Lecture Notes in Mathematics

Edited by A. Dold and B. Eckmann

572

Sparse Matrix Techniques

Copenhagen 1976

Lecture Notes in Mathematics

Edited by A. Dold and B. Eckmann

572

Sparse Matrix Techniques

Copenhagen 1976

Advanced Course Held at the
Technical University of Denmark
Copenhagen, August 9–12, 1976

Edited by V. A. Barker

Springer-Verlag
Berlin · Heidelberg · New York 1977

Editor
V. A. Barker
Institute for Numerical Analysis
Technical University of Denmark
Copenhagen/Denmark

Library of Congress Cataloging in Publication Data

Course in Advanced Sparse Matrix Techniques, Technical University of Denmark, 1976.
Sparse matrix techniques.

(Lecture notes in mathematics ; 572)
1. Matrices--Addresses, essays, lectures. 2. Equations, Simultaneous--Addresses, essays, lectures.
I. Barker, Vincent Allan, 1934- II. Title.
III. Series: Lecture notes in mathematics (Berlin) ; 572.
QA3.L28 no. 572 [QA185] 510'.8s [512.9'43] 77-1128

AMS Subject Classifications (1970): 65-02, 65F05, 65F10, 65F15, 65F20, 65N20, 94A20

ISBN 3-540-08130-5 Springer-Verlag Berlin · Heidelberg · New York
ISBN 0-387-08130-5 Springer-Verlag New York · Heidelberg · Berlin

Printed in Germany
Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr. 43210

TABLE OF CONTENTS

	Page
OWE AXELSSON Solution of linear systems of equations: iterative methods	1
J. ALAN GEORGE Solution of linear systems of equations: direct methods for finite element problems	52
JOHN K. REID Solution of linear systems of equations: direct methods (general)	102
AXEL RUHE Computation of eigenvalues and eigenvectors	130

SOLUTION OF LINEAR SYSTEMS OF EQUATIONS: ITERATIVE METHODS

Owe Axelsson

Dept. of Computer Sciences
Chalmers University of Technology
S 402 20 Gothenburg, Sweden

CONTENTS

1. Introduction 2
2. The classical conjugate gradient algorithm 5
2.1. The conjugate gradient method as an optimization method 6
2.2. The conjugate gradient algorithm 9
3. The Chebyshev semi-iterative method 11
3.1. The one-step Chebyshev semi-iterative method 11
3.2. The two-step Chebyshev acceleration method 13
4. Rate of convergence of the conjugate gradient method 17
4.1. Spectrum contained in two intervals of equal length 18
4.2. Spectrum contained in two arbitrary positive intervals 19
5. The preconditioned conjugate gradient algorithm 22
5.1. The preconditioned algorithm 22
5.2. A two-step preconditioned conjugate gradient algorithm 24
5.3. Some preconditioning techniques 26
5.4. The SSOR preconditioning matrix 27
6. Iterative methods for some special classes of matrices 31
6.1. A special system of equations 31
6.2. The Chebyshev semi-iterative method for matrices with complex eigenvalues 33
6.3. A coupled Chebyshev semi-iterative method and conjugate gradient method 34
7. Applications of iterative methods to partial differential equation problems 37
7.1. History of some iterative methods for differenced Dirichlet problems 37
7.2. SSOR preconditioning of finite difference and finite element matrices 39
7.3. Some special methods 42
7.4. Some numerical results for the SSOR preconditioned conjugate gradient method 42
7.5. Conclusions 46
References 48

1. Introduction.

This lecture series will deal with sparse matrix techniques. For years (see Section 7.1), iterative methods have been used for the solution of large linear systems of equations. The obvious advantage with iterative methods is that they usually demand a minimum of storage: only non-zero coefficients of the given matrix have to be stored plus the coefficients in one or a few vectors such as the solution vector and, for example, the residual vector.

For certain classes of matrices, like diagonally dominant matrices, the simplest methods such as the simultaneous iteration method and the successive overrelaxation (SOR) method, with overrelaxation parameter $0 < \omega < 2$, are convergent (see e.g. [1]). For faster convergence, the class of matrices to be dealt with must possess further properties.

