

# Sparse matrix methods in quantum chemistry \*

## Post-doctorale cursus quantumchemie

Han-sur-Lesse, Belgium

Dr. ir. Gerrit C. Groenenboom

14 june - 18 june, 1993  
Last revised: May 11, 2005

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Eigenvalue problems . . . . .	2
1.2	Linear equations . . . . .	2
1.3	Full matrix methods . . . . .	3
1.3.1	Eigenvalue problems [1, 2, 3, 4] . . . . .	3
1.3.2	Linear equations [3, 4] . . . . .	4
<b>2</b>	<b>Sparse eigenvalue problems</b>	<b>4</b>
2.1	The power method . . . . .	5
2.2	Power method with diagonal shift . . . . .	5
2.3	Convergence of the power method . . . . .	5
2.4	Inverse iteration . . . . .	5
2.5	The Krylov space [2] . . . . .	6
2.6	Approximation from a subspace . . . . .	6
2.7	The Lanczos method . . . . .	7
2.8	Relaxation methods . . . . .	8
2.9	Gradient methods . . . . .	8
2.10	The Davidson algorithm [10] . . . . .	8
<b>3</b>	<b>Sparse linear systems <math>Ax=b</math></b>	<b>9</b>
3.1	Conjugate Gradient method [3, 4, 11] . . . . .	10
3.2	Preconditioning [3] . . . . .	10
3.3	Minimal residual method . . . . .	11

---

\*Original version in Dutch. English translation by Guillaume Dhont and G. C. Groenenboom

# 1 Introduction

		Linear systems	
		general	positive definite
Full:	Householder-tridiagonalisation with: QR-algorithm or: Sturm-sequences and bisection	Gauss elimination ( $PA = LU$ )	Cholesky ( $A = LL^T$ )
Sparse:	Lanczos (+RRGM) Davidson	SYMMLQ, MINRES GMRES (=DIIS)	conjugate-gradient (+preconditioning)

## 1.1 Eigenvalue problems

$$A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad i = 1, \dots, n \quad (1)$$

Unless said otherwise, we restrict ourselves to  $A$  as real, symmetric  $n \times n$  matrices. The eigenvectors are orthonormal:

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \quad (2)$$

The spectrum of  $A$  is formed by the eigenvalues  $\{\lambda_1, \dots, \lambda_n\}$ . In matrix notation:

$$U = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_n] \quad (3)$$

$$U^T U = U U^T = I \quad (4)$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \quad (5)$$

$$AU = U\Lambda \quad (6)$$

## 1.2 Linear equations

$$A\mathbf{x} = \mathbf{b} \quad (7)$$

There are special methods when  $A$  is symmetric and positive definite.

**Definition 1**  $A$  is positive definite if  $\mathbf{x}^T A \mathbf{x} > 0$  for all  $\mathbf{x} \neq \mathbf{0}$ .

- The eigenvalues of symmetric positive definite matrices are positive.
- Positive definite matrices are regular and their inverse are positive definite too.
- If  $A$  and  $A^T$  are diagonal dominant ( $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$  for all  $i$ ) and all  $a_{ii} > 0$  then  $A$  is positive definite.

## 1.3 Full matrix methods

Key idea: the matrix  $A$  is small enough to fit in the fast memory of the computer. We also speak of *in-core* methods. The numerical mathematics literature speaks of *direct* methods. The quantum chemistry literature uses the term *direct* methods for sparse-matrix methods.

In-core methods are based on the application of orthogonal or non-orthogonal transformations of  $A$ . In general, the required computation time scales as  $n^3$ . The possible presence of zeroes in the matrix will not be exploited and moreover these zeroes will generally be lost during the transformations.

The structure can however be preserved during the transformations if  $A$  is a *band-matrix* ( $a_{ij} = 0$  for  $|i - j| > m$ ). This gives a considerable saving in time and memory uses for small value of  $m$ .

