191	213	389	359	<b>278</b>	1177		<b>1001</b>

In performing the above experiments, we observed that the estimates  $\lambda_n^*$  converged from below to the true value. This explains why the best results are obtained by slightly overestimating (i.e., by setting  $\delta > 0$ ) the final estimate. Furthermore, the convergence of  $\lambda_n^*$  happened to be very slow. Therefore, it is not surprising that a rather stringent restart criterion (e.g.  $\eta \in [10^{-6}, 10^{-5}]$ ) results in an optimal performance.

## 2.2. Chebyshev acceleration with automatic elimination of dominant eigenvectors

Instead of the restart criterion (iii) of the preceding subsection we now use:

(iii) *Elimination phase*: application of a three-step elimination formula of the form (1.13) with characteristic polynomial defined according to Theorem 1.4, and restart of the reduction phase (ii) with adjusted value of  $b$ . The elimination formula is applied as soon as (1.17), with  $k=3$ , is satisfied, and if the  $\lambda^*$ -estimates obtained in two successive iterations satisfy the condition  $|\lambda^*_n - \lambda^*_{n-1}| < \eta \lambda^*_n$ . The new value of  $b$  is defined by  $b = \gamma \lambda^*_n$ , where the strategy parameter  $\gamma$  is specified in the tables of results. The analogues of the Tables 2.2 and 2.4 are given below. The numbers in brackets denote the number of times that a dominant eigenvector has been eliminated. A comparison of the results listed in these tables with those of the Tables 2.2 and 2.4 reveals that the reduction of the number of iterations is rather modest and does not seem worth the additional implementational effort.

**Table 2.5.** Chebyshev-accelerated Jacobi method for the model problem {(2.1)-(2.2)} with  $M = (\Delta x)^2 I / 2$ .

$\eta$	$\Delta x=1/16$		$\Delta x=1/32$		$\Delta x=1/64$	
	$\gamma=.95$	$\gamma=.99$	$\gamma=.95$	$\gamma=.99$	$\gamma=.95$	$\gamma=.99$
$10^{-2}$	55(5)	75(3)	141(10)	174(8)	483(23)	438(12)
$10^{-3}$	52(3)	71(3)	135(7)	188(7)	445(21)	<b>435(11)</b>
$10^{-4}$	<b>49(2)</b>	66(2)	212(7)	156(5)	856(21)	456(9)
$10^{-5}$	<b>49(2)</b>	57(1)	192(4)	135(3)	1074(10)	477(7)
$10^{-6}$	62(2)	56(1)	209(3)	<b>129(3)</b>		

**Table 2.6.** Chebyshev-accelerated Jacobi method for problem (2.3) with  $M = (\Delta x)^2 I / (2e)$ .

$\eta$	$\Delta x=1/16$		$\Delta x=1/32$		$\Delta x=1/64$	
	$\gamma=.95$	$\gamma=.99$	$\gamma=.95$	$\gamma=.99$	$\gamma=.95$	$\gamma=.99$
$10^{-2}$	96(6)	124(5)	303(16)	265(8)	<b>611(31)</b>	622(14)
$10^{-3}$	137(6)	121(4)	298(14)	<b>261(8)</b>	656(30)	640(14)
$10^{-4}$	109(4)	100(3)	463(14)	263(6)	1258(31)	673(14)
$10^{-5}$	130(4)	<b>94(2)</b>	590(6)	392(4)	1963(19)	1025(13)
$10^{-6}$	130(4)	183(1)	491(4)	401(3)		

### 3. SMOOTHED JACOBI ITERATION

As we have seen, if  $M$  is chosen to be a *diagonal* matrix, then the basic iteration method is of the Jacobi-type. In the previous section, we selected  $\mathbb{R} = \{0\}$ , resulting in  $S_r = I$ , i.e., an unsmoothed process. In this section we shall exploit the matrix  $S_r$ , that is we consider a *smoothed* Jacobi-type iteration method.

We shall first derive suitable relaxation parameters for the model situation (1.20), and then we shall show, by means of numerical experiments, that these parameters are also effective in nonmodel cases.