For instance, a classical result for consistently ordered matrices (2-cyclic matrices, see [1]) is that the convergence of the SOR method with optimal parameter ω is given by the spectral radius $\rho(\mathcal{L}_\omega) = 2/[1 + \sqrt{1-\rho(B)^2}] - 1$ (cf. Section 5.4).

It is apparent that the more properties the class of matrices at hand satisfies, the faster are the methods that can be developed taking advantage of these properties. Thus, for instance, for plane self-adjoint 2^{nd} order elliptic partial differential equation problems, which after discretization by finite elements or finite differences lead to a positive definite matrix A, the following qualitative figure can be given (cf. Section 7.5).

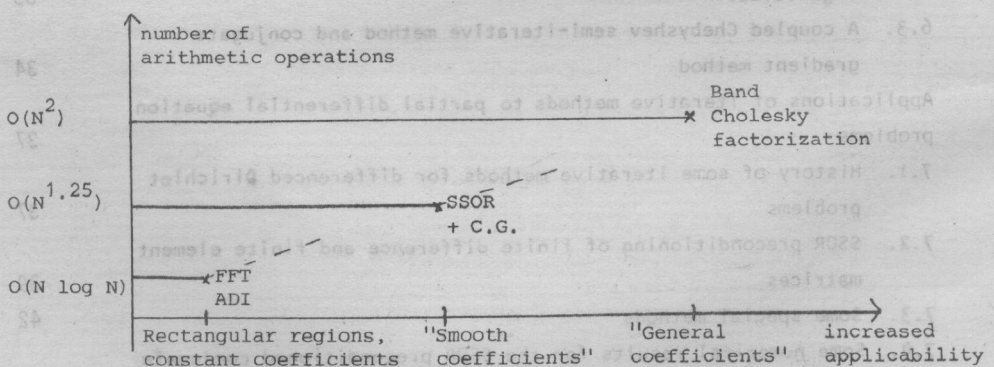

Figure 1.1

We have used the following notations:

N number of unknowns
FFT fast Fourier transform technique
ADI alternating direction implicit method - theory applicable to commutative splittings of A or to separable problems
SSOR symmetric successive overrelaxation method
C.G. conjugate gradient method

Thus in order to prove useful results and to be able to make comparisons between commonly used direct and iterative methods, we will here only deal with *symmetric and positive definite matrices*. Furthermore, in order to *prove* the efficiency of (but not to *use*) the SSOR preconditioning we will need that a further condition is satisfied. This condition is satisfied in many finite element matrix problems (see Section 5.4).

For positive definite matrices A the conjugate gradient method has been used both as a *terminating* (direct) method - theoretically at most N steps are necessary - and as an *iterative* method. We will use the method as an iterative method and as such the number of iterations to reach a given relative accuracy is directly proportional to the square root of the spectral condition number, $\mathcal{K}(A)$, of A, the quotient between the largest and the smallest eigenvalue of A. For special clusterings of the eigenvalues, we will prove that the method may converge faster, however.

The conjugate gradient method is perhaps best presented as an *optimization method* which in fact is not only applicable to linear systems of equations but also to non-linear systems of equations for which an appropriate corresponding functional to be minimized is given (see Section 2). As such, the method has been used by Hestenes [2], Hestenes and Stiefel [3], Fletcher and Reeves [4], Polak and Ribiére [5], Daniel [6] and by Bartels and Daniel [7] among others.

For linear systems of equations, there are two conjugate gradient algorithms in use :

(*i*) the above mentioned one-step optimization algorithm with updating of search directions
(*ii*) the two-step algorithm

which both have orthogonal (or conjugately orthogonal) residuals.
There is a relationship with the Chebyshev semi-iterative method which will be made clear in Section 3. Actually, a bound on the the rate of convergence of the conjugate gradient method is easily proved using Chebyshev min-max theory, both in the classical case with eigenvalues considered in a single interval of the positive real axis and in the more general case where two or more intervals are considered (see Section 4).