There is a lot of good software for *in-core* matrix problems. It should never be necessary to write a routine oneself. Use for example:

- LINPACK
- EISPACK (eigenvalue problems)
- NAGLIB
- IMSL
- BLAS (elementary matrix-vector routines)

### 1.3.1 Eigenvalue problems [1, 2, 3, 4]

First step: make  $A$  tridiagonal ( $a_{ij} = 0$  for  $|i - j| > 1$ ) with a succession of Householder transformations (reflections)

$$P^T A P = T \tag{8}$$

where  $P$  is the product of the transformations and thus orthonormal. The eigenvalues of  $T$  are thus the same as the eigenvalues of  $A$ . The cpu-time  $\sim n^3$ .

Second step

- All eigenvalues/vectors are needed: use the QR-algorithm (cpu  $\sim n^3$ ).
- Only a few eigenvalues are needed: look for the roots of the characteristic polynomial

$$P(\lambda) = \det |T - \lambda I| \tag{9}$$

The eigenvalues smaller than  $\lambda$  can be determined using the so-called Sturm-sequences (cpu  $\sim n$ ). A given eigenvalue can be found this way in combination with the bisection method without having to calculate all the other ones. The associated eigenvector can then be determined using the *inverse iteration*.

### 1.3.2 Linear equations [3, 4]

The  $LU$ -decomposition of  $A$  is computed using the Gauss-eliminations.

$$PA = LU \quad (10)$$

where  $L$  (Lower) is a lower triangular matrix:

$$L = \begin{bmatrix} 1 & & & & \\ l_{2,1} & 1 & & & \\ \vdots & & \ddots & & \\ l_{n,1} & \dots & l_{n,n-1} & 1 & \end{bmatrix} \quad (11)$$

and  $U$  (Upper) an upper triangular matrix:

$$P = \begin{bmatrix} u_{11} & \dots & \dots & u_{1n} \\ & u_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \quad (12)$$

and  $P$  is a permutation of the rows of  $A$  necessary to make the method numerically stable (*partial pivoting*). After these decomposition is determined ( $\text{cpu} \sim n^3$ ), the solutions can be determined for arbitrary right members using the backsubstitution.

$$A\mathbf{x} = \mathbf{b} \quad (13)$$

$$L\mathbf{y} = P\mathbf{b} \quad (14)$$

$$U\mathbf{x} = \mathbf{y} \quad (15)$$

If  $A$  is positive definite and symmetric, the Cholesky decomposition is more efficient:

$$A = LL^T \quad (16)$$

( $L$  does not necessary have ones on the diagonal).

## 2 Sparse eigenvalue problems

If  $A$  is sparse one can try a cunning pivoting strategy to introduce as little as possible non-zeroes elements during the  $LU$ -decomposition. Such methods are called direct-sparse methods [5]. The iterative methods form an alternative. The matrix  $A$  appears only as an operator, *i.e.* as a subroutine for the computation of

$$\mathbf{y} = A\mathbf{x} \quad (17)$$

This means that the matrix never needs to be determined explicitly. In this case quantum chemistry litterature speaks of *direct* methods. We restrict ourselves to iterative methods [1, 2, 3, 4].

## 2.1 The power method

This method is rarely used in practice. However, the power method is the basis for the development of many other methods. The idea :

$$\mathbf{x}^{(k)} = A^k \mathbf{x}^{(0)} \quad (18)$$

converges (after normalization) towards the eigenvector ( $\mathbf{u}_n$ ) belonging to the largest eigenvalue,  $|\lambda_n| > |\lambda_i|, i \neq n$  provided  $\mathbf{x}^{(0)}$  is not exactly perpendicular to this eigenvector ( $\mathbf{u}_n^T \mathbf{x}^{(0)} \neq 0$ ). Due to numerical errors, this last condition is not essential in practice. This method can be implemented iteratively :

$$\mathbf{x}^{(k+1)} = A\mathbf{x}^{(k)} \quad (19)$$

Note: an estimate of the corresponding eigenvalue can be calculated with the following:

$$\lambda_n \approx \frac{\|\mathbf{x}^{(k+1)}\|}{\|\mathbf{x}^{(k)}\|} \quad (20)$$

## 2.2 Power method with diagonal shift

Let  $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n$ . By applying the power method on  $A - \sigma I$  with the correct  $\sigma$ , the pairs  $\{\lambda_1, \mathbf{u}_1\}$  as well as  $\{\lambda_n, \mathbf{u}_n\}$  can be found.

