A Graduate Seminar Report

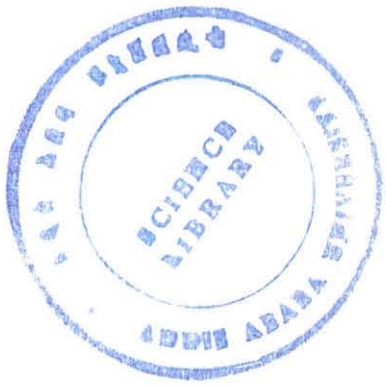
On

Iterative Methods For The Solution Of Large Systems Of Linear Equations

(Submitted in partial fulfillment of M.Sc. degree in Mathematics)

By

Genanew Gofe



Advisor

Prof.Dr. S.N. Murthy

School of Graduate studies
Addis Ababa University

June, 2004

Addis Ababa

Preface

Systems of linear equations arise in large number of areas, both directly in modeling physical situation and indirectly in the numerical solution of some mathematical models.

These applications occur in virtually all areas of the physical, biological and social sciences. Because of the wide spread importance of linear systems, much researcher has been devoted to their numerical solutions. Hence, the main objective of this report is to discuss some numerical methods in finding the numerical solution of a large systems of linear equations, which arises in the application of difference methods or finite method to approximate the solution of boundary value problem in partial differential equation.

In this report a three chapter discussion has been made.

The first chapter deals with some mathematical preliminary, iterative developments and the core theorem of this report (convergence theorem). The second chapters consists some numerical methods such as Jacobi, Gauss-Seidel, Relaxation and Conjugate gradients Method.

In last chapter we discuss the application of difference method, Block iterative method to practical problem.

I wish to express my deepest indebtedness to my advisor, Professor Dr.S.N. MURTHY, for the enormous help rendered to me in imparting the basic mathematical knowledge necessary for the realization of this seminar.

Many thanks are also due to my family, friends and classmates for their special assistance and encouragement.

Genanew Gofe

Contents page

Chapter – one

ITERATIVE METHOD

1.1. Introduction

Many problems in practice require the solution of large system of linear equation

$$\mathbf{Ax = b}$$

Where A=matrix (sparse matrix)

Systems of this type arise frequently in numerical solution of boundary-value problems and partial differential equations.

The usual-elimination methods cannot normally be applied here, since without special precaution they tend to lead to the formulation of more less dense intermediate matrices, making the methods of operation necessary for the solution much too large, even for present computer not to fit in to the usually available computer memory. For these reasons, researchers have long since move to iterative methods for solving such system of equations.

Further more, because they are economical in their use of computer memory, iterative methods are a particular advantageous for the very large system of linear equations.

Iterative methods consists in guessing an initial approximation vector

$X^0=(x_1^0, x_2^0, \dots, x_n^0)^T$, then one generate a sequence of vector.

$X^0 \rightarrow X^1 \rightarrow X^2 \rightarrow \dots$.which converges toward the desired solution X. In practice, iterative methods are seldom used for solving linear system of small dimension, since the time required for sufficient accuracy exceeds that required for direct methods such as Gauss-elimination. So, in general for large systems with a high percentage of zero entries, however, these methods are efficient in terms of computer storage and time requirements.

1.2. Norms of Matrices and Vectors

Before considering iterative methods for solving linear system, it is necessary to determine a method for quantitatively measuring the distance between vectors in K^n and matrixes in K^{nxn}, the set of all column vectors, in order to determine where the sequence of vectors, which results from an iterative method, converges to the solution of system.

Note: Here K^n represents either R^n or C^n

Definition 1: A vector norm (usually in R^n), the collection of all n-dimension column vectors with real components, is a function, $\| \cdot \|$, from K^n in to K with the following properties

i) $\|X\| \geq 0$ for all $x \in K^n$

ii) $\|X\| = 0$ if and only if $X=(0,\ldots, 0)^T=0$

iii) $\|\propto X\| = |\propto| \, \|X\|$ for all $\propto \in K$, $x \in K^n$

iv) $\|X+Y\| \leq \|X\|+\|Y\|$ for all $x,y \in K^n$.

For example the vector

$$X = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$, will generally be written $X =(x_1, x_2,\ldots,x_n)^T$

The $\| \ \|_2$ and $\| \ \|_\infty$ for the vector $X =(x_1,x_2,\ldots,x_n)^T$ are defined by

$$\|X\|_2 = \left(\sum_{i=1}^{n} x^2{}_i \right)^{1/2} \text{ and } \| \ \|_\infty = \max|x_i|$$

Definition 2: A sequence $\{x^{(k)}\}_{k=1}^{\infty}$ of vector in K^n is said to converge to x with respect to the norm $\| \cdot \|$, if given any $\varepsilon > 0$, there exists an integer $N(\varepsilon)$ such that $\|x^{(k)} - x\| < \varepsilon$ for all $k \geq N(\varepsilon)$.