Motivated by the result on the rate of convergence, some so called preconditionings of the matrix A will be mentioned. The term *preconditioning* in connection with the Chebyshev semi-iterative method was used by Evans [8]. The technique was used by D'Yakonov [9] (1961), Habetler and Wachspress [10] (1961), Gunn [11], Dupont et al. [12] and also in connexion with the conjugate gradient method by Axelsson [13] (1972), [14], [15] and [16]. Most of them have used a method similar to the SSOR method as a preconditioning device. Young [17] also used the SSOR method for preconditioning. A numerical comparison between the Chebyshev semi-iterative and the conjugate gradient methods as accelerating devices, based on the (generalized) SSOR method, is made in [13] and [15]. In Bartels and Daniel [7], the discretized Laplacian was used as a preconditioning device (and the Laplacian was "inverted" by FFT-techniques). Later Concus et al. [51] have reviewed some different preconditioning techniques and they use the term "generalized conjugate gradient method". In Section 5 a review of some preconditioning techniques will be given and a simple theory for the SSOR preconditioning will be presented.

Iterative methods for some special matrices will be considered and in Section 7 a review of iterative methods for the classical discretized second-order elliptic problem is given as well as some more recent results for this problem. Finally some numerical results are presented.

2. The classical conjugate gradient algorithm.

Consider the problem of minimizing a functional $f = f(u)$ of N variables, $u \in R^N$. Let $g = g(u) = \text{grad } f(u)$ be the gradient and $H = H(u) = [\ \frac{\partial^2 f}{\partial u_i \partial u_j}\ ]$ be the Hessian matrix of f. We suppose that H is uniformly positive definite. In order to minimize f, the conjugate gradient method is applicable. In this method the search directions, along which f is successively minimized, are chosen in such a way that they are mutually and conjugately orthogonal (see below).

Before presenting the algorithm, let us mention that such problems are of great importance in physical and engineering sciences where elliptic partial differential problems may be formulated as variational problems; i.e. where the minimal solution of so called *energy functionals*, discretized with finite element approximations, is to be calculated over the corresponding finite dimensional space S_N.

Thus the energy functional usually has such a form that

$$\text{grad } f(u) = A(u)u - b(u),$$

where $A = A(u)$ is the so called "stiffness" matrix given by

$$A_{ij} = \iint_{\Omega} a(u)\ \nabla\phi_i^T(x)\ \nabla\phi_j(x)\ dx\ .$$

$a = a(u) > 0$ is a material coefficient (for instance the diffusion coefficient),

$$u = \sum_{i=1}^{N} u_i\ \phi_i(x), \qquad S_N = \text{SPAN}\ \{\phi_i(x)\}_{i=1}^{N}\ ,$$

ϕ_i being the basis functions, and $b = b(\alpha)$ comes from the "source-terms". The equilibrium is thus reached at a point where

$$A(u)u = b(u),$$

this being a necessary condition for extremum. This is in general a nonlinear system of equations. Since the Hessian is supposed to be positive definite, it is also a sufficient condition, and we have a unique solution.

In a linear problem, a is constant and thus A is a constant matrix, independent of the solution vector. Furthermore b is constant. We will for simplicity assume that b is constant in the following but will occasionally make remarks on how to handle problems with a non-constant matrix A.

In a linear problem, the corresponding functional is quadratic,

(2.1) $$f(u) = \frac{1}{2} u^T A u - b^T u$$

and at the equilibrium point

(2.2) $$\text{grad } f(u) = Au - b = 0 .$$

This linear system of equations is the well known "equilibrium" equations arising from finite elements applied to linear problems.

For various reasons it may be preferable to consider the problem of minimizing (2.1) instead of considering the solution of (2.2) directly.

One reason for dealing with the functional f is that one can easily add penalty terms, for instance taking care of some boundary constraints. Another reason is that when using the conjugate gradient method in order to minimize the functional, f(u) is *minimized in each iterative step* using the information supplied so far during the iterations (namely the different gradients). This rather vague statement will become more clear in the following.