#### 3.1. Derivation of relaxation parameters

Let  $M$  be the identity matrix (or any diagonal matrix with constant diagonal entries), and let  $\mu$ ,  $\alpha_r(\mu)$  and  $\alpha(\mu)$  denote the eigenvalues of  $D$ ,  $\partial G_r/\partial \mathbf{u}$  and  $\partial F/\partial \mathbf{u}$ , respectively. An inspection of the zeros and extreme values of the functions  $\alpha_r(\mu)$  reveals that, if the set  $\mathbb{R}$  contains an integer  $r$ , then it should contain the integers  $0, \dots, r-1$ , otherwise  $\alpha(\mu)$  assumes values 1 in the interval  $[-1, 0)$ . This leads us to define the set  $\mathbb{R}$  by successive integers starting with  $r=0$ . From the expression (1.22) we obtain the following theorems:

**Theorem 3.1.** *Let  $D$  and  $\partial f/\partial \mathbf{u}$  be related according to (1.20), let  $\mathbb{R} := \{0, 1, \dots, q\}$ , and let  $\rho \omega_r 4^{-r} M = I$  for all  $r$  in  $\mathbb{R}$ . Then the Jacobian matrix  $\partial F/\partial \mathbf{u}$  of the iteration function in (1.18) and the corresponding damping factors are respectively given by  $S_{q+1}$  and by*

$$\alpha(\mu) := 2^{-(2q+3)} \mu^{-1} [T_{2q+1}(1 + 2\mu) - 1],$$

where  $\mu$  runs through the eigenvalues of  $D$ .

**Proof.** On substitution of  $\mathbb{R}$  and  $\rho \omega_r 4^{-r} M = I$  into (1.22) we obtain

$$\partial F/\partial \mathbf{u} = \prod_r [I + T_{2r}(I + 2D)/2].$$

Using a factorization formula for Chebyshev polynomials of degree  $m = 2p$  (cf. [2]):

$$T_m(z) = 1 - m(1-z) \prod_{j=0}^{p-1} (1 + T_{2j}(z)),$$

we find that  $\partial F/\partial \mathbf{u} = S_{q+1}$  which yields the assertion of the theorem. ■

**Theorem 3.2.** *Let the conditions of Theorem 3.1 be satisfied, let the largest value in the interval  $[-1, 0)$  where the function  $\alpha(\mu)$  assumes a maximum value be denoted by  $\mu_b$ , and let  $\mu_s$  be the largest value in  $[-1, 0)$  where  $\alpha(\mu) = \alpha(\mu_b)$ . Then the following assertions hold :*



(a) If the eigenvalues  $\mu$  of  $D$  satisfy the inequality  $-1 \leq \mu \leq \mu_s$ , then the spectrum of the matrix  $\partial F/\partial \mathbf{u}$  is contained in the interval  $[a,b] := [0, \alpha(\mu_s)]$ .

(b) For all  $q$  we have the approximation  $\mu_s \approx [\cos(\pi/2q) - 1]/3 \approx -4q\pi^2/6$ .

**Proof.** It follows from Theorem 3.1 that the eigenvalues of the matrix  $\partial F/\partial \mathbf{u}$  are given by  $\alpha(\mu)$ , where  $\mu$  runs through the spectrum of  $D$  (see Figure 3.1). From the definition of  $\mu_b$  and  $\mu_s$  it follows that

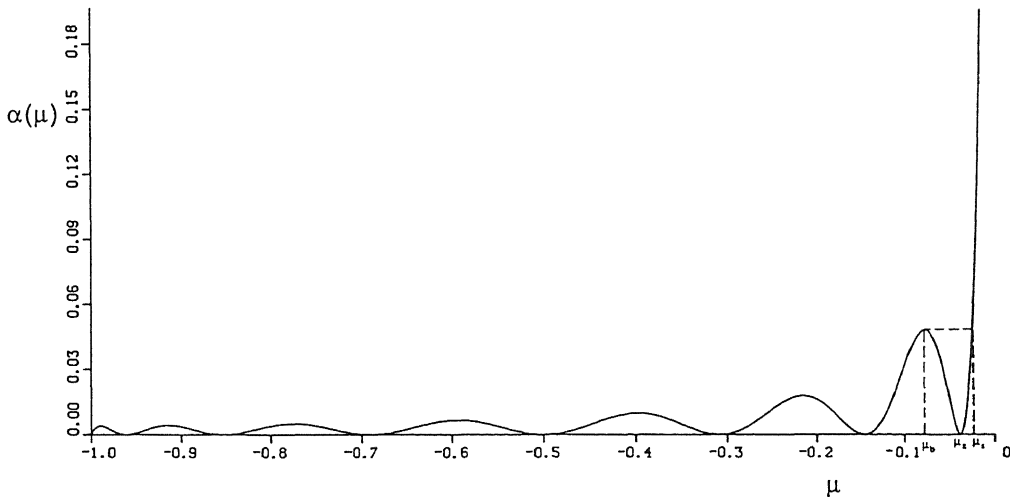
$$0 \leq \alpha(\mu) \leq \alpha(\mu_b) = \alpha(\mu_s) \text{ for all } \mu \in [-1, \mu_s]$$