Example: Let $x^{(k)} \varepsilon R^n$ be defined by

$$x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^T = (1,2+1/k,3/k^2,e^{-k}\sin k)^T$$

some typical members of the sequence are

$$x^{(1)} = \begin{bmatrix} 1 \\ 3 \\ 3 \\ \sin 1/e \end{bmatrix}, \quad x^{(10)} = \begin{bmatrix} 1 \\ 2.1 \\ 0.03 \\ \sin 10/e^{10} \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2.1 \\ 0.03 \\ -2.5x10^{-5} \end{bmatrix}$$

and

$$x^{(100)} = \begin{bmatrix} 1 \\ 2.01 \\ 3x10^{-4} \\ \sin 100/e^{100} \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2.01 \\ 3x10^{-4} \\ -1.88x10^{-44} \end{bmatrix}$$

It appears that the vector x given by $x = (1,2,0,0)^T$ is the limit of $\{x^{(k)}\}_{k=1}^{\infty}$, since all component of the sequence converges to the component of x.

To establish this fact note that $x^{(k)} - x = (0,1/k,3/k^2,e^{-k}\sin k)^T$. Expanding e^k in second degree Taylor polynomial about zero, we see that

$$e^k \geq 1 + k + \frac{1}{2}k^2 \geq \frac{1}{2}k^2$$

It follows that for $k \geq 3$

$$0 \leq |e^{-k}\sin k| \leq \frac{2}{k^2}|\sin k| \leq \frac{2}{k^2} \leq \frac{1}{k}$$

Hence, for $k \geq 3$

$$\| x^{(k)} - x \|_{\infty} = \max\{|0|, |\frac{1}{k}|, |\frac{3}{k^2}|, |e^{-k}\sin k|\} = \frac{1}{k}$$

given $\varepsilon > 0$, let N be any integer greater than both 3 and $1/\varepsilon$. If $k \geq N$, *then* $\| x^{(k)} - x \|_{\infty} = \frac{1}{k} \leq \frac{1}{N} < \in$ and $\{x^{(k)}\}_{k=1}^{\infty}$ converges to x.

Definition 3: A matrix norm on the set of all nxn matrices is a function, ||.|| defined on this set, satisfying for nxn matrices A and B and all real number $\propto$:

i) $\|A\| \geq 0$

ii) $\|A\|=0$, if and only if A is the matrix with all zero component

iii) $\|\alpha A\| = |\alpha| \, \|A\|$

iv) $\|A+B\| \leq \|A\|+\|B\|$

v) $\|A.B\| \leq \|A\|.\|B\|$

If we define a norm on K^n, then the mapping norm of A is given by $\|A\| = \sup_{\|x\|=1} \|Ax\|$

Since all matrix norms are equivalent, we can use the three norms. For all norms of the convergence of the sequence of matrix $(A)_{i,j}$ is equivalent to the component wise convergence.

Usual matrix norms: $\|A\|_1 = \max_{j=1}^{n} \sum_{i=1}^{n} | a_{ij} |$, (column sum norm)

$$\|A\|_2 = (\sum_{j,i}^{n} |a_{j,i}|^2)^{1/2}, \text{ (square sum norm)}$$

$$\|A\|_\infty = \max_{i=1}^{n} \sum_{j=1}^{n} | a_{ij} |, \text{ (row sum norm)}$$

Definition 4: If A is an nxn matrix, the polynomial defined by

$P(\lambda)=\det(A-\lambda I)$ is called the characteristic polynomial of A. Here P is an n^{th}-degree polynomial with real coefficients and consequently, has at most n-distinct zeros, some of which may complex. If λ is a zero of P then, since $\det(A-\lambda I)=0$, implies that the linear system defined by $(A-\lambda I)\, X = 0$ has a solution other than the identically zero solution.

Definition 5: If P is the characteristic polynomial of the matrix A, the zeros of P are called Eigenvalue of A. If λ is an eigenvalue of A and $x \neq 0$ has the property that $(A-\lambda I)\, X = 0$, then X is called the eigenvector of corresponding to the eigenvalue of λ.

Ex: let $A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 2 & 1 \\ -1 & 0 & 0 \end{bmatrix}$

To compute the eigenvalue of A consider $P(\lambda)= \det (A-\lambda I)$ $(2-\lambda)(\lambda^2-\lambda+1)$

The eigenvalue of A are the solution of $P(\lambda)=0, \lambda_1 =2$, $\lambda_2 =1/2 +(\sqrt{3/2}\ )i$,

$\lambda_3=1/2 -(\sqrt{3/2}\ )i$.

The spectra radius σ (A) of a matrix A is defined by σ (A) = max$|\lambda|$,

where λ is an eigenvalue of A. Here the spectra radius σ (A) of the above example is

σ (A) = Max $\{|\lambda_1|, |\lambda_2|, |\lambda_3|\} =2$

Theorem 1.1: If A is an nxn matrix, then σ (A) $\leq ||A||$ for any norm.

Proof: suppose λ is an eigenvalue of A, with eigenvector $x \neq 0$

We have, $Ax = \lambda x$

$$\Rightarrow \|\lambda x\| = \|Ax\|$$

This implies $|\lambda|\|x\| = \|\lambda x\| \leq \|A\|\|x\|$

$$|\lambda| \leq ||A||$$

$$\Rightarrow \sigma (A) = \text{Max } |\lambda| \leq ||A||. \text{ //}$$

Hence in general from the theorem, we have σ (A) $\leq ||A||$, using the convergence theorem a sufficient condition for convergence is that $||A|| \leq 1$. This is because if σ (A) $\leq ||A|| \leq 1$, then σ (A) ≤ 1.