2.1. The conjugate gradient method as an optimization method.

We will thus now derive the conjugate gradient algorithm from an optimization point of view. However we will consider a somewhat more general functional, enabling us to minimize for instance also the residuals.

Thus let

$$f(u) = \frac{1}{2} u^T A^\nu u - (A^{\nu-1} b)^T u , \qquad u \in R^N,$$

where ν is a natural number. Then $g(u) = \text{grad } f(u) = A^\nu u - A^{\nu-1} b = A^{\nu-1} r$, where $r = Au - b$ is the residual. We observe that f(u) and g(u) are calculable for all such values of ν. We have

$$f(u) = \frac{1}{2} (u - \hat{u})^T A^\nu (u - \hat{u}) - \frac{1}{2} \hat{u}^T A^\nu \hat{u}$$

where $\hat{u} = A^{-1} b$ is the solution.

Since the last term is constant, minimizing f(u) is equivalent to minimizing the error functional

(2.3) $$E(u) = \frac{1}{2} (u - \hat{u})^T A^\nu (u - \hat{u}) = \frac{1}{2} r^T A^{\nu-2} r = \frac{1}{2} g(u)^T A^{-\nu} g(u) .$$

For $\nu = 2$ we will thus minimize the Euclidian norm of the residual.

To minimize E (or f) we let $\{d^k\}_{k\geq 0}$ be search directions and $\{\lambda_k\}_{k\geq 0}$ the parameters of exact line search. From

$$u^{k+1} = u^k + \lambda_k d^k , \qquad k = 0,1,\ldots$$

we obtain immediately

$$r^{k+1} = r^k + \lambda_k A\, d^k$$

Since the optimal value of λ_k makes g^{k+1^T} orthogonal to d^k we have

$$0 = g^{k+1^T} d^k = \text{grad } f(u^k + \lambda_k d^k)^T d^k = r^{k+1^T} A^{\nu-1} d^k$$

and hence

$$\lambda_k = - r^{k^T} A^{\nu-1} d^k / d^{k^T} A^{\nu} d^k . \tag{2.4}$$

We now let

$$d^{k+1} = -r^{k+1} + \beta_k d^k , \qquad k = 0,1,\ldots, \qquad d^0 = -r^0$$

i.e., we will in the following conjugate gradient iterative step move along a plane determined by the residual in the last point and the just used search direction. The parameter β_k will be determined later on.

Apparently, since $d^0 = -r^0$, we have

$$d^{k+1} \in \text{SPAN}\{r^0,\ldots,A^{k+1}r^0\},$$

i.e., a linear combination of elements in the Krylov sequence $r^0, Ar^0, \ldots, A^{k+1}r^0$, and likewise

$$r^{k+1} \in \text{SPAN}\{r^0,\ldots,A^{k+1}r^0\}.$$

Thus r^k can be written as a polynomial of degree k in A times r^0 (with constant coefficient = 1), i.e.

$$r^k = (I + P_k(A))r^0 ,$$

where $P_k(0) = 0$. Thus

$$E(u^k) = \frac{1}{2} r^{k^T} A^{\nu-2} r^k = \frac{1}{2} r^{0^T} (I + P_k(A)) A^{\nu-2} (I + P_k(A)) r^0 . \tag{2.5}$$

Introducing the norm (we observe that A is positive definite)

$$\| u \|_{A^{\nu-2}} = (u^T A^{\nu-2} u)^{1/2} ,$$

we have

$$(E(u^k))^{1/2} = \frac{1}{\sqrt{2}} \| r^0 + P_k(A) r^0 \|_{A^{\nu-2}} .$$

Thus $-P_k(A)r^0$, where $P_k(0) = 0$, can be considered as an approximation of r^0 with error $(E(u^k))^{1/2}$. To minimize this error, it is well known from Hilbert space theory that the error has to be orthogonal (with respect to the corresponding inner product $\langle u,v \rangle = u^T A^{\nu-2} v$) to all functions (vectors) in the linear subspace of approximating functions ; i.e.,