proving part (a) of the theorem.

A numerical calculation reveals that

$$\mu_s \approx 2 \mu_z/3,$$

with  $\mu_z$  the largest value in the interval  $[-1,0)$  where the function  $\alpha(\mu)$  assumes a zero value. This leads to the approximation given by part (b). ■



**Figure 3.1.** Behaviour of the function  $\alpha(\mu)$  for  $q=3$

Recalling that the eigenvalues of  $\partial F/\partial \mathbf{u}$  are the damping factors of the iteration method, it is of interest to see to what extent the eigenvalue interval  $[a,b] = [0,b]$  of  $\partial F/\partial \mathbf{u}$  is reduced. In Table 3.1 the numerical values of  $b$  for a few values of  $q$  are given. These values show that for  $q \geq 3$  this interval is almost constant and

approximately given by  $[0, .05]$ , provided, of course, that the eigenvalues of  $D$  are less than  $\mu_S$ .

**Table 3.1.** Numerical values of  $\mu_S$  and  $b = \alpha(\mu_S)$

$q =$	1	2	3	4	5	6	7	8
$-\mu_S =$	$1/3$	$.9710^{-1}$	$.2510^{-1}$	$.6410^{-2}$	$.1610^{-2}$	$.4010^{-3}$	$.9910^{-4}$	$.2510^{-4}$
$b =$	$.0741$	$.0525$	$.0485$	$.0475$	$.0473$	$.0472$	$.0472$	$.0472$

The following example, illustrates this result.

**Example 3.1.** Consider the system (2.1) arising from the equation  $u_{xx} = g(x)$  with Dirichlet boundary conditions at  $x=0$  and  $x=1$ . Let the matrix  $D$  be defined by (1.19), then the condition (1.20) is satisfied with  $\rho = 4(\Delta x)^{-2}$ . By virtue of the Dirichlet boundary conditions incorporated in (2.1), we can restrict the space of residue vectors to the subspace of vectors with vanishing first and last component. Let  $D^*$  be the matrix obtained by omitting the first and last row and column of  $D$ . It is easily verified that, in this subspace,  $D$  and  $D^*$  have the same set of eigenvectors and eigenvalues. It is well known that  $D^*$  possesses eigenvalues given by

$$\mu_j^* = -[1 - \cos(j\pi/(m+1))]/2, \quad j = 1, \dots, m,$$

where  $m$  is the order of the matrix  $D^*$ . Thus, the relevant eigenvalues of  $D$  are in the interval  $(-1, -[1 - \cos(\pi/(m+1))]/2] \approx (-1, -\pi^2/(4(m+1)^2)]$ .

A comparison with the bound  $\mu_S$  given in Theorem 3.2 yields the condition

$$m \leq (3 \cdot 4q/2)^{1/2} - 1 = 2q \sqrt{1.5} - 1.$$

On the other hand, in order to preserve a simple structure of the factor matrices  $F_j$  we should require that

$$q \leq \log_2(1/\Delta x) \Rightarrow m \geq 2q - 1.$$

(cf. the discussion of Theorem 1.5). This leads us to the conclusion that smoothed Jacobi iteration has damping factors bounded by  $.05$  if  $q \geq 3$  and if  $m$  satisfies the above inequalities. For future reference, we list the bounds on  $m$  for a few values of  $q$  (cf. Table 3.2). ■

**Table 3.2.** Lower and upper bounds for  $m$  for the model problem (2.1).

$q =$	3	4	5	6	7	8
$m \geq$	7	15	31	63	127	255
$m \leq$	8	18	38	77	155	312

Next we consider the average rate of convergence of smoothed Jacobi for the above model problem, or more generally, for problems which satisfy the conditions of Theorem 3.1 and 3.2 (a). Then the following theorem holds.

