Iterative Solution of Linear Systems

0. Introduction

We consider the problem of solving the system

$$A\mathbf{x} = \mathbf{b} ,$$

where A is a $n \times n$ matrix of rank r; usually $r = n$ but in some situations $r = n - 1$. We ordinarilly assume that A is symmetric and positive definite. In some situations, however, we have the following cases
i) A is symmetric and indefinite
ii) $A \; != \; A^T$ but $\frac{A + A^T}{2}$ is positive definite
iii) A is positive semi-definite.
The problem of solving a linear system arises in a variety of situations, such as
1) partial differential equations
2) least squares calculations
3) geodetic calculations.
In addition one frequently deals with non-linear problems which are solved as a sequence of linear problems. The modern era, for the studies in this field, goes back to the early 50's. There were many contributions but the pioneers were D. Young, M. Hestens and E. Sirefel. Their work is of relevance today, even though it is now interpreted in a different fashion than was originally proposed. Indeed, modern computer architectures has encouraged us to rethink many of the procedures that have been in used and we consider some of the methods described previously. In this context one interesting point is the following; for 2D finite element computations direct methods were very much the vogue but, with more powerful computers, one wishes to solve 3D problems and here it is necessary to use iterative methods.

Even though we discuss the problems from a matrix analysis point of view, it is always important to keep in mind the particular application that is being solved. Thus, if a partial differential equation is under consideration we should use that as our motivation and derivation.

1. Preliminaires and Motivations

Let us consider the linear system

$$A\mathbf{x} = \mathbf{b} , \tag{1.1}$$

where entries of A are, here and in the following, real. We rewrite this system in the form

$$M\mathbf{x} \;=\; N\mathbf{x} \;+\; b\;,$$

where $M \;-\; N \;=\; A$; this is a *splitting* of A.
Given an arbitrary initial value x^0 we consider the following simple iteration

$$M\mathbf{x}^{k\,+\,1} \;=\; N\mathbf{x}^k \;+\; \mathbf{b}\;; \tag{1.2}$$

defining $\mathbf{e}^k := \mathbf{x}^k \;-\; \mathbf{x}$ we have

$$M\mathbf{e}^{k\,+\,1} \;=\; N\mathbf{e}^k\;,$$

hence

$$\mathbf{e}^k \;=\; B^k\mathbf{e}^0$$

where we have set

$$B \;=\; M^{-1}N$$

The iteration matrix B plays an important role in the analysis of the convergence of the iterative method (1.2). In fact, defining the *spectral radius* of B as the maximum of the modulus of the eigenvalues of B, that is

$$\rho(B) \;=\; max_{1\;<=\;i\;<=\;n}|\lambda_i(B)|\;, \tag{1.3}$$

we have the following
Theorem 1.1 - For any initial value $\mathbf{x}^0$, $\mathbf{e}^0 \;\rightarrow\; 0$ as $k \;\rightarrow\; \infty$ iff $\rho(B) \;<\; 1$. When $\rho(B) \;<\; 1$, we say that B is *convergent*. If $\rho(B)$ is small we have a "fast" rate of convergence; however it is important to keep in mind that the system (1.2) must be easily solvable. Hence our goal is to choose B so that the product $\rho(B)$x (number of operations per iteration) is small. Concerning the problems of storage of B we are also interested in the analysis of the data structure.

We want to compare various choices of B. For this we need several definitions and theorems.
Definition 1.1 - A matrix $A \;=\; a_{ij}\_i, j \;=\; 1, n$ is said to be an M-matrix if
i) $a_{ij} \;<=\; 0$ for $i, j \;=\; 1, ..., n\;,\; i \;\neq\; j$
ii) A is nonsingular
iii) $(A^{-\,1})_{ij} \;>=\; 0$ for $i, j \;=\; 1, ..., n$.

If A is such that $a_{ij} <= 0$ for $i \neq j$ and $a_{ii} > \sum_{j=1, j\neq i}^{n} |a_{ij}|$ (*strictly diagonal dominant*), then A is an M-matrix. In fact let us write A in the form

$$A = D - K$$

where $D = d_{ij}\_i, j = 1$, $d_{ii} = a_{ii}$, $d_{ij} = 0$ for $i \neq j$. Since A is strictly diagonal dominant it is non singular, moreover, from the Gershgorin theorem (see | |) $\rho(D^{-1}K < 1$ and we have (see | |)

$$A^{-1} = (I - D^{-1}K)^{-1} = \sum_{k=0}^{\infty} (D^{-1}K)^k D^{-1} >= 0 . \quad \square$$

Definition 1.2 - $A = M - N$ is said a *regular splitting* of the matrix A if M^{-1} exists with $M^{-1} >= 0$ and $N >= 0$.

Theorem 1.2 - Let A be such that $A^{-1} >= 0$ and consider two regular splittings $A = M_1 - N_1 = M_2 - N_2$ with $N_2 \leq N_1$, then

$$\rho(M_2^{-1}N_2) <= \rho(M_1^{-1}N_1) < 1 . \quad \square$$

This theorem, due to Varga, gives a monotonicity property: if the values of N decrease then the convergence is faster. We remark, however, that decreasing the values of N can give difficulties in solving (1.2).

Theorem 1.3 - Let A be a symmetric positive definite matrix and let $A = M - N$ where M is non singular; if the matrix $Q := M + M^T - A$ is positive definite then $\rho(M^{-1}N) < 1$. $\square$

This theorem yields an easy check for testing convergence for this class of iterative methods.

2. Sample problem

In this section we consider the problem of solving the linear system arising from the finite difference approximation of the Poisson's equation on a square.

Let $\Omega = (0,1) \times (0,1)$. Given a function f, let u be the solution of the problem

$$\begin{cases} -\Delta u = f & in\ \Omega \\ u = 0 & on\ \partial\Omega\ . \end{cases} \tag{2.1}$$

For the given discretization parameter N, we consider the following uniform mesh

Fig. 1

where $h = \frac{1}{N+1}$. Setting $x_i = ih$, $y_j = jh$ and $f_{ij} = f(x_i, y_j)$ for $i, j = 0, ..., N+1$ we can find an approximation v_{ij} of $u(x_i, y_j)$ solving the equations

$$\begin{cases} 4v_{ij} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1} = h^2 f_{ij} \quad \forall i,j = 1, ..., N \\ v_{0,j} = v_{N+1,j} = v_{i,0} = v_{i,N+1} = 0 \quad \forall i,j = 0, ..., N+1\ . \end{cases} \tag{2.2}$$

These relations can be written as a linear system as follows. We define

$$\mathbf{v}_j = \begin{bmatrix} v_{1j} \\ v_{2j} \\ . \\ . \\ . \\ v_{Nj} \end{bmatrix} , \mathbf{g}_j = h^2 \begin{bmatrix} f_{1j} \\ f_{2j} \\ . \\ . \\ . \\ f_{Nj} \end{bmatrix} \quad for\ i,j = 1, ..., N,\ \mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ . \\ . \\ . \\ \mathbf{g}_N \end{bmatrix} ,$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ . \\ . \\ . \\ \mathbf{v}_N \end{bmatrix} , B = \begin{bmatrix} 4 & -1 & 0 & . & . & 0 \\ -1 & 4 & -1 & 0 & . & 0 \\ 0 & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & -1 \\ 0 & . & . & 0 & -1 & 4 \end{bmatrix} , A = \begin{bmatrix} B & -I & 0 & . & 0 \\ -I & B & -I & 0 & 0 \\ . & . & . & . & . \\ . & . & . & . & -I \\ 0 & . & . & -I & B \end{bmatrix} .$$

Then the unknown vector $\mathbf{v}$ is the solution of the linear system

$$A\mathbf{v} \;=\; \mathbf{g}\;; \tag{2.3}$$

A is a square matrix of order N^2. This system can be solved as follows. For $j \;=\; 1,...,N$ we have

$$-\;\mathbf{v}_{j-1} \;+\; B\mathbf{v}_j \;-\; \mathbf{v}_{j+1} \;=\; \mathbf{g}_j\;, \tag{2.4}$$

since B is symmetric we can diagonalize it by an orthogonal transform:

$$B \;=\; Q\; LAMBDA\; Q^T\;.$$

From (2.4) we obtain

$$-\mathbf{v}_{j-1} \;+\; Q\; LAMBDA\; Q^T\mathbf{v}_j \;-\; \mathbf{v}_{j+1} \;=\; \mathbf{g}_j \;\; for \;\; j \;=\; 1,...,N\;,$$

and hence

$$-Q^T\mathbf{v}_{j-1} + \; LAMBDA\; Q^T\mathbf{v}_j - Q^T\mathbf{v}_{j+1} \;=\; Q^T\mathbf{g}_j \quad j \;=\; 1,...,N\;.$$

Now setting $\hat{\mathbf{v}}_j \;=\; Q^T\mathbf{v}_j$ and $\hat{g}_j \;=\; Q^T\mathbf{g}_j$ for $j \;=\; 0,...,N+1$, we obtain

$$-\hat{\mathbf{v}}_{i,j-1} \;+\; \lambda_i\hat{\mathbf{v}}_{i,j} \;-\; \hat{\mathbf{v}}_{i,j+1} \;=\; \hat{g}_{ij} \;\; for \;\; i,j \;=\; 1,...,N\;, \tag{2.5}$$

where λ_i are the eigenvalues of B. These equations can be written as

$$\begin{bmatrix} \lambda_i & -1 & 0 & . & . & 0 \\ -1 & \lambda_i & -1 & . & . & 0 \\ 0 & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & -1 \\ 0 & . & . & . & -1 & \lambda_i \end{bmatrix} \quad \begin{bmatrix} \hat{v}_{i1} \\ \hat{v}_{i2} \\ . \\ . \\ . \\ \hat{v}_{iN} \end{bmatrix} \quad = \quad \begin{bmatrix} \hat{g}_{i1} \\ \hat{g}_{i2} \\ . \\ . \\ . \\ \hat{g}_{iN} \end{bmatrix} \quad for\; i \;=\; 1,...,N\;. \tag{2.6}$$

These systems can be easily solved since the matrices are tridiagonal. Then one can obtain the vectors $\hat{\mathbf{v}}_j$ for $j = 1,...,N$. Finally the solution of (2.3) is given by

$$\mathbf{v}_j \;=\; Q\hat{\mathbf{v}}_j \quad for \quad j \;=\; 1,...,N\;. \tag{2.7}$$

We remark that the eigenvalues λ_i are given by

$$\lambda_i \;=\; 4 \;+\; 2\cos\;\frac{i\pi}{N\;+\;1}\;\;,\;\; for\;\; i \;=\; 1,...,N$$

and the elements of the matrix Q are

$$q_{ij} \;=\; \sqrt{\frac{1}{N\;+\;1}}\;\sin\;\frac{ij\pi}{N\;+\;1}\quad i,j\;=\;1,...,N$$

Hence the vectors $Q^T\mathbf{g}_j$ and $Q\hat{\mathbf{v}}_j$ can be computed using the Fast Fourier Transform (FFT). Summarizing, this method for solving (2.3) is composed of three steps:
1) compute $\hat{\mathbf{g}}_j \;=\; Q^T\mathbf{g}_j$ for $j \;=\; 1,...,N$ using FFT with, for any i, $2N\log_2 N$ operation.
2) Solve the systems (2.6) for $i \;= 1,..,N$ with, for any i, N operations.
3) Compute $\mathbf{v}_j \;=\; Q\hat{\mathbf{v}}_j$ for $j \;=\; 1,...,N$ using FFT with, for any j, $2N\log_2 N$ operations.
Note that this method can be used (with minor modifications) for solving the finite difference approximation of the Helmoltz equation, say.

$$-\,\Delta u \;+\; \sigma u \;=\; f \quad where\;\; \sigma \;=\; constant \;>\; 0\;.$$

Other techniques can be used to derive fast Poisson's solvers on a rectangular domain. For instance, making use of the separation of variables the eigenvalues and the eigenvectors of A can be computed explicitely and the linear system can be solved by diagonalisation (see | |).