## 2.3 Convergence of the power method

Let  $|\lambda_1| < |\lambda_2| \leq \dots \leq |\lambda_{n-1}| \leq |\lambda_n|$ , then the convergence is:

$$\left| \frac{\|\mathbf{x}^{(k+1)}\|}{\|\mathbf{x}^{(k)}\|} - \lambda_n \right| \sim \left( \frac{|\lambda_n|}{|\lambda_{n-1}|} \right)^k \quad (21)$$

## 2.4 Inverse iteration

Apply the power method to  $(A - \sigma I)^{-1}$ . The spectrum of  $(A - \sigma I)^{-1}$  is  $\{(\lambda_i - \sigma)^{-1}, i = 1, \dots, n\}$ . This therefore provides the eigenvector associated to the closest eigenvalue of  $\sigma$ . If a good approximation of a certain eigenvalue is known, the associated eigenvector can be frequently found in one step. Implementation: solve

$$(A - \sigma I)\mathbf{x} = \mathbf{b} \quad (22)$$

for arbitrary  $\mathbf{b} \neq \mathbf{0}$ . This seems numerically dangerous because the system will (almost) be singular. This is however not the case because the error will be in the direction of the searched eigenvector. This method is for example interesting if  $A$  is tridiagonal (so that the system can be solved easily) and one or a few eigenvalues can be computed accurately. If the eigenvalues are close to each other, there is a risk that the eigenvectors calculated with the inverse iteration will not be well orthogonal.

## 2.5 The Krylov space [2]

In the power method, all intermediary results  $A^k \mathbf{x}^{(0)}$  are thrown away. These vectors contain however valuable information.

**Definition 2** *Krylov matrices:*

$$K^m(\mathbf{x}, A) \equiv [\mathbf{x}, A\mathbf{x}, A^2\mathbf{x}, \dots, A^{m-1}\mathbf{x}] \quad (23)$$

*Krylov subspace:*

$$\mathcal{K}^m(\mathbf{x}, A) \equiv \text{span}\{K^m(\mathbf{x}, A)\} \quad (24)$$

Properties of the Krylov subspace:

$$(i) \quad \mathcal{K}^m(\sigma\mathbf{x}, \tau A) = \mathcal{K}^m(\mathbf{x}, A) \text{ if } \sigma \neq 0 \text{ and } \tau \neq 0 \quad (25)$$

$$(ii) \quad \mathcal{K}^m(\mathbf{x}, A - \sigma I) = \mathcal{K}^m(\mathbf{x}, A) \quad (26)$$

$$(iii) \quad \mathcal{K}^m(P\mathbf{x}, PAP^T) = P\mathcal{K}^m(\mathbf{x}, A) \text{ with } P^T = P^{-1} \quad (27)$$

It follows from the second property that the solutions obtained from the power method are in the Krylov space for all possible shifts. We therefore expect that the eigenvectors which lie in these Krylov spaces will converge from the extremes of the spectrum.

The third property shows that an orthogonal basis transformation of  $A$  and  $\mathbf{x}$  corresponds to an orthogonal transformation of the Krylov space. We can study therefore the properties of Krylov spaces by means of diagonal matrices.