**Theorem 3.3.** *Let the conditions of Theorem 3.1 be satisfied, let the eigenvalues  $\mu$  of  $D$  satisfy the inequality  $-1 \leq \mu \leq \mu_s$ , and let  $q \approx \log_2(1/\Delta x)$  as  $\Delta x \rightarrow 0$ , then the average rate of convergence of smoothed Jacobi iteration is given by  $c/\ln(1/\Delta x)$  where  $c \approx 2.1$  as  $\Delta x \rightarrow 0$ .*

**Proof.** It follows from Table 3.1 that per iteration step the average reduction factor for the iteration error is given by  $r = b^{1/(q+1)} \approx .05^{1/(q+1)}$ . Hence, the average rate of convergence is given by  $R = -\ln(r) \approx 3/(q+1)$ , so that for  $q \approx \log_2(1/\Delta x)$  we obtain  $R \approx 2.1/\ln(1/\Delta x)$ .

The condition that  $q$  should be as large as  $\log_2(1/\Delta x)$  without violating the condition  $-1 \leq \mu \leq \mu_s$  can be satisfied in case of the model problem considered in Example 3.1. In the case of two-dimensional model problems, these conditions can also be satisfied provided that we base the smoothing procedure on the successive application of one-dimensional smoothing matrices. In fact, the value of  $b$  will be slightly smaller than  $.05$  resulting in a slightly larger value for  $c$ . For such problems it is of interest to compare the average rate of convergence of smoothed Jacobi with that of ADI methods. For the Peaceman-Rachford version of the ADI method it is known that the average rate of convergence is given by  $R \approx c/\ln(1/\Delta x)$ , where  $c$  is some constant greater than  $.777$ . Thus, we may conclude that smoothed Jacobi has the same order of convergence rate as the ADI method, but is much cheaper per iteration step because of the absence of implicit relations to be solved.

### 3.2. Numerical experiments

In our numerical experiments, we applied Chebyshev acceleration of smoothed Jacobi iteration with prescribed interval  $[a,b]$  according to the following strategy:

(i) *Initial approximation*: linear interpolation of the boundary values.  
 (ii) *Chebyshev reduction phase*: application of the Chebyshev acceleration process  $\{(1.3) - (1.5)\}$  where  $\mathbf{F}$  is defined by (1.18) and (1.19) with  $\omega_r = 2^{2r-1}$  ( $r=0, \dots, q$ ). These  $\omega_r$  parameters give rise to a zero  $a$ -value. The value of  $b$  is specified in the tables of results and  $q$  and  $m$  are chosen as allowed by Table 3.2.

(iii) *Stopping criterion*: termination of the iteration process as soon as the residue satisfies the condition  $\|f(\mathbf{u}_n)\|_\infty \leq 10^{-2} (\Delta x)^2$ , where  $\Delta x$  denotes the mesh size of the grid on which the elliptic difference equations are defined.

As in the preceding section, possible choices of the matrix  $M$  are  $M = 2I/\rho$ , where  $\rho$  is the spectral radius of  $\partial f/\partial \mathbf{u}$ , or  $M = -[\text{diag}(\partial f/\partial \mathbf{u})]^{-1}$ .

Again we start with the model problem defined by (2.1) with  $g(x) = 6x$ ,  $u(0) = 0$  and  $u(1) = 1$ . In Table 3.3 the numbers of iterations needed to satisfy the stopping criterion are listed. It turns out that if smoothing is used, then the exact solution of the system of equations is obtained after just one cycle of smoothed Jacobi iterations, that is, after one single application of the iteration formula (1.18). This means that the value of  $b$  is irrelevant, because the iteration process stops before the Chebyshev recursion gets started. The reason for this peculiar behaviour is that for model problems of the type (2.1) and for the special grids employed in Table 3.3, all damping factors of the smoothed Jacobi method (as specified above) vanish.