We recall also the cyclic reduction scheme and we refer to Golub Van Loan | | for its description.

Poisson's equation on a square is a model problem. Many other problems can be reduced to solving a sequence of Poisson's equation on simple domains, as a square or a rectangle. We consider, for instance, Poisson's equation on a T-shape domain.

Fig. 2

Consider a set of mesh point as in figure 2. The finite difference discretization of the Poisson's equation gives rise to a linear system whose

matrix is

$$A = \begin{bmatrix} B & -I & . & 0 & . & . & . & . \\ -I & B & . & . & . & . & 0 & . \\ 0 & . & . & -I & . & . & . & . \\ . & -I & . & B & J & . & . & . \\ . & . & . & J^T & C & -I & . & 0 \\ . & . & . & . & -I & . & . & . \\ . & 0 & . & . & . & . & . & -I \\ . & . & . & . & . & . & -I & C \end{bmatrix}$$

where B, C and J are matrices of size $p \times p$, $q \times q$ and $p \times q$ respectively, given by

$$B = \begin{bmatrix} 4 & -1 & . & 0 \\ -1 & . & . & . \\ . & . & . & -1 \\ 0 & . & -1 & 4 \end{bmatrix} \quad C = \begin{bmatrix} 4 & -1 & . & 0 \\ -1 & . & . & . \\ . & . & . & -1 \\ 0 & . & -1 & 4 \end{bmatrix} \quad J^T = \begin{bmatrix} 0 \\ \\ -I_p \\ 0 \end{bmatrix}$$

where I_p denotes the identity matrix of order p.
The matrix A can be decomposed as follows

$$A = \begin{bmatrix} A_0 & . & 0 \\ & & \\ 0 & & A_1 \end{bmatrix} + \begin{bmatrix} 0 & . & 0 \\ & J & \\ 0 & J^T & 0 \end{bmatrix} =: \tilde{A} + RS^T$$

where

$$R = \begin{bmatrix} 0 & 0 \\ 0 & J \\ I_q & 0 \\ 0 & 0 \end{bmatrix} \quad \textit{and} \quad S = \begin{bmatrix} 0 & 0 \\ J & 0 \\ 0 & I_q \\ 0 & 0 \end{bmatrix} .$$

It can be verified that the $2M \times 2M$ matrix $S^T\tilde{A}^{-1}R + I_{2q}$ is non-singular, hence we can define

$$W = (S^T\tilde{A}^{-1}R + I_{2q})^{-1} . \tag{2.8}$$

We note

$$S^T = (S^T\tilde{A}^{-1}R + I_{2q})WS^T = S^T\tilde{A}^{-1}RWS^T + WS^T . \tag{2.9}$$

Then

$$(\tilde{A} + RS^T)(\tilde{A}^{-1} - \tilde{A}^{-1}RWS^T\tilde{A}^{-1}) = \tag{2.10}$$
$$= (I - RWS^T\tilde{A}^{-1} + RS^T\tilde{A}^{-1} - RS^T\tilde{A}^{-1}RWS^T\tilde{A}^{-1}) =$$
$$= (I - R(WS^T - S^T + S^T\tilde{A}^{-1}RWS^T)\tilde{A}^{-1}) .$$

From (2.9) we deduce

$$(\tilde{A} - RS^T)(\tilde{A}^{-1}RWS^T\tilde{A}^{-1}) = I ,$$

hence we have the relation

$$A^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1}RWS^T\tilde{A}^{-1} . \tag{2.11}$$

This formula suggests a method for solving the linear system arising from the finite difference discretization of the Poisson's equation on a T-shape domain, namely

$$A\mathbf{v} = \mathbf{g} \tag{2.12}$$

The first step of the method consists in solving the system

$$\tilde{A}\mathbf{u} = \mathbf{g} , \tag{2.13}$$

due to the structure of $\tilde{A}$ this system, can be solved efficiently using a Poisson solver on a square and on a rectangle. Then one compute the vector $RWS^T\mathbf{u}$ and find ξ such that

$$\tilde{A}\xi = RWS^T\mathbf{u} ; \tag{2.14}$$

from (2.11) we obtain that $\mathbf{u} - \xi$ is the solution of (2.12).
The matrix W requires $p + q$ fast Poisson solutions, i.e.

$$\tilde{A}Z = R$$

since some columns of R are null.
Because of the zero structure of S, not all components of Z need to be computed. W is called capacitance matrix. We remark that it does not depend on the data $\mathbf{g}$ but only on the domain where the Poisson's equation is to be solved. This method has been studied by Widlund and his co-authors | |; an analogous strategy is based on a splitting of A of the type (see | |)

$$A = \begin{bmatrix} A'_0 & 0 \\ 0 & A'_1 \end{bmatrix} + \begin{bmatrix} 0 & . & . & . & 0 \\ . & J^TJ & J & & . \\ . & J^T & I_q & & . \\ 0 & . & . & . & 0 \end{bmatrix} ,$$

in this case the capacitance matrices is of order q.

Fast Poisson's Solvers can be used as black boxes also for solving second order elliptic problems with non constant coefficients. Let us consider the following problem

$$\begin{array}{rcll} -del\ .\ (adelu) & = & f & in\ \ \Omega \\ u & = & g & on\ \ \partial\Omega \end{array} \tag{2.15}$$

where Ω is a rectangular domain and $a(x,y) > 0$.
The transformation $\mathbf{v} = \sqrt{a}\mathbf{u}$ applied to problem (2.15), gives

$$-\Delta\mathbf{v} + \sigma(x,y)\mathbf{v} = f\ , \tag{2.16}$$

where $f' = f/\sqrt{a}$.
Let us discretize problem (2.16) with usual finite difference schemes over a uniform mesh. Denote by A the matrix associated with the Laplace's operator and by Σ the (diagonal) matrix associated with the zero order term.
Define an iterative procedure as follows:

$$A\mathbf{v}^{k+1} + \Sigma\mathbf{v}^{k} = \mathbf{f}'\quad , \tag{2.17}$$

Then norm of the iteration matrix $A^{-1}\Sigma$ can be easily bounded:

$$\|A^{-1}\Sigma\|_2 <= \sigma_{max}\ \frac{h^2}{2-2\cos\pi h} = \sigma_{max}\ \frac{h^2}{4\sin^2\frac{\pi h}{2}} \qquad const. + O(h^2)\ . \tag{2.18}$$

where $\sigma_{max} = max_{(x,y)\ \in\ \Omega}|\sigma(x,y)|$.
The estimate (2.18) implies that the rate of convergence of the algorithm (2.17) is independent of the mesh size. Of course the amount of work for each iteration depends on the mesh size. In any case, each step makes use of a Fast Poisson's Solver and the total amount of work is reasonable.

If $\sigma(x,y)$ is known analitically, one can rewrite the problem in terms of the Helmoltz operator.

3. Jacobi and Gauss-Seidel Methods

Given a matrix $A = a_{ij}\_i, j = 1, ..., n$ we define the matrices $D = d_{ij}\_i, j = 1, n$, $L = l_{ij}\_i, j = 1, n$ and $U = u_{ij}\_i, j = 1, n$ as follows.

$$\left\{ \begin{array}{rcl} d_{ii} & = & a_{ii} \\ d_{ij} & = & 0 \quad for\ i\ !=\ j \end{array} \right. \qquad \left\{ \begin{array}{rcl} l_{ij} & = & a_{ij} \quad for\ i\ >\ j \\ l_{ij} & = & 0 \quad for\ i\ <=\ j \end{array} \right. \qquad \left\{ \begin{array}{rcl} u_{ij} & = & a_{ij} \quad for\ i\ <\ j \\ u_{ij} & = & 0 \quad for\ i\ >=\ j\ . \end{array} \right.$$

The Jacobi method is defined by choosing $M = D$ and $N = -(L + U)$ in the splitting $A = M - N$ introduced in the first section. More precisely the solution of the linear system $A\mathbf{x} = \mathbf{b}$ is obtained iteratively by solving the system

$$\left\{ \begin{array}{rcl} \mathbf{x}^0 & arbitrarly\ choosen \\ D\mathbf{x}^{k+1} & = & \mathbf{b} - (L + U)\mathbf{x}^k\ . \end{array} \right. \tag{3.1}$$

The iteration matrix associated to the Jacobi method is $B_J = -D^{-1}(L + U)$. About the convergence of the Jacobi method we have the following criteria.

Theorem 3.1 - If A is a strictly diagonal dominant matrix the Jacobi method is convergent. □

To give a convergence theorem for matrices which are not strictly diagonal dominant we need the following

Definition 3.1 - A $n \times n$ matrix A is *irreducible* if it does not exist a permutation matrix P such thast $P^T AP$ is of the form

$$P^T AP = \left| \begin{array}{cc} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{array} \right| ,$$

where $\tilde{A}_{11}$ is of order p and $\tilde{A}_{22}$ is of order q and $p > 0$, $q > 0$.

In order to verify if a matrix A is irreducible one can look at the graph $G(A)$ of A. This graph is composed of n knots $P_1, ..., P_n$ and there exists a directed path $P_i \rightarrow P_j$ if and only if $a_{ij}\ != 0$. The matrix A is irreducible if and only if $G(A)$ is connected, this means that for any pair of knots (P_i, P_j) there exists a directed path from P_i to P_j.

Theorem 3.2 - If A is irreducible with

$$|a_{ii}| >= \sum_{k=1_{k\neq i}}^{n} |a_{ik}| \quad for\ all\ i = 1, ..., n$$

and if there exists an index i_0 such that $|a_{i_0 i_0}| > \sum_{k=1_{k\neq i_0}}^{n} |a_{i_0 k}|$ then the Jacobi method is convergent. □

It can be verified that the matrix A arising from the $O(h^2)$ finite difference approximation to Poisson's equation on a square verifies the hypotheses of theorem 3.2. Hence the Jacobi method is convergent. Concerning the storage problems of the Jacobi method, we remark that two vectors, that is $2n$ components, need be stored. However if the matrix is tridiagonal one need only $n+2$ components stored. To minimize the storage it is important to look at the structure of A. If for example A is of the form

$$A = \begin{bmatrix} x & . & . & . & . & x \\ . & x & & 0 & & x \\ . & & x & & & x \\ . & 0 & & x & & x \\ . & & & & x & x \\ x & x & x & x & x & x \end{bmatrix} ,$$

one can rearrange rows ands columns of A to yield the structure

$$\begin{bmatrix} x & x & x & x & x \\ x & x & & 0 & . \\ x & & x & & . \\ x & 0 & & x & . \\ x & . & . & . & x \end{bmatrix} ,$$

and the storage is smaller for Jacobi method. To solve the finite difference system (2.3) with the Jacobi method we have $D = 4I$ and $B_J = \frac{1}{4}(A - 4I)$. It can be verified that $\rho(B_J) \quad 1 - \frac{\pi h^2}{2}$ and, since B is symmetric, we have $\|B\|_2 = \rho(B_J)$. To get an error $\|\mathbf{e}^k\| <= \epsilon\|\mathbf{e}^0\|$, we must have $\rho^k <= \epsilon$. Hence, $k >= \frac{-\log\epsilon}{-\log\rho}$; that is a number of iterations of the order of $\frac{1}{h^2}$. Suppose now we want to solve Poisson's equation on a T-shaped region Ω_0. We can study the

Fig. 3

convergence of the Jacobi method on Ω_0 by embedding Ω_0 in a larger rectangular domain Ω. Denoting $B_{J,0}$ and B_J the iteration matrices for the two domains, theorem 1.2 implies $\rho(B_{J,0}) <= \rho(B_J) < 1$.