Property one and two put together means that the properties of Krylov spaces only depend on the ratio of the differences between eigenvalues. Indeed appears the so-called *gap ratio*

$$\gamma_i = \frac{\lambda_i - \lambda_{i+1}}{\lambda_{i+1} - \lambda_n} \quad (28)$$

an important estimator to evaluate how difficult it is to find the eigenvector  $\mathbf{u}_i$ .

## 2.6 Approximation from a subspace

Approximations for  $m$  eigenvalues/vectors can be found as following in a  $m$ -dimensional subspace:

1. choose a  $m$ -dimensional orthogonal basis  $q^{(m)}$  for the subspace.

2. compute the projection of  $A$ :

$$\tilde{A} = Q^T A Q \quad (29)$$

3. Solve

$$\tilde{A} \mathbf{y}_i = \theta_i \mathbf{y}_i \quad (30)$$

4. If necessary, transform back the eigenvectors of  $\tilde{A}$

$$\mathbf{x}_i = Q \mathbf{y}_i \quad (31)$$

This is also known as the Rayleigh-Ritz procedure.

## 2.7 The Lanczos method

Within the Lanczos method the approximate eigenvectors are searched in the Krylov space. An orthonormal basis ( $Q^{(m)}$ ) is built in the Krylov space  $\mathcal{K}^m(\mathbf{x}, A)$  with the help of Gram-Schmidt orthogonalisation, so that the space can be extended step by step:

$$\beta_1 \mathbf{q}_1 = \mathbf{x} \quad (32)$$

$$\beta_{i+1} \mathbf{q}_{i+1} = A\mathbf{q}_i - \sum_{j=1}^i (\mathbf{q}_j^T A\mathbf{q}_i) \mathbf{q}_j \quad (33)$$

The part:

$$\tilde{A}_{ji} = \mathbf{q}_j^T A\mathbf{q}_i \quad (34)$$

is tridiagonal. Note:

$$\mathbf{x} \in \mathcal{K}^m \Rightarrow A\mathbf{x} \in \mathcal{K}^{(m+1)} \quad (35)$$

$$\text{span}\{Q^{(m)}\} = \mathcal{K}^m \quad (36)$$

$$\mathbf{q}_j \perp Q^{(i)} \text{ for } j > i \quad (37)$$

Thus:

$$A\mathbf{q}_i \in \text{span}\{Q^{(i+1)}\} \quad (38)$$

Or

$$\mathbf{q}_j^T A\mathbf{q}_i = 0 \text{ for } j > i + 1 \quad (39)$$

This means that  $\tilde{A}$  has a so-called upper Hessenberg structure. Because  $A$  is symmetric,  $\tilde{A}$  must be tridiagonal. For the diagonal element of this tridiagonal matrix we have:

$$T_{ii} = \mathbf{q}_i^T A\mathbf{q}_i = \alpha_i \quad (40)$$

and from equation (33) follows:

$$T_{i+1,i} = \mathbf{q}_{i+1}^T A\mathbf{q}_i = \beta_{i+1} \quad (41)$$

The Lanczos recursion can now be written as :

$$\beta_1 \mathbf{q}_1 = \mathbf{x} \quad (42)$$

$$\beta_{i+1} \mathbf{q}_{i+1} = A\mathbf{q}_i - \alpha_i \mathbf{q}_i - \beta_i \mathbf{q}_{i-1} \quad (43)$$

After the  $m$ th iteration, we thus have, in matrix notation:

$$AQ^{(m)} - Q^{(m)}T^{(m)} = \mathbf{r}_m \mathbf{e}_m^T, \text{ with } \mathbf{r}_m = \beta_{m+1} \mathbf{q}_{m+1} \quad (44)$$

We call Lanczos basis an orthonormal basis constructed that way. It sometimes happens that the  $\mathbf{q}_i$  are not orthogonal due to rounding errors. As a consequence,  $T$  can also contain “spurious” eigenvalues or copies of eigenvalues. This has been well understood in the seventies and solutions have been found to compute trustfully the eigenvalues with the Lanczos method (the original publication of Lanczos [6] is of 1950). (see for example [2, 7, 8]).