**Table 3.3.** Smoothed Jacobi method for the model problem with  $M = (\Delta x)^2 I / 2$ .

$\Delta x=1/16$	$\Delta x=1/32$	$\Delta x=1/64$	$\Delta x=1/128$	$\Delta x=1/256$
5	6	7	8	9

Our second example is the nonmodel problem (2.3). The results of the smoothed Jacobi method without acceleration (first row in Table 3.4) show an impressive reduction of the number of iterations when compared with conventional Jacobi (see Table 2.3). But also the comparison of smoothed Jacobi and Chebyshev-accelerated Jacobi (see the Tables 2.4 and 2.6) clearly shows the superiority of residue

smoothing as an acceleration technique. We then tried to improve smoothed Jacobi further by applying Chebyshev acceleration to the smoothed Jacobi process. Table 3.4 indicates only a modest increase of the rate of convergence, especially in cases where the b-value is not optimal. Therefore, one may decide to forget about Chebyshev acceleration in the case of residue smoothing. This results in one array less for storage and at the same time in an extremely simple algorithm.

**Table 3.4.** Smoothed Jacobi method for problem (2.3) with  $M = (\Delta x)^2 I / (2e)$ .

Chebyshev acceleration	[a,b]	$\Delta x=1/16$	$\Delta x=1/64$	$\Delta x=1/256$
no	--	73	129	196
yes	[0,.75]	78	137	197
yes	[0,.50]	48	88	125
yes	[0,.40]	<b>46</b>	<b>79</b>	<b>117</b>
yes	[0,.30]	55	96	142
yes	[0,.20]	62	108	162

Our next experiment illustrates the effect of tuning the matrix  $M$  to the diagonal of the Jacobian matrix  $\partial f / \partial \mathbf{u}$ . Table 3.5 shows that for problem (2.3) some reduction of the number of iterations is obtained, but it is doubtful whether it is worth the additional effort for computing the diagonal elements. A second observation is that for  $M = - [\partial f(\mathbf{u}_n) / \partial \mathbf{u}]^{-1}$  the Chebyshev acceleration does not improve the convergence, because the iteration function  $\mathbf{F}$  is non-stationary.

**Table 3.5.** Smoothed Jacobi method for problem (2.3) with alternative M matrices.

		$M = -[\partial f(\mathbf{u}_0)/\partial \mathbf{u}]^{-1}$			$M = -[\partial f(\mathbf{u}_n)/\partial \mathbf{u}]^{-1}$		
Chebyshev acceleration	[a,b]	$\Delta x=1/16$	$\Delta x=1/64$	$\Delta x=1/256$	$\Delta x=1/16$	$\Delta x=1/64$	$\Delta x=1/256$
no	--	<b>32</b>	55	87	<b>33</b>	<b>55</b>	<b>80</b>
yes	[0,.75]	99	179	260	124	216	332
yes	[0,.50]	58	102	151	64	104	169
yes	[0,.40]	48	88	125	54	96	134
yes	[0,.30]	43	74	107	44	82	116
yes	[0,.20]	37	62	95	39	69	98
yes	[0,.10]	<b>32</b>	<b>53</b>	<b>79</b>	34	61	89

#### 4. SMOOTHED SSOR ITERATION

If M is chosen according to the SSOR matrix listed in Table 1.1, then the resulting iteration method becomes a smoothed SSOR iteration method. We shall first derive suitable relaxation parameters for the model situation (1.20), and then we shall show, by means of numerical experiments, that these parameters are also effective in nonmodel cases.

##### 4.1. Derivation of relaxation parameters

Assuming that (1.20) is satisfied, we find, upon substitution of the SSOR matrix in the expression (1.22), the matrix

$$(4.1) \quad \frac{\partial F}{\partial \mathbf{u}} = \prod_{r \in \mathbb{R}} [I - \rho \omega_r A^{-T} (2/\varpi - 1) (C/\varpi + U)^{-1} C (C/\varpi + L)^{-1} [T_2 r(I+2D) - I]/2].$$

In the following we shall allow that  $\varpi$  also depends on r, and we shall write  $\gamma_r := \omega_r A^{-T}$ .