For the finite difference approximation to Poisson's equation on a square one can also use a block Jacobi method. In this case the matrix A of the system (2.1) is splitted as

$$A \;=\; M \;-\; N$$

where

$$M \;=\; \begin{bmatrix} B & \dots & 0 \\ . & & . \\ . & & . \\ 0 & \dots & B \end{bmatrix} \quad and \quad B \;=\; \begin{bmatrix} 4 & -1 & . & 0 \\ -1 & & & . \\ . & & & -1 \\ 0 & . & -1 & 4 \end{bmatrix}$$

The block Jacobi method converges faster that the usual Jacobi one even though the asymptotic behavior is equivalent. Note that at each stage a tridiagonal system must be solved.

The Gauss-Seidel method is defined using the splitting

$$A \;=\; M \;-\; N$$

where

$$M \;=\; D \;+\; L \quad and \quad N \;=\; -\,U$$

The iteration matrix is given by

$$B_{GS} \;=\; -\,(D \;+\; L)^{-1}U\;,$$

for each iteration one implicitly solves a linear system with a triangular matrix. If A is positive definite, the convergence of the Gauss-Seidel method follows from theorem 1.3. In fact the matrix $Q \;=\; M \;+\; M^T \;-\; A$ is positive definite since

$$Q \;=\; M \;+\; M^T \;-\; A \;=\; D \;+\; L \;+\; (D \;+\; L)^T \;-\; A \;=\; D \;>=\; 0\;.$$

Note that the Gauss-Seidel method requires only one vector to be stored.

We can be more specific about the implementation and the study of the convergence of the Gauss-Seidel scheme for an important class of matrices. Suppose the matrix A has the following property, introduced by Young (cf. | |):

Definition 3.2 - A matrix A has *Property* (A) if there exists a permutation matrix Π such that

$$\Pi\; A\; \Pi^T \;=\; \begin{bmatrix} D_1 & F \\ F^T & D_2 \end{bmatrix} \quad , \tag{3.2}$$

where $D_i, i = 1,2$ are diagonal matrices. $\square$
Note that if a matrices is written in the form (3.2), each iteration of the Gauss-Seidel scheme can be parallelized in the following way: splitting the unknown vector $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)^T$, first solve in parallel $D_1 \mathbf{v}_1^{k+1} = \mathbf{b} - F\mathbf{v}_2^k$, then $D_2 \mathbf{v}_2^{k+1} = \mathbf{b} - F^T \mathbf{v}_1^{k+1}$, again parallelizing on the components.
Let us see some examples of matrices with Property (A).
i) Let A arise from the second order finite difference approximation of the second derivative on a uniform mesh. If we order the points in the natural order from the left to the right, A is a tridiagonal symmetric matrix. Instead, if we reorder the points putting first all the odd points and then all the even ones, we obtain a matrix of the form (3.2)

$$\begin{bmatrix}
1 & 3 & 5 & \cdot & 2 & 4 & 6 & . \\
2 & & & & -1 & & & \\
 & 2 & & & -1 & -1 & & \\
 & & 2 & & & -1 & -1 & \\
 & & & & & & & \\
 & & & & & & & \\
-1 & -1 & & & 2 & & & \\
 & -1 & -1 & & & 2 & & \\
 & & & -1 & & & 2 &
\end{bmatrix}$$

ii) If A is the matrix associated with the 5-points discretization of the Poisson's equation, we obtain the parallelizable form (3.2) ordering the points in a checkerboard fashion.

fig. 4

First, all the Red points are collected together, then all the Black points.This ordering will be refered as the Red/Black ordering.

Almost all the matrices arising from partial differential equations have either point Property (A) or block Property (A). This latter property is given as follows.
Definition 3.3 - A has *block - Property* (A) if there exists a permutation matrix Π such that

$$\Pi A \Pi^T = \begin{bmatrix} D_1 & F \\ F^T & D_2 \end{bmatrix} , \qquad (3.3)$$

where $D_i, i = 1, 2$ are block-diagonal matrix.
iii) If A is the matrix associated with the 9-points discretization of the Poisson's equation, A has the form (3.3) by ordering first all the odd lines, and then all the even lines.

fig. 5

iv) Let A be the matrix associated with the 13-points formula for the biharmonic problem

fig. 6

A takes the form

$$A = \begin{bmatrix} A_1 & B_1 & C_1 & 0 & \cdot & \\ 0 & A_2 & B_2 & C_2 & & \\ & & A_i & B_i & C_n - 2 & \\ & & B_i & A_{i+1} & B_{n-1} & \\ & & & & A_n & \end{bmatrix}$$

That is, to enforce Property (A) we need to invert two lines at a time. Of course, one takes advantage of the structure of each matrix. Note that the A_i's are a 5-diagonal mtrices, while the B_i's are tridiagonal. It is better to reorder the unknowns in order to get a 9-diagonal matrix.

We will prove a comparison theorem between the Jacobi and the Gauss-Seidel methods in the case of a matrix A with Property-(A).
Now suppose A takes the form

$$A = \begin{bmatrix} I & F \\ F^T & I \end{bmatrix}$$

which corresponds to (3.2) where a rescaling has been introduced. Then,

the Jacobi iteration matrix B_J is as follows:

$$B_J \;=\; -\,D^{-1}(L\,+\,U) \;=\; \begin{bmatrix} 0 & -\,F \\ -\,F^T & 0 \end{bmatrix} \quad .$$

The Gauss-Seidel iteration matrix B_{GS} is

$$B_{GS} \;=\; -\,(D\,+\,L)^{-1}U \;=\; -\quad \begin{bmatrix} I & 0 \\ F^T & I \end{bmatrix} \quad [\,^{-1}\,] \quad \begin{bmatrix} 0 & F \\ 0 & 0 \end{bmatrix}$$

$$= \quad -\begin{bmatrix} I & 0 \\ -F^T & I \end{bmatrix} \quad \begin{bmatrix} 0 & F \\ 0 & 0 \end{bmatrix} \quad = \; - \quad \begin{bmatrix} 0 & F \\ 0 & -F^TF \end{bmatrix} \quad .$$

Notice that

$$B_J^2 \;=\; \quad \begin{bmatrix} FF^T & 0 \\ 0 & FF^T \end{bmatrix} \quad .$$

Therefore we can state the following theorem.

Theorem 3.3 - If the matrix A has property (A), then

$$\rho(B_{GS}) \;=\; \rho(B_J^2) \quad .$$

This means, of course, that if the methods converge, the Gauss-Seidel scheme obtains the same accuracy as the Jacobi one in roughly half the number of iterations.

If A is associated with the 5-points discretization of the Poisson equation, then

$$\rho(B_J) \;=\; 1 \;-\; \frac{\pi h^2}{2} \;+\; 0(h^4)$$

$$\rho(B_{GS}) \;=\; 1 \;-\; \pi^2 h^2 \;+\; 0(h^4)$$

In order to reach convergence, the number of iterations is in both cases asymptotically of the order of $1/h^2(\quad N^2)$. Together with Theorem 3.3, there are many theorems comparing Jacobi and Gauss-Seidel methods. We refer to Varga | | for an extensive discussion of them.

4. Acceleration algorithms

In this section we want to introduce parametrizations into the algorithm and define accelaration devices. Consider the simple algorithm:

$$\left\{ \begin{array}{rcl} \mathbf{x}^0 & given & ; \\ \mathbf{x}^{k+1} & = & \mathbf{x}^k + \alpha_{k+1}\mathbf{r}^k \ , \\ & where & \\ \mathbf{r}^k & = & \mathbf{b} - A\mathbf{x}^k \ . \end{array} \right. \tag{4.1}$$

The vector $\mathbf{r}^k$ is the residual at the k-th step and α_{k+1} an acceleration parameter.
Notice that $\mathbf{r}^k = A(\mathbf{x} - \mathbf{x}^k) = A\mathbf{e}^k$, therefore

$$\mathbf{e}^{k+1} = \mathbf{x} - \mathbf{x}^{k+1} = (I - \alpha_{k+1}A)\mathbf{e}^k \ . \tag{4.2}$$

Thus, applying recursively (4.2), we obtain

$$\mathbf{e}^k = (I - \alpha_k A)(I - \alpha_{k-1}A)....(I - \alpha_1 A)\mathbf{e}^0 \ . \tag{4.3}$$

The expression (4.3) emphasizes that the error $\mathbf{e}^k$ at the k-th step is obtained by applying to the initial error $\mathbf{e}^0$ a polynomial of degree k in A, say $p_k(A)$, with the property $p_k(0) = 1$.
$p_k(\lambda)$ takes the form:

$$P_k(\lambda) = \Pi_{j=0}^{k}(1 - \alpha_j\lambda) \ .$$

According to different choices of the set of parameters, or, equivalently, of the polynomial $p_k(\lambda)$, we can define a large class of algorithms.

The simplest choice is to set $\alpha_k \equiv \alpha$ for any k. The corresponding $p_k(\lambda)$ reads $p_k(\lambda) = (1 - \alpha\lambda)^k$. Now suppose A has eigenvalues

$$0 < \lambda_1 <= ... \leq \lambda_n \ .$$

We wish to choose $\hat{\alpha}$ that minimizes the $max_{\ \lambda_i}\ |1 - \alpha\lambda_i|$, namely

$$min_{\alpha \in \mathbf{R}}\ max_{\ \lambda_i}\ |1 - \alpha\lambda_i| = max_{\ \lambda_i}\ |1 - \hat{\alpha}\lambda_i| \tag{4.4}$$

Set $\gamma_i(\alpha) = 1 - \alpha\lambda_i$. We see that the method converges as long as $|\gamma_1(\alpha_1)| < 1$, i.e. $0 < \alpha < 2/\lambda_1$. It is clear from figure 7 that $\hat{\alpha}$ satisfies

$$1 - \hat{\alpha}\lambda_n = -(1 - \hat{\alpha}\lambda_1) \ . \tag{4.5}$$

fig. 7

Therefore

$$\hat{\alpha} \;=\; \frac{2}{\lambda_1 \;+\; \lambda_n} \tag{4.6}$$

We can compute explicitely the spectral radius ρ of the iteration matrix $I \;-\; \hat{\alpha}A$:

$$\rho \;=\; |1 \;-\; \hat{\alpha}\lambda_n| \;=\; \frac{\lambda_n \;-\; \lambda_1}{\lambda_n \;+\; \lambda_1} \;=\; \frac{\kappa \;-\; 1}{\kappa \;+\; 1}\;, \tag{4.7}$$

where $\kappa \;=\; \frac{\lambda_n}{\lambda_1}$ is the condition number of A.
The scheme is convergent, but it could be very slow if κ is big. Moreover the definition of the optimal $\hat{\alpha}$ requires the knowledge of the estrem eigenvalues of A and this can be analitically impossible and/or computationally prohibitive, due to the high-cost in many cases.

A more general procedure is defined if we seek $\hat{p}_h(\lambda)$ with $p_k(0) \;=\; 1$, which minimizes $max_{\lambda_1 \leq \lambda \leq \lambda_n}\; |p_k(\lambda)|$ and k arbitrary.
The solution of this minimization problem exploits the properties of the Chebyshev polynomials. $\hat{p}_k(\lambda)$ is found to be:

$$\hat{p}_k(\lambda) \;=\; T_k\Big(\; \frac{2\lambda \;-\; (\lambda_1 \;+\; \lambda_n)}{\lambda_n \;-\; \lambda_1}\Big) \Big/ T_k\Big(\; \frac{\lambda_1 \;+\; \lambda_n}{\lambda_1 \;-\; \lambda_n}\;\Big)\;, \tag{4.8}$$

where $T_k(\mu)$ is the Chebyshev polynomial of degree k defined by the formula

$$T_k(\mu) \;=\; \begin{cases} \cos(k\cos^{-1}\mu) & if \quad |\mu| \leq 1 \\ \cos h(k\cosh^{-1}\mu) & if \quad \mu \geq 1 \end{cases}\;. \tag{4.9}$$

The zeroes of $T_k(\mu)$ are known. Setting $a \;=\; \lambda_1$ and $b \;=\; \lambda_n$, the zeroes of $\hat{p}_k(\lambda)$ are

$$\mu_l \;=\; \frac{a+b}{2} \;+\; \frac{(b-a)}{2}\;\cos\;(\frac{2l-1}{k}\;\frac{\pi}{2})\;,\; l \;=\; 1,...,k\;. \tag{4.10}$$

Therefore the optimal parameters $\hat{\alpha}_l$ take the simple expression:

$$\hat{\alpha}_l \;=\; 1/\mu_l\;,\; l \;=\; 1,...,k\;. \tag{4.11}$$

A short manipulation shows that

$$|\hat{p}_k(\lambda)| \;=\; 2(\frac{\sqrt{\kappa} \;-\; 1}{\sqrt{\kappa} \;+\; 1})^k\;, \tag{4.12}$$

where, as in (4.7), $\kappa = \frac{\lambda_n}{\lambda_1}$.