A *block-Lanczos* method has to be used if  $A$  contains degenerate eigenvalues.

## 2.8 Relaxation methods

The main point of relaxation methods for the finding of the lowest eigenvalue of  $A$  is the minimalization of the Rayleigh quotient:

$$\rho(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad (45)$$

If we change one component of  $\mathbf{x}$ :

$$\mathbf{y} = \mathbf{x} + \epsilon \mathbf{e}_k \quad (46)$$

and we solve

$$\frac{\partial \rho(\mathbf{y})}{\partial \epsilon} = 0 \quad (47)$$

we find

$$\epsilon = (\rho(\mathbf{y}) - a_{kk})^{-1} (A - \rho(\mathbf{y})) \mathbf{x} \quad (48)$$

This is an implicit expression for  $\epsilon$  because  $\mathbf{y}$  depends on  $\epsilon$ . The development of this equation gives a quadratic expression for  $\epsilon$  (the  $\epsilon^3$  term is neglected):

$$[(A - \rho(\mathbf{x})) \mathbf{x}]_k + \epsilon (a_{kk} - \rho(\mathbf{x})) + \epsilon^2 \frac{[A \mathbf{x}]_k - a_{kk} x_k}{\mathbf{x}^T \mathbf{x}} = 0 \quad (49)$$

These equation is the main point of the ‘‘Method of Optimal Relaxation’’ (MOR) [9].

If we neglect the  $\epsilon^2$  term we get

$$\epsilon = -[a_{kk} - \rho(\mathbf{x})]^{-1} (A - \rho(\mathbf{x})) \mathbf{x} \quad (50)$$

## 2.9 Gradient methods

In the ‘‘method of steepest descent’’, one looks at each step in the direction of the gradient of  $\rho(\mathbf{x})$ :

$$\nabla \rho(\mathbf{x}) = \begin{bmatrix} \frac{\partial \rho}{\partial x_1} \\ \vdots \\ \frac{\partial \rho}{\partial x_n} \end{bmatrix} = \frac{(A - \rho(\mathbf{x})) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad (51)$$

This leads to an iterative process in the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + c_k (A - \rho) \mathbf{x}^{(k)} \quad (52)$$

It thus appears that the gradient method is in fact a search in the Krylov space. One should preferably use the Lanczos method.

## 2.10 The Davidson algorithm [10]

Add to the Lanczos method the idea of looking for solutions in a space whose dimension increases by one at each iteration. Use at each iteration the extra vector

$$(D - \rho(\mathbf{x}))^{-1} (A - \rho(\mathbf{x})) \mathbf{x}^{(k)} \quad (53)$$

with

$$D_{ij} = \delta_{ij} a_{ii} \quad (54)$$



This “search direction” was suggested through the relaxation formula (50). For matrices such as the ones which appears in CI-calculations, this is frequently much efficient than the Lanczos method. A disadvantage is that  $A$  is full in the subspace, instead of a tridiagonal as in the Lanczos methods. This means that at each iteration, one extra basis vector must be stored (generally on disk).

### 3 Sparse linear systems $\mathbf{Ax}=\mathbf{b}$

For an approximate solution  $\mathbf{x}_k$  we define the residu  $\mathbf{r}_k$

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k \quad (55)$$

We look for solutions of the form

$$\mathbf{x}_k = Q_k\mathbf{y}_k \quad (56)$$

where  $Q_k$  is a  $n \times k$  matrix and  $\mathbf{y}_k$  is a vector of length  $k$ . We consider two methods to establish an equation for  $\mathbf{y}_k$