In studying iterative methods of a linear system of equation, it is of particular importance to know when power of matrix become small; that is when all of the entries approach to zero. Matrices of this type are called convergent.

Now we call an (nxn) matrix A is convergent if $\lim_{k \to \infty} (A^k)_{i,j} = 0$

For i=1,2,...n, j=1,2,...n

Ex: let $A=\begin{bmatrix}1/2 & 0\\ 1/4 & 1/2\end{bmatrix}$

Computing power of, we obtain $A^2=\begin{bmatrix}1/4 & 0\\ 1/4 & 1/4\end{bmatrix}$, $A^3=\begin{bmatrix}1/8 & 0\\ 3/16 & 1/8\end{bmatrix}$ and in general

$A^k=\begin{bmatrix}(1/2)^k & 0\\ k/2^{k+1} & 1/2^k\end{bmatrix}$, since ${}_k\underline{\lim}_{\infty}(1/2^k)=0$ and ${}_k\underline{\lim}_{\infty}(k/2^{k+1})=0$

Therefore A is a convergent matrix.

Theorem 1. 2: The following statements are equivalent:

i) A is a convergent matrix

ii) ${}_k\underline{\lim}_{\infty} \| A^k \|=0$, for some norm $\| . \|$

iii) $\sigma(A)\leq 1$.

Proof: i)⇒ii) suppose (i) is true

$\Rightarrow {}_k\underline{\lim}_{\infty} A^k = 0$, by definition

$\Rightarrow \| \ {}_k\underline{\lim}_{\infty} A^k \|=0 \ \Rightarrow {}_k\underline{\lim}_{\infty} \| A^k \|=0$, Since limit of a matrix is continuous

ii⇒iii) by definition $\sigma(A) = {}_k lim_{\infty} \| A^k \|^{1/k}$

$\sigma(A)={}_k lim_{\infty} \| A^k \|^{1/k} \leq 1$. //

Lemma 1: If the spectra radius $\sigma(A)$ satisfies $\sigma(A)<1$,then $(I-A)^{-1}$ exists and $(I-A)^{-1}=I+A+A^2+\ldots$

Proof: For any eigenvalue λ of A, $1-\lambda$ is an eigenvalue of I-A.

Since $|\lambda|\leq\sigma(A)<1$, it follows that no eigenvalue of I-A could be zero and, consequently, I-A is non-singular.

Let $S_m=I+A+A^2+\ldots+A^m$. Then $(I-A)S_m=I-A^{m+1}$ and since A is convergent, by Theorem 2

${}_k\underline{\lim}_{\infty}(I-A)S_m = {}_k\underline{\lim}_{\infty}(I-A^{m+1}) = I$.

Thus, ${}_m\underline{\lim}_{\infty} S_m = (I-A)^{-1}$.

1.3. Principle of successive Iterative methods and Convergence theorem

Let A be a non-singular nxn matrix.

Consider the solution of the matrix equation

$$Ax = b \qquad (1)$$

Suppose we split the matrix A as $A = M - N$, the matrix equation will becomes

$(M - N)X = b$

$$MX = NX + b \qquad (2)$$

The iterative methods associated to this equation in (1) consists the sequence $X^1, X^2, \ldots, X^{k+1}$ of approximation vectors in the following manner:

$$\left.\begin{aligned} x^1 &= M^{-1}Nx^0 + M^{-1}b \\ x^2 &= M^{-1}Nx^1 + M^{-1}b \\ &\cdot \\ &\cdot \\ &\cdot \\ x^{k+1} &= M^{-1}Nx^k + M^{-1}b \end{aligned}\right\} \qquad (3)$$

Now, with an initial guess X^0, the following can represent the above equality iterative relation. $X^{k+1} = TX^k + V$, $k=0,1,2\ldots$ Where $T = M^{-1}N$, $V = M^{-1}$ (4)

Note: The matrix T and the vector V are independent of the index K.Thus if we proceed with an iterative methods of the above from the solution of the matrix (1), then it remains to study the convergence of the method.

The iterative methods considerd in (3) produce from each initial vector X^0 a sequence of vector $\{X^i\}i=0,1,2,\ldots$.We like to know whether X^{k+1} will converges to the exact solution $X = A^{-1}b$. In order to answer this question, we proceed as follows.

Let us define the error vector, $e^k = X^k - X$, associated to the k^{th} iteration.

By definition, X and X^k will satisfies the equation

i.e $X = TX + V$ and $X^k = TX^{k-1} + V$,respectively.

Now, subtracting one equation from the other , we obtain

$\mathbf{e^k = X^k - X = TX^{k-1} - TX = T(X^{k-1} - X) = Te^{k-1}}$, for $k=1,2,\ldots$.

Thus $e^1 = Te^0$, $e^2 = Te^1$, $e^3 = Te^2$. This implies $e^k = Te^{k-1} = T^2e^{k-2} = \ldots = T^ke^0$.

Since the initial approximation X^0 is arbitrary and therefore, for any initial vector e^o, the convergence of the above process is assured if

$\lim_{k\to\infty} e^k = 0$ i.e $\lim_{k\to\infty} T^k e^0 = 0$, for any vector e^0.

This is equivalent to saying $\lim_{k\to\infty} T^k = 0$, where 0 is null matrix. Therefore regarding to the convergence of the iterative method of the form (4), we can state the convergence theorem as follows.