We point out some disadvantages of this scheme. First, a precise estimate of the extreme eigenvalues is needed and, as already stressed, this can be a quite a difficult computation in itself.
Second, the parameters are not chosen dynamically, but the algorithm requires the a priori choice of k. One cannot switch to an optimal polynomial of degree $k+1$ within the scheme without restarting the process. Usually one uses the parameters cyclically.
Finally, we remark that for the k parameters provided, the ordering in formula (4.11) is arbitrary. Unfortunately due to round-off effects, the ordering of the use of the parameters is very important. The scheme could dramatically blow up if a "good" ordering is not chosen. This problem was solved by Lebeshev and Fenogenov | | .

We recall that there are versions of this scheme that do not make use of the positive definitess of A . De Boor and Rice have considered the use of Chebyshev polynomials over other sets than that considered here, cfr. | |.

Now we go back to the generic scheme (4.1) and we describe a method of choice of the α_k's that does not depend upon the knowledge of the eigenvalues of A.
The scheme (4.1) implies:

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_{k+1} Ar^k . \tag{4.13}$$

We wish to determine the parameter α_{k+1} in order to minimize the norm

$$(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1}) . \tag{4.14}$$

We call this quantity $F(\alpha_{k+1})$, using (4.13) we obtain

$$F(\alpha_{k+1}) = (\mathbf{r}^k, A^{-1}\mathbf{r}^k) - 2\alpha_{k+1}(\mathbf{r}^k, \mathbf{r}^k) + \alpha_{k+1}^2(A\mathbf{r}^k, \mathbf{r}^k) .$$

The minimum of $F(\alpha_{k+1})$ is clearly taken in

$$\hat{\alpha}_{k+1} = \frac{(\mathbf{r}^k, \mathbf{r}^k}{(A\mathbf{r}^k, \mathbf{r}^k)}$$

and thus we are led to the *Steepest Descent Algorithm*

$$\begin{cases} \mathbf{x}^0 \quad given , \\ \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_{k+1}\mathbf{r}^k \quad k >= 1 , \end{cases} \tag{4.15}$$

with

$$\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k \tag{4.16}$$

and

$$\alpha_{k+1} \;=\; \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(A\mathbf{r}^k, \mathbf{r}^k)} \quad . \tag{4.17}$$

Now, note that

$$\frac{(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1})}{(\mathbf{r}^k, A^{-1}\mathbf{r}^k)} \;=\; 1 \;-\; \frac{(r_k, \mathbf{r}_k)^2}{(\mathbf{r}^k, A\mathbf{r}^k)(r^k, A^{-1}r^k)} \; . \tag{4.18}$$

The relative error (4.18) can be bounded by using the *Kantorovich inequality*

$$\frac{(\mathbf{x}, A\mathbf{x})(\mathbf{x}, A^{-1}\mathbf{x})}{(\mathbf{x}, \mathbf{x})^2} \;<=\; \left(\frac{\sqrt{\kappa} \;+\; \frac{1}{\sqrt{\kappa}}}{2} \right)^2$$

where $\kappa \;=\; \frac{\lambda_n}{\lambda_1}$.
We obtain

$$\frac{(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1})}{(\mathbf{r}^k, A^{-1}\mathbf{r}^k)} \;<=\; (\frac{\kappa - 1}{\kappa + 1})^2 \tag{4.19}$$

This inequality implies convergence, but the scheme can be quite slow if κ is big. A preconditioning procedure may be desirable.

5. Successive overrelaxation

The successive overrelaxation method (S.O.R.) was originally developed by D. Young. The idea underlying this method is to parametrize the Gauss-Seidel scheme. If A is a $n \times n$ matrix with $a_{ii} \neq 0$, $i = 1, ..., n$ and we want to solve the system $A\mathbf{x} = \mathbf{b}$, the S.O.R. iteration is the following:

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k), \ for\ i = 1, ..., n\ . \quad (5.1)$$

ω is a real parameter to be choosen, if $\omega = 1$ the S.O.R. method coincides with the Gauss-Seidel method. Writing A in the form $A = D + L + U$ (see section 3), the S.O.R. can be written in matrix form as

$$\mathbf{x}^{k+1} = (1-\omega)\mathbf{x}^k + \omega D^{-1}(\mathbf{b} - L\mathbf{x}^{k+1} - U\mathbf{x}^k)$$

or

$$(I + \omega D^{-1}L)\mathbf{x}^{k+1} \ = \ |(1-\omega)I - \omega D^{-1}U|\mathbf{x}^k + \omega D^{-1}\mathbf{b}\ . \quad (5.2)$$

The iteration matrix is

$$\_\omega := (D + \omega L)^{-1}((1-\omega)D - \omega U) \quad (5.3)$$

Since we have

$$det \quad \_\omega = (1-\omega)^n \leq |\rho(\ \_\omega)|^n\ ,$$

a necessary condition for the convergence of the S.O.R. method is that $0 < \omega < 2$. If A is symmetric positive definite, this is also a sufficient condition, in fact we have the following

Theorem 5.1 - Let A a $n \times n$ symmetric positive definite matrix and suppose that $0 < \omega < 2$, then the S.O.R. method is convergent.

Proof.

Defining $Q := M^T + M - A$, where $M = \frac{1}{\omega}(D + \omega L)$, we have

$$Q = \frac{1}{\omega}(D + \omega L) + \frac{1}{\omega}(D + \omega U) - (D + L + U) = (\frac{2}{\omega} - 1)D\ .$$

Therefore since A is positive definite, Q is positive definite too for $0 < \omega < 2$. From theorem 1.3 we deduce the convergence. □

Remark 5.1 - A block S.O.R. method can be defined and the same proof shows the convergence for this method when $0 < \omega < 2$.

Our aim is now to study which is the best choice of the relaxation parameter ω. We suppose that the matrix A has the Property (A) (see definition 3.2), and the red/black ordering, that is A can be written in the form

$$A = \begin{bmatrix} I_q & F \\ F^T & I_p \end{bmatrix} . \tag{5.4}$$

We write the Singular Value Decomposition (S.V.D.) of F, i.e.

$$F = U\ M\ V^T \tag{5.5}$$

where

$$U^T U = I_q, V^T V = I_p$$

and

$$M = \begin{bmatrix} \mu_1 & & & \\ & \ddots & & \\ & & \mu_p & \\ 0 & & & 0 \end{bmatrix} , \quad \mu_1 \geq \mu_2 \geq, ..., \mu_p \geq 0, \quad \text{with } q \geq p$$

From (5.4) we deduce

$$\mathcal{L}_\omega = \begin{bmatrix} I_q & 0 \\ \omega F^T & I_p \end{bmatrix}^{-1} \begin{bmatrix} (1-\omega)I_q & -\omega F \\ 0 & (1-\omega)I_p \end{bmatrix} ,$$

but

$$\begin{bmatrix} I_q & 0 \\ \omega F^T & I_p \end{bmatrix}^{-1} = \begin{bmatrix} I_q & 0 \\ -\omega F^T & I_p \end{bmatrix}$$

Hence we obtain from (5.5):

$$\mathcal{L}_\omega = \begin{bmatrix} (1-\omega)I_q & -\omega F \\ -\omega(1-\omega)F^T & (1-\omega)I_p + \omega^2 F^T F \end{bmatrix} =$$

$$= \begin{bmatrix} (1-\omega)UU^T & -\omega UMV^T \\ -\omega(1-\omega)VM^T U^T & (1-\omega)I_p + \omega^2 VM^T MV^T \end{bmatrix} =$$

$$= \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} (1-\omega)I_q & -\omega M \\ -\omega(1-\omega)M^T & (1-\omega)I_p + \omega^2 M^T M \end{bmatrix} \begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} .$$

This relation shows that the eigenvalues of $\_\omega$ are the eigenvalues of the matrix.

$$\begin{bmatrix} (1-\omega)I_q & -\omega M \\ -\omega(1-\omega)M^T & (1-\omega)I_p + \omega^2 M^T M \end{bmatrix} .$$

We can now reorder rows and columns of this matrix and write it in a block diagonal form as follows

$(1-\omega)$ is an eigenvalue of $\_\omega$, of multiplicity $q-p$, the other $2p$ eigenvalues are those of the matrices

$$\begin{bmatrix} 1-\omega & -\omega\mu_i \\ -\omega(1-\omega)\mu_i & (1-\omega)+\omega^2\mu^2 \end{bmatrix} \qquad for\ i = 1, ..., p\ .$$

The characteristic polynomials of these matrices are

$$(1-\omega-\lambda)^2 - \lambda\omega^2\mu_i^2 = 0 \quad for\ i = 1, ..., p\ . \tag{5.6}$$

Our aim is to minimize, with respect to ω , the quantity

$$\rho(\ \_\omega) = max(max_{1\le i\le p}|\lambda_i|, |1-\omega|)\ .$$

We remark that $\rho(\ \_1) = \mu_1^2$, and since for $\omega = 1$ we obtain the Gauss-Seidel method we have $\mu_1^2 = \rho(B_{GS}) = \rho^2(B_J)$ (see theorem 3.3).
From (5.6) it turns out that setting $\hat{\omega} = \frac{2}{1+\sqrt{1-\rho^2(B_J)}}$

$$\rho(\ \_\omega) = \begin{cases} \omega - 1 & for\ \ \hat{\omega} \le \omega \le 2 \\ 1-\omega+\frac{1}{2}\omega^2\rho^2(B_J) + \omega\rho(B_J)\sqrt{1-\omega+\frac{1}{2}\omega^2\rho(B_J)} & for\ 0 \le \omega \le \hat{\omega}\ . \end{cases}$$

The curve of $\rho(\quad \_\omega)$ is

fig. 8

Hence the optimum value for the relaxation parameter ω is $\hat{\omega}$ and

$$\rho(\ \_\omega) = \hat{\omega} - 1 = \frac{2}{1 + \sqrt{1 - \rho^2(B_J)}} - 1 = \frac{1 - \sqrt{1 - \rho^2(B_J)}}{1 + \sqrt{1 - \rho^2(B_J)}} \ .$$

We emphasize here that to known the optimal value ω one must know the spectral radius of the Jacobi iteration matrix.

Remark 5.2 - All the eigenvalues of $\ \_\hat{\omega}$ have the same modulus equal to $\hat{\omega} - 1$.