$$\{[I + P_k(A)]r^0\}^T A^{\nu-2} P_l(A) r^0 = 0 , \qquad l \leq k$$

or equivalently

$$\{[I + P_k(A)]r^0\}^T A^{\nu-1} [I + P_l(A)] r^0 = 0 , \qquad l \leq k-1 .$$

Thus

$$r^{k^T} A^{\nu-1} r^l = 0 , \qquad l \leq k-1 ,$$

i.e., the residual vectors will be mutually orthogonal for $\nu = 1$ and in general will be *conjugately orthogonal* with respect to the matrix $A^{\nu-1}$. We will now show that this implies that the search directions will be conjugately orthogonal with respect to the matrix A^ν, i.e. $d^{k^T} A^\nu d^l = 0$, $l \neq k$.

Thus let $l < k$. Then

$$d^{k^T} A^\nu d^l = (A d^k)^T A^{\nu-1} d^l = \frac{1}{\lambda_k} (r^{k+1} - r^k)^T A^{\nu-1} d^l$$

$$= \frac{1}{\lambda_k} (r^{k+1} - r^k)^T A^{\nu-1} (-r^l + \beta_{l-1} d^{l-1}) = \frac{\beta_{l-1}}{\lambda_k} (r^{k+1} - r^k)^T A^{\nu-1} d^{l-1} ,$$

where we have used the conjugacy between the residual vectors. By induction,

$$d^{k^T} A^\nu d^l = \frac{1}{\lambda_k} (r^{k+1} - r^k)^T A^{\nu-1} d^l = \frac{\beta_{l-1}}{\lambda_k} (r^{k+1} - r^k)^T A^{\nu-1} d^{l-1} = \ldots =$$

$$= \beta_{l-1}\beta_{l-2} \cdots \beta_0 / \lambda_k (r^{k+1} - r^k)^T A^{\nu-1} d^0 ,$$

which is = 0, *since* $d^0 = - r^0$. From this conjugacy, we have in particular

$$(-r^{k+1} + \beta_k d^k)^T A^\nu d^k = 0 ,$$

i.e.,

$$\beta_k = \frac{r^{k+1^T} A^{\nu} d^k}{d^{k^T} A^{\nu} d^k} .$$

Since due to exact line searches (which is achieved in a linear problem with the given λ_k-value) we have $d^{k^T} g^{k+1} = r^{k+1^T} A^{\nu-1} d^k = 0$; i.e.,with

$$(2.6) \qquad d^k = -r^k + \beta_{k-1} d^{k-1} = -r^k + \beta_{k-1}(-r^{k-1} - \ldots)$$

we get

$$\beta_k = \frac{r^{k+1^T} A^{\nu-1}(r^{k+1} - r^k)}{d^{k^T} A^{\nu-1}(r^{k+1} - r^k)} = \frac{r^{k+1^T} A^{\nu-1}(r^{k+1} - r^k)}{r^{k^T} A^{\nu-1} r^k} .$$

This formula with $\nu = 1$ (due to Polak, Ribiére [5]) is recommended by Powell [18] for more general (non quadratic) optimization problems. For the above considered quadratic problem it can further be simplified to

$$\beta_k = \frac{r^{k+1^T} A^{\nu-1} r^{k+1}}{r^{k^T} A^{\nu-1} r^k} = \frac{g^{k+1^T} r^{k+1}}{g^{k^T} r^k} .$$

We observe that in the last expression, the Hessian is not needed. Furthermore, from (2.6) we can simplify (2.4),

$$\lambda_k = g^{k^T} r^k / d^{k^T} A^{\nu} d^k .$$

2.2. The conjugate gradient algorithm.

For an arbitrary initial approximation u^0, the algorithm thus takes the following form:

```
        u:= u^0;
        r:= Au - b;  g:= A^(ν-1) r;  d:= -r;  δ0:= g^T r;
R:      λ:= δ0/d^T A^ν d;
        u:= u + λd;
        r:= Au - b;  g:= A^(ν-1) r;  δ1:= g^T r;
        β:= δ1/δ0;  δ0:= δ1;
        d:= -r + βd;
        IF  r^T r > ε  THEN GOTO  R;
```