Theorem1.3: (convergence theorem)

For any initial vector $X^0 \in K^n$, the sequence $\{X^k\}_{k=1}^{\infty}$ defined by $X^k = TX^{k-1} + V$, for each $K \geq 1$ and $V \neq 0$ (*)

Converges to the unique solution for $X = TX + V$ if and only if $\sigma(T) < 1$.

Proof: Form equation (*) $X^k = TX^{k-1} + V$

$= T(TX^{k-2} + V) + V$

$= T^2X^{k-2} + (T+I)\,V$

.

.

.

$= T^kX^0 + V(T^{k-1} + T^{k-2} + \ldots + T + I)$

($\Rightarrow$) Assuming $\sigma(T) < 1$, we can use theorem (1.2) and Lemma (1)

$$\lim_{k\to\infty} X^K = \lim_{k\to\infty} T^k X^0 + \lim_{k\to\infty} (\sum_{j=1}^{k-1} T^j)V$$

$= 0.\ X^0 + (I-T)^{-1}\ V = (I-T)^{-1}\ V$

$\lim_{k\to\infty} X^K = (I-T)^{-1}\ V = X$. But $X = TX + V \Rightarrow X = (I-T)^{-1}\ V$.

($\Leftarrow$) To prove the converse, let $\{X^k\}$ converges to X for any vector X^0 from equation (*), it follows that, $X = TX + V$, so for each k

$X - X^k = T\,(X - X^{k-1}) = \ldots = T^k\,(X - X^0)$. Hence for any vector X^0

$\lim_{k\to\infty} T^k (X - X^0) = \lim_{k\to\infty} (X - X^k) = 0$, Since x^0 is arbitrary we obtain.

Therefore $\sigma(T) < 1$. //

Corollary 1: If for any matrix norm $||T|| < 1$, then the sequence $\{X^k\}_{k=0}^{\infty}$ in equation $X^k = TX^{k-1} + V$, converges, for any $X^0 \in K^n$, to a vector X and the following error bonds holds.

$$||X-X^k|| \leq ||T||^k ||X^0-X||$$

and hence, $$||X-X^k|| \leq \frac{||T||^k}{1-||T||} ||X^1 - X||$$

1 .4. Decomposition of the matrix A

Let a non –singular (nxn) matrix A be given, and the system of linear equations Ax = b, with exact solution $X= A^{-1} b$. For iterative methods of the form $X^{k+1} = TX^k + V$, k= 0,1,2, …, it is clear that we should choose a matrix T which confirms the convergence and at the same time TX + V is not difficult to calculate. Therefore we will find certain decomposition of the matrix A of the form M – N in a manner that M can be invertible and satisfy the condition $\sigma(M^{-1}N)<1$ or it is sufficient to verify the condition $||M^{-1}N||<1$

Let define the matrix $D = (d_{ij})$, $L = (l_{ij})$ and $U = (u_{ij})$

Such that $d_{ii} = a_{ii}$ for all i = 1,2 ,… and D is a diagonal matrix

$$l_{ij} = \begin{cases} -a_{ij}, for..i > j \\ 0....., for..i \leq j \end{cases}$$

$$u_{ij} = \begin{cases} -a_{ij},..for..j > i \\ 0.,.....for...j \leq i \end{cases}$$

i.e L is strictly lower triangular matrix and U is strictly upper triangular matrix

From the above definition we can easily verify that A= D-L-U

Now, with different choice of the splitting matrices M and N the following three types of methods are defined.

1) Jacobi-method: For this method, the splitting matrices M and N are defined as

   M = D and N= L+U so that A = M – N = D –L-U

2) Gauss-Seidel method: For this method, the splitting matrices M and N are defined as M = D – L and N = U so that A = M – N = D – L -U

3) Relaxation Method: In this method, the splitting matrices M and N are chosen as

   $M = D/\omega - L$, $N = (1-\omega)/\omega * D + U$

   So that $A = M - N = D/\omega - L - ((1-\omega)/\omega) D + U)$

CHAPTER-TWO

NUMERICAL METHODS

As was mentioned in the introduction, many linear systems are too large to be solved by direct methods based on Gaussian elimination. For these systems, iterative methods are often the only possible method of solution as well as being faster than elimination in many cases. The large area for the application of iterative method is to the linear systems arising in the numerical solution of partial differential equations.

Besides being large, the linear systems to be solved $Ax=b$; usually have several other important properties. They are usually sparse, which means that only small percentages of the coefficients are non-zero. The non-zero coefficients generally have special pattern in the way they occur in A.

2.1. Jacobi –Method

2.1.1. principle of the method

As we have already seen, in this method the matrix A is decomposed in to

$A = M - N = D - (L+U)$

and hence the matrix equation $Ax = b$ will become

$$Dx = (L+U)x + b$$

Now following the iterative procedure explained in section (1.3) the Jacobi method will be written as

$DX^{k+1} = (L+U)X^k + b$, $k = 0,1,2 \ldots$ this implies

$$X^{k+1} = D^{-1}(L+U)X^k + D^{-1}b \qquad (1)$$

The matrix (1) is equivalent to the following system of equations:

$X_1^{k+1} = (b_1 - a_{12}x_2^k - a_{13}x_3^k - \ldots - a_{1n}x_n^k)/a_{11}$

$X_2^{k+1} = (b_2 - a_{11}x_1^k - a_{23}x_3^k - \ldots - a_{2n}x_n^k)/a_{22}$

.