In the case of the finite difference approximation to Poisson's equation $\rho(B_J) = \cos \pi h$. Hence we have

$$\hat{\omega} = \frac{2}{1 + \sin \pi h} \quad and \quad \rho(\ \_\hat{\omega}) = \frac{1 - \sin \pi h}{1 + \sin \pi h} = 1 - h + 0(h^2) \ . \qquad (5.7)$$

With respect to the Jacobi method the number of iterations required for obtaining a prescribed error reduction are here reduced of an order of magnitude. This theory applies to red-black ordering of the finite difference matrix (see section 4), however (5.7) holds true also for other ordering as the natural one and the zebra one, i.e.

fig. 9

We remark that the Jordan form of $\ \_\hat{\omega}$ is not diagonal. This implies that $\| \ \_\hat{\omega}^k \|_2 \ \ k\rho^{k-1}(\ \_\hat{\omega})$ and the error can increase at least for the first iterations (see fig. 10).

fig. 10

Let us now introduce a variant of the S.O.R. method, called Symmetric Successive Overrelaxation method (S.S.O.R.). Given an initial value $\mathbf{x}^0$ we

define the iterations

$$\begin{cases} a_{ii}y_i^{k+1} = \omega(b_i - \sum_{j<i} a_{ij}y_j^{k+1} - \sum_{j>i} a_{ij}x_j^k) + (1-\omega)a_{ii}x_i^k & for\ i = 1, ..., n \\ a_{ii}x_i^{k+1} = \omega(b_i - \sum_{j<i} a_{ij}y_j^{k+1} - \sum_{j>i} a_{ij}x_j^{k+1}) + (1-\omega)a_{ii}y_i^{k+1} & for\ i = 1, .., n. \end{cases} \tag{5.8}$$

We assume that A is symmetric positive definite and that $a_{ii} = 1$. The S.S.O.R. method can be written in matrix form as

$$\begin{cases} \mathbf{y}^{k+1} = \omega(\mathbf{b} - L\mathbf{y}^{k+1} - U\mathbf{x}^k) + (1-\omega)\mathbf{x}^k \\ \mathbf{x}^{k+1} = \omega(\mathbf{b} - L\mathbf{y}^{k+1} - U\mathbf{x}^{k+1}) + (1-\omega)\mathbf{y}^{k+1} \end{cases} ,$$

hence we have

$$\mathbf{x}^{k+1} - \mathbf{x} = (I + \omega U)^{-1}((1-\omega)I - \omega L)(I + \omega L)^{-1}((1-\omega)I - \omega U)(\mathbf{x}^k - \mathbf{x})$$

The iteration matrix is

$$\_\omega = (I + \omega U)^{-1}((1-\omega)I - \omega L)(I + \omega L)^{-1}((1-\omega)I - \omega U) \ .$$

Since the matrices ($(1-\omega)I - \omega L$) and $(I + \omega L)$ have the same eigenspaces, they commute and we have

$$((1-\omega)I - \omega L)(I + \omega L)^{-1} = (I + \omega L)^{-1}((1-\omega)I - \omega L) \ .$$

This implies that the iteration matrix can be written in the form

$$\_\omega = (I + \omega U)^{-1}(I + \omega L)^{-1}((1-\omega)I - \omega L)((1-\omega)I - \omega U) \ . \tag{5.9}$$

Moreover we have

$$(I+\omega U)\ \_\omega(I+\omega U)^{-1} = (I+\omega L)^{-1}((1-\omega)I-\omega L)((1-\omega)I-\omega U)(I+\omega U)^{-1} \ .$$

We now remark that since $U^T = L$ we have

$$(I + \omega L)^{-1}((1-\omega)I - \omega L) = |((1-\omega)I - \omega U)(I + \omega U)^{-1}|^T \ ,$$

hece defining

$$G_\omega := (I + \omega L)^{-1}((1 - \omega L)$$

we have

$$(I + \omega U)\ \ \_\omega(I + \omega U)^{-1} = G_\omega G_\omega^T \ .$$

This relation prove that $\_\omega$ is similar to a symmetric positive semi definite matrix and hence its eigenvalues are real and non negative. Let us consider the eigenvalue problem

$$\_\omega \mathbf{z} = \lambda \mathbf{z} \ . \tag{5.10}$$

Using (5.9) we can write this system in the form

$$((1-\omega)I - \omega L)((1-\omega)I - \omega U)\mathbf{z} = \lambda(I+\omega L)(I+\omega U)\mathbf{z} \ ,$$

hence we obtain

$$\lambda = \frac{\mathbf{z}^T((1-\omega)I - \omega L)((1-\omega)I - \omega U)\mathbf{z}}{\mathbf{z}^T(I+\omega L)(I+\omega U)\mathbf{z}} \ =$$

$$= \ \frac{\mathbf{z}^T((1-\omega)^2 I - \omega(1-\omega)(L+U) + \omega^2 LU)\mathbf{z}}{\mathbf{z}^T(I+\omega L)(I+\omega U)\mathbf{z}} \ =$$

$$= \frac{\mathbf{z}^T(I + \omega(L+U) + \omega^2 LU + (\omega^2 - 2\omega)I + (\omega^2 - 2\omega)(L+U))\mathbf{z}}{\mathbf{z}^T(I+\omega L)(I+\omega U)\mathbf{z}}$$

This relation gives

$$\lambda = 1 + (\omega - 2)\omega \ \frac{\mathbf{z}^T A\mathbf{z}}{\mathbf{z}^T(I+\omega U)(I+\omega L)\mathbf{z}} \ .$$

Since

$$\lambda \geq 0 \ \ and \ \ \frac{\mathbf{z}^T A\mathbf{z}}{\mathbf{z}^T(I+\omega U)(I+\omega L)\mathbf{z}} \ > \ 0$$

we deduce that the eigenvalues of $\mathcal{L}_\omega$ are less than 1 iff $0 < \omega < 2$. We have proved the following

Theorem 5.2 - The eigenvalues of the iteration matrix of the S.S.O.R. method are real and non negative; moreover, the method converges iff ω is in the interval: $0 < \omega < 2$.

The choice of the optimal relaxation parameter $\tilde{\omega}$ for the S.S.O.R. method is a difficult task. D. Young has proved that if $\|L\|_2 \ \leq \ \frac{1}{2}$, then $\tilde{\omega}$ is given by

$$\tilde{\omega} \ = \ \frac{2}{1 + \sqrt{2(1-\rho^2(B_J))}} \ . \tag{5.10}$$

By plotting the spectral radius of $\mathcal{L}_\omega$ against ω one obtains the curve of fig. 11. Notice that $\rho(\mathcal{L}_\omega)$ is not very sensitive to an over (or under) estimate of $\tilde{\omega}$.

fig. 11

For the finite difference approximation to Poisson's equation with red/black ordering the condition $\|L\|_2 \leq \frac{1}{2}$ is not verified. In this case $\tilde{\omega} = 1$, hence there is no advantage in using the S.S.O.R. method for this ordering .

The condition $\|L\|_2 \leq \frac{1}{2}$ is however fulfilled for the natural ordering and here the value (5.10) of $\tilde{\omega}$ seems to work relatively well in practice.

6. Acceleration Schemes and Conjugate Gradient

We wish to introduce other techniques for accelerating convergence. So far, we have defined many methods of the form:

$$\mathbf{x}^{k+1} = B\mathbf{x}^k + M^{-1}\mathbf{b} \tag{6.1}$$

Remark that the exact solution $\mathbf{x}$ of Problem (1.1) verifies

$$\mathbf{x} = B\mathbf{x} + M^{-1}\mathbf{b} \ . \tag{6.2}$$

The convergence of the scheme (6.1) may be quite slow in many situations. In order to accelerate the convergence, we define a new process by taking suitable averages of the sequence $\mathbf{x}^l \quad l = 0, ..., k$:

$$\mathbf{y}^k = \sum_{l=0}^{k} a_{kl}\mathbf{x}^l \ , \quad \text{with} \quad \sum_{l=0}^{k} a_{kl} = 1 \ . \tag{6.3}$$

Note that if $a_{kk} = 1$ and $a_{kl} = 0 \ k \neq l$, then we are back to the original scheme.
We are interested in studying the error $\mathbf{y}^k - \mathbf{x}$. Recalling that $\mathbf{x}^l - \mathbf{x} = B^l(\mathbf{x}^0 - \mathbf{x})$, we see that

$$\mathbf{y}^k - \mathbf{x} = \sum_{l=0}^{k} a_{kl}B^l(\mathbf{x}^0 - \mathbf{x}) = p_k(B)(\mathbf{x}^0 - x) \tag{6.4}$$

As in (4.3), the error $\mathbf{y}^k - \mathbf{x}$ is obtained by applying a polynomial of B of degree k to the initial error $\mathbf{x}^0 - \mathbf{x}$. Such a polynomial $p_k(\lambda)$ takes the form

$$p_k(\lambda) = \sum_{l=0}^{k} a_{kl}\lambda^l \ , \quad \text{with } p_k(1) = 1 \ .$$

From (6.4), we deduce

$$\|\mathbf{y}^k - \mathbf{x}\| \ <= \ \|p_k(B)\|_2\mathbf{x}^0 - \mathbf{x}\| \ . \tag{6.5}$$

Now we assume B symmetric with eigenvalues $\lambda_1 \leq ... \leq \lambda_n$. Then

$$\|p_k(B)\|_2 = max \ _{\lambda_i} \ |p_k(\lambda_i)| \ .$$

Again, we seek the optimal polynomial $\hat{p}_k(\lambda)$, with $\hat{p}_k(1) \ = \ 1$, that minimizes the $max_{\lambda_1 \leq \lambda \leq \lambda_n}|p_k(\lambda)|$. This leads to the Chebyshev polynomials

again. We can use the three term recurrence formula that defines the Chebyshev polynomials:

$$\begin{aligned} T_0(z) = 1, T_1(z) = z \\ T_{k+1}(z) = 2zT_k(z) - T_{k-1}(z) \end{aligned} \tag{6.6}$$

For the sake of simplicity, the assumption $\mu := \lambda_1 = -\lambda_n$ will be introduced. Since the scheme (6.1) is assumed convergent, we have $|\mu| < 1$. The $p_k(\lambda)$'s take the form

$$p_k(\lambda) = \frac{T_k(\frac{\lambda}{\mu})}{T_k(\frac{1}{\mu})} \ , \tag{6.7}$$

in order to satisfy the constraint $p_k(1) = 1$.

The p_k's also satisfy a three term recurrence formula derived from (6.6) and (6.7):

$$T_{k+1}(\frac{1}{\mu})p_{k+1}(z) = \frac{2z}{\mu}T_k(\frac{1}{\mu})p_k(z) - T_{k-1}(\frac{1}{\mu})p_{k-1}(z) \ . \tag{6.8}$$

By exploiting (6.8), from (6.4), we obtain

$$\mathbf{y}^{k+1} - \mathbf{x} = p_{k+1}(B)(\mathbf{x}^0 - \mathbf{x}) = \frac{2T_k(\frac{1}{\mu})}{\mu T_{k+1}(\frac{1}{\mu})} \ B(\mathbf{y}^k - \mathbf{x}) - \tag{6.9}$$

$$- \ \frac{T_{k-1}(\frac{1}{\mu})}{T_{k+1}(\frac{1}{\mu})} \ (\mathbf{y}^{k-1} - \mathbf{x}) \ .$$

Now, by using (6.2), adding and subtracting $\mathbf{y}^{k-1}$, we have

$$\mathbf{y}^{k+1} - \mathbf{x} = \frac{2T_k(\frac{1}{\mu})}{\mu(T_{k+1}(\frac{1}{\mu}))} \ (B\mathbf{y}^k + M^{-1}\mathbf{b}) - \frac{T_{k-1}(\frac{1}{\mu}) + T_{k+1}(\frac{1}{\mu})}{T_{k+1}(\frac{1}{\mu})} \ \mathbf{y}^{k-1} \tag{6.10}$$

$$+ \ \frac{(T_{k-1}(\frac{1}{\mu}) - \frac{2}{\mu}T_k(\frac{1}{\mu})}{T_{k+1}(\frac{1}{\mu})} \ \mathbf{x} \ + \ \mathbf{y}^{k-1}$$

Then, by using again the recurrence (6.6), we obtain

$$\mathbf{y}^{k+1} = \omega_{k+1}(B\mathbf{y}^k + M^{-1}\mathbf{b} - \mathbf{y}^{k-1}) + \mathbf{y}^{k-1} \ , \tag{6.11}$$

where

$$\omega_{k+1} = \frac{2T_k(\frac{1}{\mu})}{\mu T_{k+1}(\frac{1}{\mu})} \tag{6.12}$$