- Choose the projection of the residu on  $\text{span}\{Q_k\}$  equal to zero:

$$Q_k^T \mathbf{r}_k = 0 \quad (57)$$

Or

$$Q_k^T A Q_k \mathbf{y}_k = Q_k^T \mathbf{b} \quad (58)$$

- minimize the length of the residu  $\|\mathbf{r}_k\|$ . This leads to the equation

$$(A Q_k)^T \mathbf{r}_k = 0 \quad (59)$$

Or

$$Q_k^T A^T A Q_k \mathbf{y}_k = Q_k^T A^T \mathbf{b} \quad (60)$$

If  $A$  is symmetrical and if we take a Lanczos basis for  $Q_k$  (with  $A_1 = \mathbf{b}/\|\mathbf{b}\|$ ) then we get from equation (58) the following tridiagonal system:

$$T\mathbf{y} = \|\mathbf{b}\|\mathbf{e}_1 \quad (61)$$

If  $A$  is also positive definite, this scheme can be simply solved using the Cholesky decomposition. It is possible to iterately carry out these decomposition, to solve the resulting system and to backsubstitute for  $\mathbf{y}_k$  in equation (56), so that the complete matrix  $Q_k$  never needs to be stored (the necessary space is for four vectors of length  $n$ ). The resulting method is proved to be mathematically equivalent to the Conjugate Gradient method of Hestenes en Stiefel (see below).

The above discussion however suggests also a solution for the case where  $A$  is symmetrical, but not positive definite. In this case, equation (61) must be solved with a method which does not require  $T$  to be positive definite. In 1975, Paige and Saunders showed that the use of the LQ algorithm is very suitable for this problem. They called the resulting algorithm SYMMLQ [12].

### 3.1 Conjugate Gradient method [3, 4, 11]

The machinery of CG (Conjugate Gradient) has as a main point that the minimalisation of

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b} \quad (62)$$

corresponds to the solution of  $A\mathbf{x} = \mathbf{b}$  if  $A$  is positive definite. If we change  $x$  in the  $\mathbf{p}_k$  direction:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k \quad (63)$$

we find that  $f(\mathbf{x}_k)$  is minimalized for

$$\alpha_k = \mathbf{p}_k^T \mathbf{r}_{k-1} / \mathbf{p}_k^T A \mathbf{p}_k \quad (64)$$

The ingenious idea of the CG method is that the search direction  $\mathbf{p}_k$  is chosen in a simple manner such that

$$\mathbf{p}_i^T A \mathbf{p}_j = 0 \text{ for } i \neq j \quad (65)$$

The result is that for

$$\mathbf{x}_k = \sum_{i=1}^k \alpha_i \mathbf{p}_i \quad (66)$$

we get

$$f(\mathbf{x}_k) = \sum_{i=1}^k \left( \frac{1}{2} \alpha_i^2 \mathbf{p}_i^T A \mathbf{p}_i - \alpha_i \mathbf{p}_i^T \mathbf{b} \right) \quad (67)$$

so that the  $\alpha$ 's can be computed one after the other. The final algorithm consists - apart from practical details - as follows:

*Initialisation:*

$$\mathbf{x}_0 = \mathbf{0} \quad (68)$$

$$\mathbf{r}_0 = \mathbf{b} \quad (69)$$

For  $k = 1, \dots$

If  $\mathbf{r}_{k-1} = \mathbf{0}$ , stop:  $\mathbf{x}_{k-1}$  is the solution.

*Otherwise:*

$$\beta_k = \mathbf{r}_{k-1}^T \mathbf{r}_{k-1} / \mathbf{r}_{k-2}^T \mathbf{r}_{k-2} \quad (\beta_1 \equiv 0) \quad (70)$$

$$\mathbf{p}_k = \mathbf{r}_{k-1} + \beta_k \mathbf{p}_{k-1} \quad (\mathbf{p}_1 \equiv \mathbf{r}_0) \quad (71)$$

$$\alpha_k = \mathbf{r}_{k-1}^T \mathbf{r}_{k-1} / \mathbf{p}_k^T A \mathbf{p}_k \quad (72)$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k \quad (73)$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k A \mathbf{p}_k \quad (74)$$