From this expression we obtain the following theorem:

**Theorem 4.1.** (a) *Let D and  $\partial f/\partial \mathbf{u}$  be related according to (1.20). Then the Jacobian matrix  $\partial F/\partial \mathbf{u}$  of the iteration function in (1.18) is given by*

$$(4.2) \quad \partial F / \partial \mathbf{u} = \prod_{r \in \mathbb{R}} [I - \gamma_r (2 - \omega_r) (D + E)^{-1} [T_{2r}(I + 2D) - I] / 2],$$

$$E := [\omega_r LC^{-1}U + (1 - \omega_r)C / \omega_r] / \rho$$

(b) If  $\partial f / \partial \mathbf{u}$  is a tridiagonal matrix with lower diagonal, diagonal and upper diagonal elements respectively given by  $l_j, j=2, \dots, m$ , by  $c_j, j=1, \dots, m$ , and by  $u_j, j=1, \dots, m-1$ , then  $E$  is a diagonal matrix with entries

$$(4.3) \quad e_j = [\omega_r l_j u_{j-1} / c_{j-1} + (1 - \omega_r) c_j / \omega_r] / \rho, \quad j = 1, \dots, m; \quad l_1 = 0,$$

with  $e_0$  and  $e_{m+1}$  irrelevant.

**Proof.** It is easily shown that the matrix (4.1) can be written in the form

$$(4.1') \quad \partial F / \partial \mathbf{u} = \prod_{r \in \mathbb{R}} [I - \rho \gamma_r (2 - \omega_r) (\partial f / \partial \mathbf{u} + \rho E)^{-1} [T_{2r}(I + 2D) - I] / 2],$$

where  $E$  is defined as in (4.2). Hence, using (1.20) yields the representation (4.2). The proof of part (b) of the theorem is straightforward by verification.  $\blacksquare$

In order to get some insight into the eigenvalues of the matrix  $\partial F / \partial \mathbf{u}$ , we consider the case where  $\partial f / \partial \mathbf{u}$  is tridiagonal; then the theorem states that  $E$  is diagonal. The usual approach now is to apply the frozen coefficient technique, that is, to consider the entries of the matrix  $E$  to be independent of  $j$ , thus simulating the analysis for a linear problem. Unfortunately, even in the case of the model problem, the assumption that  $E$  is a constant matrix is not true. This can be seen from the definition of  $E$ : all entries of  $E$  are equal (in the model case), except for the first element  $e_1$ , because the matrix  $LC^{-1}U$  gives no contribution for this first element ( $l_1=0$ ). This exception has some consequences which will be discussed below. For the moment we ignore this deficiency and continue the analysis.

Let us write

$$e_j = d \omega_r + c (1 - \omega_r) / \omega_r, \quad j = 1, \dots, m,$$

with  $d$  and  $c$  constant (for example, in the model problem (2.1), we have  $d = -1/8$  and  $c = -1/2$ ), then the eigenvalues of  $\partial F / \partial \mathbf{u}$  are given by

$$(4.4) \quad \alpha(\mu) = \prod_{r \in \mathbb{R}} \alpha_r(\mu),$$

$$\alpha_r(\mu) := 1 - \gamma_r (2 - \omega_r) [\mu + d \omega_r + c (1 - \omega_r) / \omega_r]^{-1} [T_{2r}(1 + 2\mu) - 1] / 2.$$

Here, as before,  $\mu$  runs through the eigenvalues of  $D$ .

Using this expression, we performed numerically a minimization process for the maximal value of  $|\alpha(\mu)|$  on the eigenvalue interval of  $D$ . In this minimization process we imposed the constraint  $0 < \omega_r < 2$ , which is natural in the SSOR-context. Furthermore, because of the deficiency discussed above, we examined the

eigenvalues of  $\partial\mathbf{F}/\partial\mathbf{u}$  for the first element of the cycle separately (i.e., for  $r = 0$ ), using the actual value for  $e_1$ . That is, we considered the matrix  $(\mathbf{D}+\mathbf{E})^{-1}\mathbf{D}$  (cf. (4.2)). It turned out that, for all values of  $\varpi_0$ , this matrix possesses an eigenvalue  $1/(2-\varpi_0)$ , resulting in an eigenvalue  $1-\gamma_0$  for  $\partial\mathbf{F}/\partial\mathbf{u}$ . Hence, convergence of this unsmoothed SSOR process requires  $0 < \gamma_0 < 2$ . The necessity of this requirement was experimentally verified. For this reason, we imposed the additional constraint  $0 < \gamma_r < 2$  in the minimization process.