From the splitting $A = M - N$, we have $B = I - M^{-1}A$. Therefore (6.11) becomes

$$\mathbf{y}^{k+1} = \omega_{k+1}(\mathbf{z}^k + \mathbf{y}^k - \mathbf{y}^{k-1}) + \mathbf{y}^{k-1} \tag{6.13}$$

with

$$M\mathbf{z}^k = \mathbf{r}^k = (\mathbf{b} - A\mathbf{y}^k) \ . \tag{6.14}$$

Moreover, by exploiting once more (6.6) we obtain

$$\omega_{k+1} = 1 + \frac{2T_{k-1}(\frac{1}{\mu})}{\mu T_k(\frac{1}{\mu})} \quad \frac{\mu}{2} \quad \frac{T_k(\frac{1}{\mu})}{T_{k+1}(\frac{1}{\mu})} \ ,$$

hence

$$\omega_{k+1} = \frac{1}{1 - \frac{\mu^2}{4}\omega_k} \ . \tag{6.15}$$

Summarizing, we obtained the Chebyshev *semi-iterative* algorithm associated with (6.1):

$$\left\{ \begin{array}{c} \mathbf{y}^0 \ given \ ; \\ \mathbf{y}^1 = \mathbf{z}^0 + \mathbf{y}^0 \ , \ with \ M\mathbf{z}^0 = \mathbf{b} - A\mathbf{y}^0 \\ \mathbf{y}^{k+1} = \omega_{k+1}(\mathbf{z}^k + \mathbf{y}^k - \mathbf{y}^{k-1}) + \mathbf{y}^{k-1} \ , \\ with \\ M\mathbf{z}^k = \mathbf{r}^k = \mathbf{b} - A\mathbf{y}^k \ and \\ \omega_{k+1} = \frac{1}{1-\frac{\mu^2}{4}\omega_k} \ , \omega_1 = 1 \end{array} \right. \tag{6.16}$$

Naturally, the relative error will be as discussed earlier (see (6.5)):

$$\frac{\|\mathbf{y}^k - \mathbf{x}\|}{\|\mathbf{y}^0 - \mathbf{x}\|} \ <= \ \frac{const.}{T_k(\frac{1}{\mu})} \ . \tag{6.17}$$

(The const. is equal to $\|Q\|_2\|Q^{-1}\|_2$ where Q consists of the eigenvectors of B).

Since $|\mu| < 1$, then $T_k(\frac{1}{\mu}) = \cosh(k \cosh^{-1} \frac{1}{\mu})$. Hence the error estimate (6.17) shows that the scheme (6.16) converges faster than the original scheme (6.1) and the convergence rate grows rapidly as μ decreases.

As in the Chebyshev scheme described in Section 4, the parameters ω_k play an important role and are sensitive to the value of μ. Here, however, the number of iterations need not be prescribed a priori. Note that $1 < \omega_k < 2$ and the sequence of ω_k's decreases. Moreover it is possible to prove that it converges to the optimal $\hat{\omega} = \frac{2}{1+\sqrt{1-\rho(B_J)^2}}$ of the SOR iterations (whenever

$\hat{\omega}$ is definable see Section 5). We can say that the semi-iterative scheme is similar to SOR, except that here ω_{k+1} is changed at each iteration.
Note that this method applies whenever the starting iteration matrix B has real eigenvalues. This is not the case for the SOR matrix and the SOR scheme cannot be coupled with this semi-iterative procedure. This is one of the motivation of the introduction of the SSOR algorithm.
Finally, let us see how once more Property (A) can be very helpful from the computational point of view. For instance, consider as starting algorithm (6.1) the Jacobi method. Then, with $A = \begin{bmatrix} I & F \\ F^t & I \end{bmatrix}$, consider the iteration matrix $B = \begin{bmatrix} 0 & -F \\ -F^T & 0 \end{bmatrix}$.
Splitting $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$ in such a way that $A\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 & +F\mathbf{y}_2 \\ F^T\mathbf{y}_1 & +\mathbf{y}_2 \end{bmatrix}$, the algorithm (6.16) reads as follows:

$$\begin{bmatrix} \mathbf{y}_1^{k+1} \\ \mathbf{y}_2^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^{k-1} \\ \mathbf{y}_2^{k-1} \end{bmatrix} + \omega_{k+1} \begin{bmatrix} \mathbf{b}_1 - F\mathbf{y}_2^k & -\mathbf{y}_1^{k-1} \\ \mathbf{b}_2 - F^T\mathbf{y}_1^k & -\mathbf{y}_2^{k-1} \end{bmatrix} \tag{6.18}$$

Formula (6.18) shows that for any k, $\mathbf{y}_1^{2k+1}$ depends only on $\mathbf{y}_1^{2k-1}$ and $\mathbf{y}_2^{2k}$, while $\mathbf{y}_2^{2k}$ depends only on $\mathbf{y}_1^{2k-1}$ and $\mathbf{y}_2^{2k-2}$. So it is sufficcient to compute $\mathbf{y}_1^{2k+1}$ and $\mathbf{y}_2^{2k}$.
In the case of a matrix with Property (A) the Chebyshev semi-iterative algorithm converges as rapidly as the SOR algorithm.

Two key points are to be stressed here.

First, a good estimate of the eigenvalues is needed and since, as already pointed out, this information may not be available, methods that do not make use of it are desirable.

Second, in the above description we assumed that the problem $M\mathbf{z} = \mathbf{r}$ in (6.16) is solved exactly. We can ask if this step can be skipped and only the approximate solution provided. In many cases, such as in domain decomposition algorithms, the exact solution of $M\mathbf{z} = \mathbf{r}$ is too costly.

Now, let us consider a class of algorithms that overcome the first point. Consider a scheme formally similar to algorithm (6.16):

$$\left\{ \begin{array}{c} \mathbf{x}_1^0 \ \ assigned \\ x^1 = \mathbf{x}^0 + \alpha_0 \mathbf{z}^0 \ , \ \ Mz^0 = \mathbf{b} - A\mathbf{x}^0 \\ \mathbf{x}^{k+1} = \mathbf{x}^{k-1} + \omega_{k+1}(\alpha_k \mathbf{z}^k + \mathbf{x}^k - \mathbf{x}^{k-1}) \ , \end{array} \right. \tag{6.19}$$

with

$$M\mathbf{z}^k = \mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k \ . \tag{6.20}$$

Here the matrix M is supposed to be symmetric positive definite.
The parameters ω_{k+1} , α_k will be chosen dynamically in order to enforce the M-orthogonality among all the $\mathbf{z}^k$'s. Since M is positive definite this imply that the $\mathbf{z}^k$'s are linearly independent and after at most n steps (n being the rank of M) the residual must be zero. Of course, this property of finite termination holds in the absence of rounding errors. It turns out that it is enough to impose the two relations:

$$(\mathbf{z}^k, M\mathbf{z}^{k+1}) = 0 \tag{6.21}$$

and

$$(\mathbf{z}^{k-1}, M\mathbf{z}^{k+1}) = 0 \ , \tag{6.22}$$

A short production argument on k will show that (6.21) and (6.22) imply:

$$(\mathbf{z}^j, M\mathbf{z}^l) = 0 \ \ \forall j, l = 0, ..., k+1 \ j \neq l \tag{6.23}$$

In fact, for k=2 the statement (6.23) is obviously satisfied. Suppose (6.23) verified for $j, l = 0, ..., k$.
Due to (6.21) and (6.22), we have only to show that

$$(\mathbf{z}^{k+1}, M\mathbf{z}^j) = 0 \quad \forall j \leq k-2 \ .$$

From (6.19) and (6.20) we obtain

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \omega_{k+1}(\alpha_k A\mathbf{z}^k + \mathbf{r}^{k-1} - \mathbf{r}^k),$$

and hence

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \omega_{k+1}(\alpha_k M^{-1} A\mathbf{z}^k + \mathbf{z}^{k-1} - \mathbf{z}^k) \tag{6.24}$$

Then, by using the induction assumption we have:

$$(\mathbf{z}^{k+1}, M\mathbf{z}^j) = -\omega_k\alpha_k(A\mathbf{z}^k, \mathbf{z}^j) = -\omega_k\alpha_k(\mathbf{z}^k, A\mathbf{z}^j), j \leq k-2\ . \tag{6.25}$$

From (6.24) we know that $A\mathbf{z}^j$ is a linear combination of $M\mathbf{z}^{j-1}$, $M\mathbf{z}^j$ and $M\mathbf{z}^{j+1}$, therefore by applying again the induction hypothesis to (6.25), we obtain the thesis.

Now, let us determine the α_k's and ω_k's that enforce (6.21) and (6.22) respectively.
By using the expression (6.24) of $\mathbf{z}^{k+1}$ we have:

$$\begin{aligned}(\mathbf{z}^k, M\mathbf{z}^{k+1}) = (\mathbf{z}^k, M\mathbf{z}^{k-1}) - \omega_{k+1}|\alpha_k(\mathbf{z}^k, Az^k)+ \\ +(\mathbf{z}^k, M\mathbf{z}^{k-1}) - (\mathbf{z}^k, M\mathbf{z}^k)|\end{aligned} \tag{6.26}$$

So (6.21) is fulfilled when

$$\alpha_k = \frac{(\mathbf{z}^k, M\mathbf{z}^k)}{(\mathbf{z}^k, A\mathbf{z}^k)}\ , \tag{6.27}$$

since $(\mathbf{z}^k, M\mathbf{z}^{k-1}) = 0$, thanks to (6.23). Analogously, we see that (6.22) is satisfied when

$$\begin{aligned}\omega_{k+1} = \frac{(\mathbf{z}^{k-1}, M\mathbf{z}^{k-1})}{\alpha_k(\mathbf{z}^{k-1}, A\mathbf{z}^k) + (\mathbf{z}^{k-1}, M\mathbf{z}^{k-1})} \ = \\ = \frac{1}{1 - \alpha_k \frac{(\mathbf{z}^{k-1}, A\mathbf{z}^k)}{(\mathbf{z}^{k-1}, A\mathbf{z}^{k-1})}}\ .\end{aligned} \tag{6.28}$$

Now, we give a recursive formula for ω_{k+1} which can be more efficiently used. Recall (6.23) and the relationship:

$$M\mathbf{z}^k = M\mathbf{z}^{k-2} - \omega_k(\alpha_{k-1}A\mathbf{z}^{k-1} + M\mathbf{z}^{k-2} - M\mathbf{z}^{k-1})\ .$$

Then we obtain

$$(\mathbf{z}^{k-1}, A\mathbf{z}^k) = (\mathbf{z}^k, A\mathbf{z}^{k-1}) = -\frac{(\mathbf{z}^k, M\mathbf{z}^k)}{\omega_k\alpha_{k-1}},$$

hence (6.28) provides:

$$\omega_{k+1} = (1 - \frac{\alpha_k}{\alpha_{k-1}}\ \frac{(\mathbf{z}^k, M\mathbf{z}^k)}{(\mathbf{z}^{k-1}, M\mathbf{z}^{k-1})}\ \frac{1}{\omega_k})^{-1} \tag{6.29}$$

Finally we can state the following algorithm:

$$\begin{array}{c} \mathbf{x}^0 \ \ given \\ \mathbf{x}^1 = \mathbf{x}^0 + \alpha_0 \mathbf{z}^0 \ , \\ M\mathbf{z}^0 = r^0 = \mathbf{b} - A\mathbf{x}^0 \\ \alpha_0 = \frac{(\mathbf{z}^0, M\mathbf{z}^0)}{(\mathbf{z}^0, A\mathbf{z}^0)} \\ \omega_1 = 1 \ ; \\ for \ k = 1, 2, ..., n-1. \end{array} \tag{6.30}$$

$$\begin{array}{c} M\mathbf{z}^k = \mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k \\ \alpha_k = \frac{(\mathbf{z}^k, M\mathbf{z}^k)}{(\mathbf{z}^k, A\mathbf{z}^k)} \ , \\ \omega_{k+1} = \quad 1 - \frac{\alpha_k}{\alpha_{k-1}} \ \frac{(\mathbf{z}^k, M\mathbf{z}^k)}{(\mathbf{z}^{k-1}, M\mathbf{z}^{k-1})} \ \frac{1}{\omega_k} \quad ^{-1} \ . \\ \mathbf{x}^{k+1} = \mathbf{x}^{k-1} + \omega_{k+1}(\alpha_k \mathbf{z}^k + \mathbf{x}^k - \mathbf{x}^{k-1}) \end{array}$$

This method is called *Preconditioned Conjugate Gradient* and the symmetric positive definite matrix M is called *Preconditioner.*
Note that if $M = I$ and $\omega_{k+1} = 1$ the scheme coincides with the Steepest Descent algorithm described at the end of Section 4. Actually, if $M \neq I$ it is the preconditioned version of that scheme.