### 3.2 Preconditioning [3]

The convergence of the CG method depends on the condition number of  $A$  (largest eigenvalue divided by the smallest). If the condition number is 1,  $A$  is a constant multiplied by the identity matrix and the algorithm converges in one step. The idea of *preconditioning* is to multiply the linear system (formal) by an approximate inverse of  $A$ ,  $M^{-1}$ , such that  $M^{-1}A$  has a smaller condition number:

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \quad (75)$$

Because  $M^{-1}A$  does not need to be symmetric, even if  $M$  is symmetric, one actually use

$$M^{-\frac{1}{2}}AM^{\frac{1}{2}}\mathbf{x} = M^{-1}\mathbf{b} \quad (76)$$

The matrix  $M^{-\frac{1}{2}}$  does not need to be explicitly constructed, it is sufficient that there is a routine available for the solution of

$$M\mathbf{x} = \mathbf{y} \quad (77)$$

for an arbitrary  $\mathbf{y}$ . The diagonal of  $A$  can sometimes be a good preconditioner. Preconditioned CG can be used for the inverse iteration of the lowest eigenvalue of  $A$ . This idea is used in a modification of the Davidson algorithm [13].

### 3.3 Minimal residual method

Instead of equation (58) we can use equation (60). This is done in the MINRES algorithm of Paige and Saunders [12]. They compare SYMMLQ and MINRES and note that:

- MINRES is somewhat less precise for very badly conditioned systems.
- MINRES is also suitable for stiff systems, SYMMLQ is not.
- MINRES is sometimes faster to converge if we use the length of the residu as stop-criterium.

For nonsymmetric systems, one can use the Generalized Minimal Residual Methode (GMRES [14]) (this method is also known as DIIS [15], direct inversion of iterated subspace [15]). This results in a full system in the subspace, so that the memory use increases with the number of iterations. Possibly one can restart the calculation: the best approximation  $\mathbf{x}_k$  has been reached at a certain moment and we then write

$$\mathbf{x} = \mathbf{x}_k + \mathbf{d} \quad (78)$$

and continue with

$$A\mathbf{d} = \mathbf{b}' = \mathbf{b} - A\mathbf{x}_k \quad (79)$$

Preconditioning is now also possible, and it is not necessary to use the symmetrized version (equation (76)).

## References

- [1] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford (1965). [A classic]
- [2] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Inc., Englewood Cliffs, N. J. (1980). [Very strongly recommended]
- [3] G. H. Golub and C. F. van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore (1983). [Relatively easy readable]

- [4] J. Stoer and R. Bulirsch, *Introduction to Numerical analysis*, Springer, New York (1980). [Somewhat more mathematical. Does not consider matrix calculations only]
- [5] D. A. H. Jacobs ed., *The State of the Art in Numerical Analysis*, Academic Press, London (1977), Chapter I.3 (by J. K. Reid).
- [6] C. Lanczos, J. Res. Nat. Bur. Stand. **45**, 255 (1950).
- [7] C. C. Paige, J. Inst. Math. Appl. **18**, 341 (1976).
- [8] J. Cullum and R. A. Willoughby, J. Comput. Phys. **44**, 329 (1981).
- [9] I. Shavitt, C. F. Bender, A. Pipano, and R. P. Hosteny, J. Comput. Phys. **11**, 90 (1973).
- [10] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).
- [11] P. E. S. Wormer, F. Visser, and J. Paldus, J. Comput. Phys. **48**, 23 (1982).
- [12] C. C. Paige and M. A. Saunders, SIAM J. Numer. Anal. **12**, 617 (1975).
- [13] J. H. van Lenthe and P. Pulay, J. Comput. Chem. **11**, 1164 (1990).
- [14] Y. Saad and M. H. Schultz, SIAM J. Sci. Stat. Comput. **7**, 856 (1986).
- [15] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980), P. Pulay, J. Comput. Chem. **3**, 556 (1982).