The eigenvalues of  $\mathbf{D}$  defined by (1.19) are known from which we derive  $-1 \leq \mu \leq -[1 - \cos(\pi/2q)]/2$ , where we have assumed the relation  $m + 1 = 2q$ ,  $m + 2$  being the dimension of  $\mathbf{D}$ . In Table 4.1, more or less optimal parameters are given, together with the value of  $\tilde{\beta}$ , denoting the spectral radius of  $\partial\mathbf{F}/\partial\mathbf{u}$ , i.e.,

$$\tilde{\beta} := \text{maximum of } |\alpha(\mu)| \text{ on the interval of eigenvalues } \mu \text{ of } \mathbf{D}.$$

These parameters are determined on the basis of the model problem (2.1), where  $d = -1/8$ ,  $c = -1/2$ ,  $\mathbb{R} := \{0, 1, \dots, q\}$ , and  $q = 1$  until 8. These values were produced by the NAG routine E04 JAF for a suitable initial guess which was obtained by trial and error.

Although the above analysis is of restricted value (because of the deficiency in the frozen coefficient approach) and consequently the given parameters  $\gamma_r$  and  $\varpi_r$  will not be optimal, the values of  $\tilde{\beta}$  given in Table 4.1 are impressive small. In addition, our numerical experiments show that these parameters considerably increase the rate of convergence. We emphasize, however, that the given parameter sets are not unique with respect to minimizing the maximum of  $|\alpha(\mu)|$ . We found different sets of parameters which resulted in more or less the same damping factor  $\tilde{\beta}$ .



**Table 4.1.** Smoothed SSOR parameters for the model problem.

$r$	$\gamma_r$	$\omega_r$	$\gamma_r$	$\omega_r$	$\gamma_r$	$\omega_r$	$\gamma_r$	$\omega_r$
0	1.29483	1.16429	1.52664	1.31372	1.43313	1.26135	1.39671	1.23887
1	0.67942	0.93100	0.83590	0.39225	0.59135	0.73279	0.61539	0.66521
2			0.49908	0.49426	0.23099	1.30867	0.25340	1.35313
3					0.19771	0.88092	0.19440	0.95840
4							0.18481	0.84568
	q = 1, $\tilde{\omega} = .0066$		q = 2, $\tilde{\omega} = .0028$		q = 3, $\tilde{\omega} = .0043$		q = 4, $\tilde{\omega} = .0043$	
0	1.92161	1.46464	1.45986	1.26980	1.29871	1.15502	1.27984	1.12992
1	0.90909	1.55156	0.76625	1.54084	0.80108	1.55903	0.45913	1.29275
2	0.24421	1.31166	0.12622	1.45593	0.24561	1.68935	0.31452	1.68786
3	0.73262	0.73607	0.18459	1.24836	0.16416	1.88198	0.25115	1.06478
4	0.32470	0.57986	0.09664	1.43038	0.21041	1.10603	0.38700	1.22488
5	1.24289	0.18750	0.07353	1.29864	0.20718	1.38827	0.10989	1.09238
6			0.19498	0.78077	0.09785	1.11741	0.21708	0.79981
7					0.11317	1.04884	0.08668	1.21313
8							0.09560	1.15921
	q = 5, $\tilde{\omega} = .0041$		q = 6, $\tilde{\omega} = .019$		q = 7, $\tilde{\omega} = .025$		q = 8, $\tilde{\omega} = .019$	

We conclude our analysis of smoothed SSOR with a picture of the behaviour of the function  $\alpha(\mu)$ . In Figure 4.1 this function is plotted for  $q=3$ .