So far, we have based our discussion about the Conjugate Gradient algorithm on orthogonality and it does make it easier to derive the formulas. The usual presentation is somewhat different and starts from the minimization of a quadratic functional. We shall outline it and in order to simplify the formulas, we shall set $M = I$. We know that solving the system (1.1) with A symmetric positive definite is equivalent to minimizing the functional

$$\phi(x) = \frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (\mathbf{b}, \mathbf{x}). \tag{6.31}$$

We successively minimize ϕ along a set of directions $\mathbf{p}^1, ..., \mathbf{p}^k$. Namely, we choose γ_k that minimizes $\phi(\mathbf{x}^k + \gamma \mathbf{p}^k)$, that is

$$\gamma_k = \frac{(\mathbf{p}^k, \mathbf{r}^k)}{(\mathbf{p}^k, A\mathbf{p}^k)} \ ,$$

where $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$.
With this choice, if $\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_k \mathbf{p}^k$, it can be shown that

$$\phi(\mathbf{x}^{k+1}) = \phi(\mathbf{x}^k) - \frac{1}{2} \ \frac{(\mathbf{p}^k, \mathbf{r}^k)^2}{(\mathbf{p}^k, A\mathbf{p}^k)} \ . \tag{6.32}$$

We are left with the problem of choosing the $\mathbf{p}^k$'s. We do so by imposing:

$$(\mathbf{p}^i, A\mathbf{p}^j) = 0 \quad for\ i \neq j \ .$$

This choice leads to the following scheme:

$$\begin{gathered} \mathbf{x}^0 \ \ given \\ \mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0 \\ \mathbf{p}^0 = \mathbf{r}^0 \ ; \\ for\ k = 1, ..., n-1 \ . \end{gathered} \tag{6.33}$$

$$\begin{gathered} \beta_k = -\frac{(\mathbf{p}^{k-1}, A\mathbf{r}^k)}{(\mathbf{p}^{k-1}, A\mathbf{p}^{k-1})} \\ \mathbf{p}^k = \mathbf{r}^k + \beta_k \mathbf{p}^{k-1} \\ \gamma_k = \frac{(\mathbf{p}^k, \mathbf{r}^k)}{(\mathbf{p}^k, A\mathbf{p}^k)} \\ \mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_k \mathbf{p}^k \\ \mathbf{r}^{k+1} = \mathbf{r}^k - \gamma_k A\mathbf{p}^k \end{gathered}$$

We return now to the first formulation (6.30) and we derive some properties of the method.
Since $A = M - N$ and we have derived (6.24) we obtain

$$\mathbf{z}^{k+1} = \mathbf{z}^{k-1} - \omega_{k+1}(\alpha_k(I - M^{-1}N)\mathbf{z}^k - \mathbf{z}^k + \mathbf{z}^{k-1}) \ ,$$

which implies (by induction argument):

$$\mathbf{z}^{k+1} = (I - KP_k(K))\mathbf{z}^0 \ , \tag{6.34}$$

where $K = M^{-1}A$ and $P_k(K)$ is a polynomial of degree k in K.
Furthermore, together with (6.34), a simple induction argument shows:

$$\mathbf{x}^{k+1} = \mathbf{x}^0 - P_k(K)\mathbf{z}^0 \ . \tag{6.35}$$

Now, we introduce the error norm:

$$\begin{gathered} E(\mathbf{x}^{k+1)} = (A(\mathbf{x}^{k+1} - \mathbf{x}), x^{k+1} - \mathbf{x}) = \\ = (\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1}). \end{gathered} \tag{6.36}$$

Suppose we have another process which generates a polynomial $Q_k(K)$ so that

$$\begin{gathered} \bar{\mathbf{x}}^{k+1} = \bar{\mathbf{x}}^0 + Q_k(K)\bar{\mathbf{z}}^0 \ , \\ \bar{\mathbf{x}}^0 = \mathbf{x}^0 \ , \\ \bar{\mathbf{z}}^{k+1} = (I - KQ_k(K))\bar{\mathbf{z}}^0 \ . \end{gathered} \tag{6.37}$$

We can state the following theorem:
Theorem 6.1 - Amongst all methods of the form (6.37), the Preconditioned Conjugate Gradient method generates the *unique* polynomial which minimizes the error $E(\mathbf{x}^k)$ defined by (6.36).
The proof is basically algebraic and is given for instance in | |.

Note that (6.35) shows that the error in the Conjugate Gradient scheme takes into account the distribution of initial error and one may take advantage of that property.
The C.G. has been constructed in order to achieve termination of the algorithm in at most n steps if exact arithmetic operations are performed. By exploiting Theorem 6.1, in some cases we can be more precise. Since $K = M^{-1}A$ is a symmetric positive definite matrix, then it can be diagonalized as

$$K = QLAMBDAQ^{-1} \ .$$

Suppose now K has distinct eigenvalues $\lambda_1, ..., \lambda_d$. Then the minimal polynomial is

$$\Pi_{i=1}^{d}(1 - \lambda/\lambda_i)$$

So we have

$$\Pi_{i=1}^{d}(I - \frac{1}{\lambda_i}K) = 0 \ . \tag{6.38}$$

By taking $Q_k(K)$ in (6.37) such that $I - KQ_k(K) = \Pi_{i=1}^{k}(I - \frac{1}{\lambda_i}K)$ $k \leq d$, Theorem 6.1 and (6.38) imply $Q_d = P_d$, where P_d is defined by (6.34). Using again (6.38), we obtain $\mathbf{z}^d = 0$, and hence $\mathbf{r}^d = 0$. In other words, the Conjugate Gradient method must converge in at most d iterations if K has d distinct eigenvalues.
This property can be quite useful in practice, since in some significant situation we are able to build up operators with d distinct eigenvalues.
For instance, let us consider again the 5-point finite difference approximation to Poisson's equation on a T-shaped domain (see section 2).

insert figure

The domain is decoupled in two rectangles R_1 , R_2, where R_1 includes the interface I. With obvious notation, the matrix A appears to be:

$$A = \begin{bmatrix} A_1 & . & . & . \\ & & J & \\ & J^T & & \\ & & & A_2 \end{bmatrix} ,$$

where J is a $p \times q$ matrix, with $p < q$ (see notation of Section 2). So, if we choose

$$M = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} ,$$

then, the corresponding N is

$$N = \begin{bmatrix} 0 & . \\ & -J \\ -J^T & \\ & 0 \end{bmatrix}$$

and the rank of N is at most 2p. Recalling that $K = M^{-1}A = I - M^{-1}N$, K has at most $2p+1$ distinct eigenvalues and the Conjugate Gradient converges in at most $2p + 1$ iterations.

So far, we have consider the Conjugate Gradient method as a direct method, but due to roundoff errors it is usually regarded as an iterative scheme. So the question about the rate of convergence is an important question. The result of optimality of Theorem 6.1 allows to derive the following error estimate for the $E(\mathbf{x}^k)$ defined in (6.36):

$$\frac{E(\mathbf{x}^k)}{E(\mathbf{x}^0)} \leq 4(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^{2k} , \tag{6.39}$$

where $\kappa = \lambda_{max}(K)/\lambda_{min}(K)$ is the condition number of the preconditioned matrix $K = M^{-1}A$.

It can be shown the C.G. algorithm is strictly related to the Lanczos process for the orthonormalization of a matrix. In fact, by using (6.24) and $K = M^{-1}A$, we obtain

$$\omega_{k+1}\alpha_k K\mathbf{z}^k = (1-\omega_{k+1})\mathbf{z}^{k-1} + \omega_{k+1}\mathbf{z}^k - \mathbf{z}^{k+1} \ ,$$

and hence

$$K\mathbf{z}^k = b_{k-1}\mathbf{z}^{k-1} + a_k\mathbf{z}^k + c_k\mathbf{z}^{k+1} \ , \tag{6.40}$$

where $b_{k-1} = \frac{1-\omega_{k+1}}{\omega_{k+1}\alpha_k}$, $a_k = 1/\alpha_k$ *and* $c_k = -\frac{1}{\omega_{k+1}\alpha_k}$. Denote by Z the matrix whose k-column is the vectors $\mathbf{z}^k$, that is

$$Z = |\mathbf{z}^0, ..., \mathbf{z}^{n-1}| \ ,$$

and by J the following matrix

$$J = \begin{bmatrix} a_0 & c_0 & . & . \\ b_0 & a_1 & c_1 & \\ & & & c_{n-2} \\ & & b_{n-2} & a_{n-1} \end{bmatrix}$$

Then, the relationship (6.40) takes the compact form:

$$KZ = ZJ \tag{6.41}$$

Now, recalling that $(\mathbf{z}^i, M\mathbf{z}^j) = 0 \ \forall i \neq j$, and assuming $\|z_i\| \neq 0 \ i = 0, ..., n-1$, we obtain that Z is not singular.
Them, (6.41) implies the following:

$$Z^{-1}KZ = J \ . \tag{6.42}$$

So the eigenvalues of J are the same than the eigenvalues of K. Due to the simple structure of J, we are able to compute the eigenvalues of K. The main cost of the process of course comes from the computations of the coefficients. The Kaniel-Paige theory on the Lanczos procedure indicates that the eigenvalues of

$$J_l = \begin{bmatrix} a_0 & c_0 & . \\ b_0 & a_1 & c_1 \\ & & \\ & b_{l-1} & a_l \end{bmatrix}$$

are good approximations to those of the original matrix J, so that a truncated process may be sufficient.

Finally, we stress two key points to be kept in mind in order to improve the performances of the C.G. algorithm. First, the goal is minimizing the condition number $\kappa(M^{-1}A)$, subjected to minimize the amount of work. Second, a good distribution of the eigenvalues is to be seeked. Let us see some possible choices of preconditioners.

1. $M = D$, that is the underlying iterative scheme is the Jacobi method. It has been shown by Forsythe-Strauss that if A has Property (A), D is the best choice of preconditioner amongst all the diagonal matrices.

2. $M = (D + \omega L)D^{-1}(D + \omega U)$. This choice leads to acceleration of the SSOR scheme. Note that C.G. cannot be coupled with the SOR iterations, since the corresponding matrix $M = (D + \omega L)$ is not symmetric.

3. Incomplete Cholesky Factorization. The matrix A is splitted as $A = LL^T + R$ (with the usual notation, $M = LL^T$ and $N = -R$), where L is a lower-triangular matrix and R is some sort of remainder. Note that if the Cholesky factorization is performed, then $R = 0$. Different choices of L can be made, all of them trying to preserve as much as possible of the structure of the original matrix, in particular some sparsity structure. Meijerink and Vandervost | | proposed a matrix L with the same band structure of A, which leads to a condition number $\kappa(M^{-1}A)$ of the same order of the condition number $\kappa(A)$, but improves the distribution of the eigenvalues. Concus, Golub and Meurant proposed a matrix L with the same block-structure as A.

If R is searched such that $R\mathbf{e} = \mathbf{0}$, with $\mathbf{e} = (1, 1, ..., 1)$, then the associated incomplete factorization produces a condition number $\kappa(M^{-1}A)$ with order of magnitude half of the order of the condition number $\kappa(A)$. (see | |).