.

.

$Xn^{k+1} = (b_1 - a_{n1}x_1^k - a_{n2}x_2^k - \ldots - a_{1n-1}x_n^k)/a_{nn}$

Here, we assume the diagonal elements are different from zero. But if it has zero and since A is non-singular matrix, then by permuting rows and columns it is possible to get a non-singular matrix. It is suggested that the equations be arranged so that a_{ii} is as large as possible in order to speed the convergence.

Example: Solve the linear system Ax = b given by

$10x_1 - x_2 + 2x_3 = 6$

$-x_1 + 11x_2 - x_3 + 3x_4 = 25$

$2x_1 - x_2 + 10x_3 - x_4 = -11$

$3x_2 - x_3 + 8x_4 = 15$, has solution $x = (1,2,-1,1)^T$

To convert Ax = b to the form x = Tx + V, solve each equation for x_i, for each i = 1,2,3,4 to obtain

$X_1 = (1/10)x_2 - (1/5)x_3 + 3/5$

$X_2 = (1/11)x_1 + (1/11)x_3 - (3/11)x_4 + 25/11$

$X_3 = (-1/5)x_1 + (1/10)x_2 + (1/10)x_4 - 11/10$

$X_4 = (-3/8)x_2 + (1/8)x_3 + 15/8$

Here, we have

$$T = \begin{bmatrix} 0 & \frac{1}{10} & \frac{-1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & \frac{-3}{11} \\ \frac{-1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & \frac{-3}{8} & \frac{1}{8} & 0 \end{bmatrix} \quad \textit{and} \quad V = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ \frac{-11}{10} \\ \frac{15}{8} \end{bmatrix}$$

For an initial approximation let $x^{(0)} = (0,0,0,0)^T$ and generated $x^{(1)}$ by

$X_1^{(1)} = (1/10)x_2^{(0)} - (1/5)x_3^{(0)} + 3/5 = 0.6000$

$X_2^{(1)} = (1/11)x_1^{(0)} + (1/11)x_3^{(0)} + 25/11 = 2.2727$

$X_3^{(1)} = (-1/5)x_1^{(0)} + (1/10)x_2^{(0)} - 11/10 = -1.1000$

$X_4^{(1)} = (-3/8)x_2^{(0)} + (1/8)x_3^{(0)} + 15/8 = 1.8750$

$X^{(1)} = (0.6000, 2.2727, -1.1000, 1.8750)$

Additional iterates, $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^T$, are generated in a similar manner and are presented in the following:

k	2	3	4	5	6	7	8	9	10
$X_1^{(k)}$	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9999	1.0001
$X_2^{(k)}$	1.7159	2.0533	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$X_3^{(k)}$	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$X_4^{(k)}$	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

Table (1)

2.1.2. Condition on the termination of iterations

Suppose we denote the residue vector r as $r^k = b - AX^k$

$r_i^k = (b_i - \sum_{j=1}^{n} a_{ij} x_j^k)$, for i=1,2,…,n . Then the standard criterion for termination of the iteration is $\frac{\| r^K \|}{\| b \|} < \varepsilon$, where ε is arbitrary small. Another standard termination condition used for the relative improvement for x is stated as

$$\frac{\left\| x^k - x^{k+1} \right\|}{\left\| x^{k+1} \right\|} < \varepsilon$$

Practically these two conditions are equivalent.

In conclusion the Jacobin iterative algorithm can be stated as in the following manner, by assigning an arbitrary initial approximation vector $x^{(0)}$

Jacobin iterative Algorithm

To solve Ax=b given an initial approximation x^0

INPUT: The number of equations and unknown n: the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entire b_i, $1 \le i \le n$ of the term b, the entries $x_{oi}; 1 \le i \le n$ of $x^{(0)}$; tolerance ε_1; maximum number of iterations N

OUTPUT: The approximate solution $x_1, x_2, ..., x_n$

Step 1. Set k=1

Step2.While $(k \le n)$ do step 3-6

Step 3: for $i = 1,2,...,n$ set $x_i = \dfrac{-\sum_{\substack{j=1 \\ j \neq 1}} a_{ij} x_{oj} + b_i}{a_{ii}}$

Step 4: If $||x-x_0|| < \varepsilon_1$, then out put $(x_1, x_2, ..., x_n)$ (procedure completed successfully)

STOP

Step 5 : set k=k+1

Setp 6: For i=1,2,…,n, set $x_{oi} = x_i$

Step 7: Out put (Maximum number of iterations exceeded)

(Procedure-completed unsuccessfully)

STOP

Theorem: 2.1. A sufficient condition for the convergence of Jacobi method is that the matrix A of the linear system Ax =b is diagonally dominant.

Proof : we know that the Jacobi iterative method will be written as

$X^{k+1} = D^{-1}(L+U)x^k + D^{-1}b$, k= 0,1,2,…

Let $X^{k+1} = T_j X^k + V$ (1)

Now, to assure the convergence of the iteration method (1) we must have $\sigma(T_j) < 1$. Since the calculation of $\sigma(T_j) < 1$ is often very complicated, one should be satisfied with sufficient conditions like $||T_j|| < 1$.Now if $T_j = (t_{ij})$, $1 \le i, j \le n$.