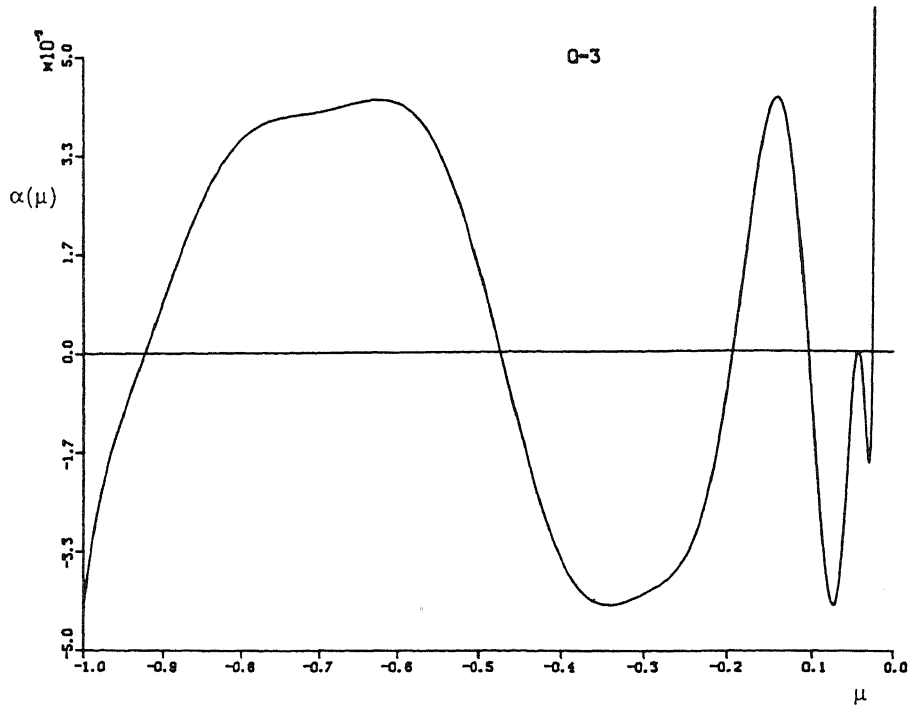


Figure 4.1. Behaviour of the function  $\alpha(\mu)$  for  $q = 3$ .

#### 4.2. Numerical experiments

By using residue smoothing employing the parameter values of Table 4.1, we obtain for the model problem the results as listed in Table 4.2. If, in addition, the Chebyshev acceleration is applied to this smoothed process, then the gain is almost negligible. This behaviour is similar to that observed in the preceding section on smoothed Jacobi iteration.

**Table 4.2.** Smoothed SSOR method for the model problem.

Chebyshev acceleration	[a,b]	$\Delta x=1/16$	$\Delta x=1/64$	$\Delta x=1/256$
no	--	<b>17</b>	39	64
yes	$[-\tilde{b},+\tilde{b}]$	18	<b>38</b>	<b>56</b>
yes	$[-.01,+.01]$	21	39	63
yes	$[-.1,+.1]$	27	50	74

The conclusion must be that the accelerating effect of the smoothing technique is so strong that the Chebyshev acceleration (which was invented to speed up traditional, unsmoothed basic iteration methods) is of no use in this case.

Next, we apply the same methods to the nonmodel problem (2.3). The analogue of Table 3.4 is given by Table 4.3. The conclusions that can be drawn from this table are the same as mentioned above. Finally, when compared with smoothed Jacobi, smoothed SSOR, provided with more or less optimal parameters, is slightly faster; however, the price to be paid is a more complicated algorithm.

**Table 4.3.** Smoothed SSOR method for problem (2.3).

Chebyshev acceleration	[a,b]	$\Delta x=1/16$	$\Delta x=1/64$	$\Delta x=1/256$
no	--	26	50	83
yes	$[-\tilde{b},+\tilde{b}]$	<b>26</b>	50	<b>83</b>
yes	$[-.01,+.01]$	26	50	83
yes	$[-.1,+.1]$	31	<b>49</b>	91
yes	$[-.2,+.2]$	37	58	110

## REFERENCES

- [1] HAGEMAN, L.A. & YOUNG, D.M. (1981), Applied iterative methods. Academic Press, New York.
- [2] HOUWEN, P.J. VAN DER, BOON, C. & WUBS, F.W. (1988), Analysis

- of smoothing matrices for the preconditioning of elliptic difference equations.  
To appear in *Z. Angew. Math. Mech.* 68.
- [3] HOUWEN, P.J. VAN DER & SOMMEIJER, B.P. (1988), A fast algorithm for solving nonlinear elliptic difference equations on vector computers. In preparation.
  - [4] LYUSTERNIK, L.A. (1947), *Trudy Mat. Inst. Steklov* 20, pp. 49-64 (Russian).
  - [5] THIJE BOONKKAMP, J.H.M. ten (1987), Residual smoothing for accelerating the ADI iteration method for elliptic difference equations. Report NM-R8721, CWI, Amsterdam.