For $\nu = 1$ we get the classical conjugate gradient algorithm (cf. [4]), and in variational formulation of elliptic partial differential equations (2.1) then represents the energy in the given system. $\nu = 1$ is thus the appropriate choice. The stopping criterion however ought perhaps be a test on a small enough change in f instead of the test on the residuals. We also observe that $g = r$ and $r^T r = \delta 1$ in that case, so some simplifications are possible in the algorithm (see Section 5).

For $\nu = 2$ we will minimize the Euclidian norm of the residuals, and this may be more natural in other applications than the one just mentioned. Then we also have a stopping criterion in the iteration cycle of a quantity which is to be minimized.

Remark. Instead of calculating the residual according to the definition as done above, it is in a linear problem possible to use the recursion formula

$$r := r + \lambda A d .$$

In the case $\nu = 1$, this will decrease the number of matrix-vector multiplications at each iterative step from 2 to 1 at the expense of having to store one extra vector, namely Ad.

Reid [19] has done a numerical comparison between these two approaches and found only minor differences in the true residual and the one calculated by the recursion formula. The number of vectors to be stored is 3 (u,r,d) compared to 4. (For further results on the computational complexity see [19].)

3. The Chebyshev semi-iterative method.

In this section we will prove that the rate of convergence of the Chebyshev semi-iterative method is determined by the spectral condition number λ_0/λ_1, where λ_1,λ_0 are the smallest and largest of the eigenvalues of A. This will then enable us to give an upper bound on the number of iterations needed in the conjugate gradient method (see Section 4).

3.1. The one-step Chebyshev semi-iterative method.

Let us at first consider the following one-step iterative procedure

$$u^{l+1} = u^l - \tau_{l+1}(Au^l - b), \qquad l = 0,1,\ldots \tag{3.1}$$

for the solution of $Au = b$. Here $\{\tau_l\}$ is a parameter set, the proper choice of which gives a possible accelerated convergence over the simplest choice $\tau_l = \tau$, $l = 1,2,\ldots$. With $\tau = 2/(\lambda_0 + \lambda_1)$ we then have the smallest spectral radius $\rho(I - \tau A)$ $= (1 - \lambda_1/\lambda_0)/(1 + \lambda_1/\lambda_0)$. The relative error in the Euclidian norm is then decreased to a number at most $\varepsilon > 0$, if $[(1 - \lambda_1/\lambda_0)/(1 + \lambda_1/\lambda_0)]^p \leq \varepsilon$, and this inequality is satisfied if

$$p \geq \frac{1}{2}\,\frac{\lambda_0}{\lambda_1}\ln\frac{1}{\varepsilon}\,.$$

The number of necessary iterations are thus in general directly proportional to the spectral condition number of A.

To get an accelerated convergence, we choose a suitable set $\{\tau_l\}$ in (3.1) in order to minimize the corresponding iteration matrix achieved after p iterations. Then the errors satisfy

$$e^p = Q_p(A)e^0$$

where

$$e^p = u^p - u,$$

$$Q_p(\lambda) = \prod_{l=1}^{p}(1 - \tau_l\lambda)\,,$$

and $Q_p(A)$ is the corresponding matrix polynomial. We observe that $Q_p(0) = 1$. We denote by Π_p^1,Π_p^0 the set of polynomials of degree at most p that are = 1 and = 0, respectively, at the origin.