$\Rightarrow \| T_j \|_\infty \leq 1$ if and only if $\sum_{j=1}^{n} t_{ij} < 1$, for all i=1,2,…,n and since

$T_j = D^{-1}(L+U)$, this implies $\sum_{j=1}^{n} | l_{ij} + u_{ij} | < | d_{ii} |$, for all i

Therefore, by definition of D, L and U the sufficient conditions for convergence of the Jacobi iteration method to solve Ax = b will be as $\sum_{j=1, j\neq i} | a_{ij} | \leq | a_{ii} |$, for all i=1,2,…,n

Hence the theorem.//

2.2. Gauss-seidel method

2.2.1. principle of Gauss-seidel method

Let us assume that the matrix A be decomposed as

$$A = M - N = (D-L) - U$$

Now, in a manner of $X^{(k+1)} = TX^{(k)} + V$, k=1,2, …

The Gauss iterative methods can be defined as

$$X^{(k+1)} = (D-L)^{-1} UX^{(k)} + (D-L)^{-1} b \qquad (1)$$

Since the inverse of (D –L) is complicated to compute, equation (1) will be rewritten in the following manner.

$$(D-L)X^{(k+1)} = UX^{(K)} + b$$

$$\Rightarrow DX^{(K+1)} = LX^{(k+1)} + UX^{(k)} + b$$

$$\Rightarrow X^{(k+1)} = D^{-1} LX^{(k+1)} + D^{-1} UX^{(k)} + D^{-1} b \qquad (2)$$

using this vectorial recurrence relation, we obtain the following formula

$$X_1^{(k+1)} = (b_1 - a_{12} X_2^{(k)} - a_{13} X_3^{(k)} - \ldots - a_{1n} X_n^{(k)})/a_{11}$$

$$X_2^{(k+1)} = (b_2 - a_{21} X_1^{(k+1)} - a_{23} X_3^{(k)} - \ldots - a_{2n} X_n^{(k)})/a_{22} \qquad (3)$$

.

.

.

$$X_n^{(k+1)} = (b_n - a_{n1} X_1^{(k+1)} - a_{n2} X_2^{(k)} - \ldots - a_{n,n-1} X_{n-1}^{(k+1)})/a_{nn}$$

Gauss- Seidel Algorithm

Gauss-Seidel Algorithm for Ax = b

To solve Ax = b given an initial approximation $x^{(0)}$

INPUT: The number of equations and unknown; the entries a_{ij}, $1 \leq .i, j \leq n$ of the matrix A: the entries b_i, $1 \leq .i \leq n$ of the inhomogeneous term b; the entries xo_i, $1 \leq .i \leq n$ of $x^{(0)}$; tolerance ε_1; maximum number of iterations N.

OUTPUT: The approximation solution $x_1, x_2, \ldots, x_n$ or message that the number of iterations was exceeded.

Step 1 Set k = 1

Step2 While (k ≤ N) do Step 3-6

Step 3 For i = 1, 2,...,n

$$X_i = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} xo_j + b_i}{a_{ii}}$$

Setp 4: If $\|x - x_o\| < \varepsilon_1$, then out put $(x_1, x_2, \ldots, x_n)$

(procedure completed successfully)

STOP

Setp5: set k = k+1

Step 6: For i = 1,2,...,n set $xo_i = x_i$

Setp 7: OUTPUT (maximum number of iterations exceeded)

(Procedure completed unsuccessfully)

STOP

2.2.2. Condition on the convergence of Gauss-seidel Method

The iterative formula (1) for the Gauss-seidel method allows one to define the matrix T_{GS} and the vector V_{GS} by

$$T_{GS} = (D - L)^{-1} U \, , \; V_{GS} = (D - L)^{-1} b$$

As we have explained for the Jacobi Method, the Gauss-Seidel Method will converge if $\| T_{GS} \| < 1$; which will reduces to the condition

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| < |a_{ii}| \text{ for } i=1,2, \ldots ,n$$

That is the above condition will ensure the convergence of the Gauss-Seidel Method, more precisely the method will converge if the matrix A is diagonally dominant.

Remark: 1. A single permutation of the row can be transforming the Convergence in to divergence and vice-versa.

2. In the Gauss-Seidel Method, once $X_i^{(k+1)}$ is computed, the value of $X_i^{(k)}$ is no more necessary for latter calculations and, therefore, for successive iteration it is sufficient to have a single one-dimensional array to store the values of the approximate vector.

Each new component $X_i^{(k+1)}$ is immediately used in the computation of the next component. This is convenient for computer calculations, since the new value can be immediately stored in the calculation that held the old value; this minimize the number of necessary storage location.

Definition: Let A be Hermitian matrix of order n. It is called positive definite if and only if $(Ax, x) > 0$ for all $x \neq 0$ in C^n.

$$\text{where } (Ax, x) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \; ;$$

Theorem 1: If the matrix A is positive definite, the Gauss-Seidel

Method

$X_i^{(k+1)} = (D - L)^{-1} U X_i^{(k)} + (D - L)^{-1} b$ converges

independently of the initial vector.