4. Let the matrix A be associated with finite difference approximation of the biharmonic problem on a rectangular domain.

Bjpistad described the biharmonic matrix approximation which behaves like the square of the laplacian matrix plus a low rank modification near the boundary.

The associated splitting $A = M - N$ leads to a very fast C.G. scheme, thanks to the low rank of N. Moreover, Fast Poisson's Solvers are used in order to invert M. $0(N^2 logN)$ operations are needed in order to solve the system.

We can also take advantage of matrices which have Property (A). As for the Chebyshev semi-iterative procedure, we can compute only $\begin{matrix}\mathbf{x}_1^{2k-1}\\ \mathbf{x}_2^{2k}\end{matrix}$ (see (6.18)).
It is clear that the new computer architectures will affect the choice of M.

7. Domain Decomposition Methods

In this section we overview some problems arising from the domain decomposition techniques for the solution of partial differential equations in irregular regions. We consider a model problem, that is the finite difference approximation to Poisson's equation on a T-shaped domain. However the theory applies in more general situations. For instance, in combustion problems one has to solve time dependent equations in domains of the type of fig. 12 (see | |).

fig. 12

Here the boundary r is time dependent and a domain decomposition method is suitable. Another case where domain decomposition is useful is when the mesh is not uniform (see fig. 13).

fig. 13

For instance, in reservoir simulation problems in petroleum engineering the domain is often decomposed as in fig. 13, where the circles represent some wells. (see | |). The domain decomposition methods are also exten-

sively used for fluid dynamics computation, see e.g. | |.
We now come back to the finite difference approximation to Poisson's equation on a T-shaped domain.

fig. 14

We decompose the domain in three subdomains (see fig. 14). The third subdomain contains the interface points between the first and second one. We have to solve a linear system whose matrix is of the form

$$A = \begin{bmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ B_1^T & B_2^T & Q \end{bmatrix} . \tag{7.1}$$

The matrices A_1 and A_2 represent the Poisson's operator in the domains 1 and 2 respectively, B_1 represents the relations between the unknowns of the domains 1 and 3, while B_2 represents the relations between the unknowns of the domains 2 and 3. Finally, Q is built in order to impose the Poisson's equation on the points of the domain 3.

We denote by $\tilde{A}_2$ the matrix which represents the Poisson's equation on the union of the domains 2 and 3. Splitting A in the form $M - N$, two natural choices of M are the following.

$$M_1 = \begin{bmatrix} A_1 & 0 \\ 0 & \tilde{A}_2 \end{bmatrix} \tag{7.2}$$

$$M_2 = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & Q \end{bmatrix} \qquad \begin{bmatrix} A_1 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & A_2 \end{bmatrix} . \tag{7.3}$$

If $N_1 = M_1 - A$ and $N_2 = M_2 - A$, we have $N_2 \geq N_1$ and, since $A^{-1} \geq 0$, theorem 1.2 gives $\rho(M_1^{-1}N_1) \leq \rho(M_2^{-1}N_2)$. Hence, from the spectral radius

point of view, M_1 is the better choice. However, this is not the only criterium for choosing the preconditioner M. Let us consider the more general case:

$$A = \begin{bmatrix} A_1 & & & B_1 \\ & \ddots & & \\ & & A_r & B_r \\ B_1^T & & B_r^T & Q \end{bmatrix}, \tag{7.4}$$

this is the form of the matrix A when r subdomains are considered. A possible choice for M is

$$M = \begin{bmatrix} A_1 & . & . & . \\ & & & \\ & & A_r & Q \end{bmatrix} \quad . \tag{7.5}$$

Note that the system $M\mathbf{z}^k = \mathbf{r}^k$ of the Conjugate Gradient method can be solved parallelizing on the blocks of M. If the blocks $B_i, i = 1, ..., r$, have p columns, the CG method converges in $2p+1$ iterations, but since M has the block Property (A) the convergence is actually achieved in $\frac{2p+1}{2}$ iterations.

Another idea for solving the system $A\mathbf{x} = \mathbf{b}$, where A is given in (7.4), is the following.
Let $\mathbf{b}^T = (\mathbf{c}_i^T, \mathbf{d}^T)$ and $\mathbf{x}^T = (\mathbf{x}_i, \xi)$ for $i = 1, ..., r$ and write the linear system in the form

$$\begin{cases} A_i\mathbf{x}_i + B_i\xi_i = \mathbf{c}_i \quad i = 1, ..., r \\ \sum_{i=1}^r B_i^T\mathbf{x}_i + Q\xi = \mathbf{d} \ . \end{cases} \tag{7.6}$$

We obtain

$$\mathbf{x}_i = A_i^{-1}\mathbf{c}_i - A_i^{-1}B_i\xi \quad i = 1, ..., r$$

hence

$$\sum_{i=1}^r B_i^T(A_i^{-1}\mathbf{c}_i - A_i^{-1}B_i\xi) + Q\xi = \mathbf{d}$$

and

$$Q\xi = \mathbf{d} - \sum_{i=1}^r B_i^T A_i^{-1}(\mathbf{c}_i - B_i\xi) \ . \tag{7.7}$$

Of course, we would never form the vectors $B_i^T A_i^{-1}(\mathbf{c}_i - B_i\xi)$ but rather solve in parallel

$$A_i\eta_i = (\mathbf{c}_i - B_i\xi) \quad i = 1, .., r \ .$$

One can solve (7.7) iteratively, that is find a sequence ξ^k through

$$Q\xi^{k+1} = \mathbf{d} - \sum_{i=1}^r B_i^T A_i^{-1}(\mathbf{c}_i - B_i\xi^k) \ ,$$

and use the Conjugate Gradient method with the preconditioner $M = Q$. This method gives convergence in p iterations, that is the same number of iteration as for the choice of M given in (7.5). We remark that if we have some insight with our problem, we might choose some other preconditioners. For Poisson's equation, we can approximate the Green's function by looking

at the differential equation in semi-infinite strips | |.
Many authors have done extensive works using domain decomposition techniques, we refer for instance to Dryja | | Bramble | |, Pasciak | |, Widlund | |.

When the Conjugate Gradient method is used, in particular in the framework of a domain decomposition technique, at each stage one must solve the system

$$M\mathbf{z}^k = \mathbf{r}^k \ . \tag{7.8}$$

In many situations this system is solved approximately, that is a vector $\bar{\mathbf{z}}^k$ is obtained from

$$M\bar{\mathbf{z}}^k = \mathbf{r}^k + \mathbf{q}^k$$

where

$$\|\mathbf{q}^k\|/\|\mathbf{r}^k\| \leq \delta$$

and δ is a prescribed tolerance. For instance (7.8) is solved with an iterative method and the iterations are stopped when the residual (which is the vector $\mathbf{q}^k$) is sufficiently small. The vector $\mathbf{q}^k$ can be also regarded as the effect of roundoff errors in solving (7.8). The basic algorithm is now written as

$$\mathbf{x}^{k+1} = \mathbf{x}^{k-1} + \omega_{k+1}(\alpha_u \bar{\mathbf{z}}^k + \mathbf{x}^k - \mathbf{x}^{k-1}) \ .$$

Unfortunately we do not have an analysis of this procedure when the Conjugate Gradient method is used. It is possible to analyse the procedure when the Chebyshev or Richardson method is used (see | |). (The Richardson 2nd order method corresponds to setting $\omega_k = \omega = \frac{2}{2+\sqrt{1-\mu^2}}$ for all k). For the Richardson method, the rate of convergence is dependent upon

$$\tilde{\rho} = \rho + \frac{\omega\epsilon}{2} + (\frac{\rho\omega\epsilon}{2} + \frac{\omega^2\epsilon^2}{4})^{\frac{1}{2}} \ ,$$

where ρ is the spectral radius of the exact iteration matrix, ϵ is equal to δ times various parameters which depend on the problem under consideration and ω is the acceleration parameter. This analysis is closely connected to the roundoff analysis of procedures.

8. Remarks on the non-symmetric case

In this section we consider the case of a matrix A which is not symmetric. We shall suppose that the symmetric part of A is positive definite, that is

$$(\mathbf{x}, A\mathbf{x}) > 0$$

for every $\mathbf{x} \neq \mathbf{0}$.
This situation is not unusual, it occurs for instance in solving the problem

$$-\Delta u + \sigma u_x = f \ .$$

Let us split A as $A = M - N$ where

$$M = \frac{A + A^T}{2} \quad and \quad N = -\frac{A - A^T}{2} \ .$$

If, for instance, the SOR method is applied to a matrix A of the form

$$A = \begin{bmatrix} I & S \\ -S^T & I \end{bmatrix} \ ,$$

then $M = I$ and it turns out that we have convergence for $0 < \omega < \tilde{\omega}$ where

$$\tilde{\omega} = \frac{2}{1 + \|S\|_2} \ ,$$

the optimal choice of the relaxation parameter being

$$\hat{\omega} = \frac{2}{1 + \sqrt{1 + \|S\|_2^2}} \ .$$

If $\|S\|_2$ is big, the convergence is very slow. In this simple case it is possible to overcome the difficulties using a cyclic reduction technique. More precisely we write the system in the form

$$\begin{bmatrix} I & S \\ -S^T & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

and compute $\mathbf{y}$ solving

$$(I + S^T S)\mathbf{y} = (I + S^T)\mathbf{b} \ .$$

Since $I + S^T S$ is symmetric positive definite, it is convenient to use an iterative procedure for this system

The Conjugate Gradient method can be generalized to the non-synmmetric case using $M = \frac{A+A^T}{2}$. The basic iteration is

$$\mathbf{x}^{k+1} = \mathbf{x}^{k-1} + \omega_{k+1}(\mathbf{z}^k + \mathbf{x}^k - \mathbf{x}^{k-1}) \ , \quad (\alpha_k = 1) \ ,$$

where

$$\omega_{k+1} = (1 + \frac{(\mathbf{z}^k, M\mathbf{z}^k)}{(\mathbf{z}^{k-1}, M\mathbf{z}^{k-1})} \cdot \frac{1}{\omega_k})^{-1} \ .$$

$\mathbf{z}^k$ is obtained solving $M\mathbf{z}^k = r^k$ (where $\mathbf{r}^k$ is the residual), but it is often better to solve the approximate system $M\mathbf{z}^k = \mathbf{r}^k + \mathbf{q}^k$ (see section 7).

If the eigenvalues λ_i of $M^{-1}N$ are in an ellipse in the complex plane and $Re\lambda_i > 0$, the Chebyshev method can be generalized (see Mantenffel | |); it is not known how to use the method when this hypothesis is not fulfilled.
Another idea for the solution of $A\mathbf{x} = \mathbf{b}$ when A is not symmetric is to solve the normal equations

$$A^T A\mathbf{x} = A^T\mathbf{b} \ . \tag{8.1}$$

However a better strategy is preconditioning before solving (8.1), that is to find a preconditioner C and solve the system

$$A^T C^T C A\mathbf{x} = A^T C^T \mathbf{b}$$

using an iterative procedure.

The case when the matrix is symmetric but only positive semidefinite occours, for instance, in constrained minimization of quadratic forms. In this case the following linear system must be solved

$$\begin{bmatrix} A & C \\ -C^T & 0 \end{bmatrix} \qquad \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \quad , \tag{8.2}$$

where C represents the constraints and λ the Lagrange multipliers. An idea to solve (8.2) is to solve the systems

$$\begin{bmatrix} A & C \\ -C^T & \epsilon I \end{bmatrix} \qquad \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \qquad \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \tag{8.3}$$

for a variety of ϵ's and then to use an extrapolation method. In this case setting

$$M_\epsilon = \quad \begin{bmatrix} A & 0 \\ 0 & \epsilon I \end{bmatrix}$$

it is possible to apply the Conjugate Gradient method to solve (8.3).