More generally, we would like to minimize the residual $r^p = Au^p - b$, or the error $e^p = A^{-1}r^p$, in the norm

$$\| v \|_{A^{-\nu}} = (v^T A^{-\nu} v)^{1/2} , \qquad \nu \text{ an integer.}$$

Then we have

$$\| r^p \|_{A^{-\nu}} \leq \| Q_p(A) \|_{A^{-\nu}} \| r^0 \|_{A^{-\nu}} ,$$

where, as is easily demonstrated,

$$\| Q_p(A) \|_{A^{-\nu}} = \max_i |Q_p(\lambda_i)| ,$$

which we thus want to minimize.

It is well known that the least maximum is achieved by the Chebyshev polynomials, namely

$$(3.2) \qquad \min_{Q_p \in \Pi_p^1} \max_{\lambda_1 \leq \lambda \leq \lambda_0} |Q_p(\lambda)| = \max_{\lambda_1 \leq \lambda \leq \lambda_0} \frac{|T_p((\lambda_0 + \lambda_1 - 2\lambda)/(\lambda_0 - \lambda_1))|}{T_p((\lambda_0 + \lambda_1)/(\lambda_0 - \lambda_1))}$$

$$= 1/T_p((\lambda_0 + \lambda_1)/(\lambda_0 - \lambda_1)) ,$$

where

$$T_p(z) = \frac{1}{2}\left[(z + \sqrt{z^2-1})^p + (z - \sqrt{z^2-1})^p\right] .$$

The optimal choice of the parameters τ_l are thus given by the zeros of T_p, i.e.,

$$\frac{1}{\tau_l} = \frac{\lambda_0 - \lambda_1}{2} \cos \theta_l + \frac{\lambda_0 + \lambda_1}{2} ,$$

where

$$\theta_l = \frac{2 \cdot l - 1}{2p} \pi , \qquad l = 1,2,\ldots,p.$$

In practice, the smallest and largest eigenvalues are not known, so we need lower and upper bounds, a ($a > 0$) and b, respectively. Then we have to use the parameters

$$\frac{1}{\tau_l} = \frac{b-a}{2} \cos \theta_l + \frac{b+a}{2} .$$

It is an easy matter to find that

$$1/T_p((b + a)/(b - a)) \leq 2\left(\frac{1 - \sqrt{a/b}}{1 + \sqrt{a/b}}\right)^p ,$$

so that if

$$p \geq \frac{1}{2}\sqrt{\frac{b}{a}}\ln\frac{2}{\varepsilon} , \qquad \varepsilon > 0$$

then

$$\min_{Q_p \in \Pi_p^1} \max_{a \leq \lambda \leq b} |Q_p(\lambda)| \leq \varepsilon ,$$

i.e., the relative error in the norm $\|\cdot\|_{A^{-\nu}}$ is at most ε after a cycle of p iterations, as given above.

We observe that in this process, which defines the classical Chebyshev semi-iterative method or Richardson method (see [20]), we have to choose the value of p in advance. Furthermore, it is easily seen that in the parameter set $\{\tau_l\}$ there is a non-empty set for which the matrices $I - \tau_l A$ have spectral radius much larger than 1. This will cause the process to be numerically unstable unless we use some particular permutations of the parameters (see [21]).

Both these disadvantages may be eliminated in the following way.

3.2. The two-step Chebyshev acceleration method.

Thus consider the following two-step formula

$$(3.3) \qquad \begin{aligned} u^{l+1} &= \alpha_l u^l + (1 - \alpha_l)u^{l-1} - \beta_l r^l , \qquad l = 1,2,\ldots, \\ u^1 &= u^0 - \tfrac{1}{2}\beta_0 r^0 \end{aligned}$$

We will show how to choose the parameter set $\{\alpha_l, \beta_l\}$ in order that this process apart from rounding errors gives the same result as the one-step Chebyshev process *for every p*. We have with the already introduced notations,

$$e^l = Q_l(A)e^0$$

and observing that the recursion formula (3.3) is valid for all initial vectors, we get

$$Q_{l+1}(A) - \alpha_l Q_l(A) + \beta_l A Q_l(A) + (\alpha_l - 1)Q_{l-1}(A) = 0 , \qquad l = 1,2,\ldots$$

Comparing this with the recursion formula for the Chebyshev polynomials