Proof: we write $A = -L + D - L^T$, since A is symmetric. The T_{GS} is then

$$\mathbf{T_{GS}} = (D - L)^{-1} L^T$$

Let $-\lambda$ and x be respectively, an eigenvalue and eigenvector of T_{GS}. Then

$(D - L)^{-1} L^T x = -\lambda x$

This implies $L^T x = -\lambda (D - L)x$

Even though A is postive definte, the eigenvalue of T_{GS} may still be complex. We have

$$X^* L^T X = -X^* \lambda (D - L) X \qquad (1)$$

where X^* denotes the conjugate transpose of X.

Adding $X^* (D - L) X$ to both side

We get $X^* AX = (1+\lambda) X^* (D - L)X$, (2)

Since A is real and symmetric, the conjugate transpose of the left hand side of (1) leaves this quantity unchanged. Therefore,

$$(1+\bar{\lambda}) X^* (D - L)^T X = (1+\lambda) X^* (D - L)X$$

$$= (1+\lambda)(X^* DX - X^* LX)$$

$$= (1+\lambda)[X^* DX + \bar{\lambda} X^* (D + L)^T X]$$

The last line following from use of conjugate transpose of (1). Rearranging the terms, we have

$$(1-|\lambda|^2) X^* (D - L)^T X = (1+\lambda) X^* DX \qquad (3)$$

Multiplying both sides Eq.(3) by $(1+\bar{\lambda})$ and then using the conjugate transpose of (2), we get

$$(1-|\lambda|^2)X^*AX = |1+\lambda|^2X^*DX$$

since A is positive definite, so is D; More over, no eigenvalue of $-\lambda$ of T_{GS} can equal to one. Therefore, we must have $1-|\lambda|^2 > 0$,which means the eigenvalue of T_{GS} lie with in the unit circle .

This implies

$$\delta(T_{GS}) < 1. \; //$$

Note: The difference between the Jacobi and Gauss-Seidel Method is that in the later as each component of $X_i^{(k+1)}$ is computed, we use immediately in the iteration. For this reason the Gauss-Seidel Method is sometimes called the method of successive displacement.

The Jacobi-Method as we have presented it here is seldom used. This is largely because the Gauss-Seidel Method almost always converges when the Jacobi Method does, may converge when the Jacobi Method does not, and generally converges faster than Jacobi Method. Further more, the implementation of the Gauss-Seidel Method on computer is more efficient than that of the Jacobi Method.

Obviously, solving problem with minimum memory storage is very important. It is often unnecessary to store A explicitly, so we note that in the case of Jacobi-Method it requires 3n memory places where as the Gauss-Seidel Method requires only 2n memory places.

Finally, note that the Gauss-Seidel differs from the Jacobi Method in substituting the newly computed values of $X_i^{(k+1)}$ in place of $X_i^{(k)}$ at the $(k+1)^{th}$ iteration, and this is possible because in the expression of $X_i^{(k+1)}$ all the $X_j^{(k+1)}$ terms with $j < i$ have been already evaluated. And also, as in the case of the Jacobi Method, we assume that the pivot a_{ii} are non-zero.

Definition: If $x^* \in R^n$ is an approximation to the solution of linear system defined by $Ax = b$, the residual vector for x^* with respect to this system is defined by $r = b - Ax^*$.

In procedures such as the Jacobi or the Gauss-Seidel Methods, a residual vector is associated with each calculation of an approximate component to the solution vector. If we let

$$r_i^{(k+1)} = (r_{1i}^{(k+1)}, r_{2i}^{(k+1)}, \ldots, r_{ni}^{(k+1)})^T$$

denote the residual vector for the Gauss-Seidel method corresponding to the approximate solution vector $(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k)}, x_i^{(k)}, \ldots, x_n^{(k)})^T$, the m^{th} component of $r_i^{(k)}$ is

$$r_{mi}^{(k+1)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k+1)} - \sum_{j=i}^{n} a_{mj} x_j^{(k)}$$

$$= b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k)} - a_{mi} x_i^{(k)}, \quad \text{for each } m = 1,2,\ldots,n$$

In particular, the i^{th} component of $r_i^{(k)}$ is

$$r_{ii}^{(k+1)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} - a_{ii} x_i^{(k)}$$

So

$$a_{ii} x_i^{(k)} + r_{ii}^{(k+1)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \qquad (1)$$

Recall, however, that in the Gauss-Seidel method

$$x_i^{(k)} - \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right]$$

So, Eq.(1), can be written as

$$a_{ii} x_i^{(k)} + r_{ii}^{(k+1)} = a_{ii} x_i^{(k+1)}$$

$$or \quad x_i^{(k+1)} = x_i^{(k)} + \frac{r_{ii}^{(k+1)}}{a_{ii}}$$

2.3. Relaxation Method

Since the rate of convergence of stationary iterative process depends on the spectra radius of T, any modification of the matrix T that will reduce the spectra radius will increase the rate of convergence. Now, we consider a method of accelerating the convergence of iterative process.

The result of the previous section suggest looking for simple matrices T for which, the iterative method

$$X^{(k+1)} = TX^{(k)} + V \text{ , } K= 0,1,2,\ldots$$

Converges perhaps still faster than the Gauss-Seidel. More generally, one can consider classes of suitable matrices $T(\omega)$ depending on parameter ω and try to choose the parameter ω in an "optimal" way.

So that $\sigma(T_\omega)$ as a function of ω becomes as small as possible.

$$T(\omega) = \frac{1}{\omega} D(I - \omega L) \qquad (1)$$

2.3.1. Principle of the Method

In this section, we present an iterative method, which has the same advantageous as those of the Gauss-Seidel, method, but it converges more rapidly. For this we introduce a parameter $\omega \neq 0$, and suppose for (k+1) th approximation $X^{(k+1)}$, we already know the component $X_i^{(k+1)}$, k= 1, 2,…, n.

As in the Gauss-Seidel Method, we then define an auxiliary quantity $\bar{X}_i^{(k+1)}$ by

$$\bar{X}_i^{(k+1)} = [b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k] \Big/ a_{ii} \; i = 1,2,\ldots,n \qquad (2)$$

$X_i^{(k+1)}$ is obtained by certain averaging of $\bar{X}_i^{(k+1)}$ and $X_i^{(k)}$

$$X_i^{(k+1)} = (1-\omega) X_i^{(k)} + \omega \bar{X}_i^{(k+1)} \qquad (3)$$

If ω =1, then it is clear that the approximation scheme will reduce the Gauss-Seidel. For ω > 1 it is called an over-relaxation Method or successive over-relaxation (SOR) Method and ω <1, it is known as under relaxation Method.

Now by substituting (2) in (3), we obtain:

$$X_i^{(k+1)} = X_i^k + \omega\{[(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k) \Big/ a_{ii}] - x_i^k\} \quad (4)$$

for i = 1,2,...n

This implies that

$$X_i^{(k+1)} = X_i^k + \omega\{[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k) - a_{ii}x_i^k] \Big/ a_{ii}\} \text{ , for i=1,2,...,n}$$

Finally, adding the last two terms in the above equation we obatin

$$X_i^{(k+1)} = X_i^k + \frac{\omega}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i}^{n} a_{ij}x_j^k) \text{for i=1,2,...,n} \quad (5)$$

Relaxation Algorithm

To solve Ax = b given the parameter ω and an initial approximation $x^{(0)}$:

INPUT: The number of equations and unknowns; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of b; the entries x_{oi}, $1 \le i, j \le n$ of $x^{(0)}$; the parameter ω; tolerance ε_1; maximum number of iterations N.

OUTPUT: The approximate solution $x_1, x_2, \ldots, x_n$ or message that the number of iterations was exceeded

Step 1: Set k = 1

Step 2: While (k ≤ N) do steps 3 – 6

Setp 3: For i = 1,2,…,n

$$\text{Set } x_i = (1 - \omega)x_{0i} + \frac{\omega(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum a_{ij}xo_j + b_i)}{a_{ij}} \qquad (6)$$

Setp 4: If $\|x - x_o\| < \varepsilon_1$, then out put $(x_1, x_2, \ldots, x_n)$

(Procedure completed successfully)

STOP

Setp5: set k = k+1

Step 6: For i = 1,2,…,n set $xo_i = x_i$

Setp 7: OUTPUT (maximum number of iterations exceeded)

(Procedure completed unsuccessfully)

STOP

The number of memory position required to store the matrix A and the vector $X_i^{(k)}$ is identical to that of Gauss-Seidel Method namely, 2n memory position without considering A.

We represent relaxation method in matrix form as follows

Writing Eq (5) in the form

$$a_{ii}X_i^{(k+1)}+\omega\sum_{j=1}^{i=1}a_{ij}x_j^{(k+1)}$$

$$=a_{ii}x_i^{(k)}-a_{ii}\omega x_i^{k}-\omega\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}+\omega b_i \qquad (7)$$

will lead to the matrix equation

$$(D-\omega L)X^{(k+1)} = \{(1-\omega)D+\omega U\}X^{(k)}+\omega b \qquad (8)$$

Since $(D-\omega L)$ is non-singular for all choices of ω and if we define

$E= D^{-1}L$ and $F =D^{-1}U$, then equation (8) becomes

$$(I-\omega E)X^{(k+1)} = \{(1-\omega)I+\omega F\}X^{(k)}+\omega D^{-1}b \qquad (9)$$

and the relation iterative method is rewritten as

$$X^{(k+1)} = (I-\omega E)^{-1}\{(1-\omega)I+\omega F\}X^{(k)}+\omega(I-\omega E)^{-1}D^{-1}b \qquad (9)$$

Suppose we write Eq(10) in the following iterative form

$$X^{(k+1)} = T_\omega X^{(k)}+V_\omega \qquad (11)$$

where $T_\omega = (I-\omega E)^{-1}\{(1-\omega)I+\omega F\}$

$$V_\omega = \omega(I-\omega E)^{-1}D^{-1}b$$

Now, we can see that Eq (10) is a stationary iterative method. In fact, a relaxation method is a particular case of large set of acceleration method of iterative method.

2.3.2. Termination Criteria for Gauss-Seidel and Relaxation Method

We know that the Gauss-Seidel Method is a particular case of relaxation method with ω =1. Also, note that the relaxation Algorithm does not calculate explicitly the residual vector $r^k = b-AX^k$

Thus, we cannot construct a termination criterion depending on this vector r. However, it is linked to the vector r^k whose i^{th} component is defined by

$$\bar{r}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)} \tag{12}$$

Now putting this Eq (12) in algorithm (6), we get

$$x_i^{(k+1)} = x_i^{(k)} - (\frac{\omega}{a_{ii}}) \bar{r}_i^{(k)} \tag{13}$$

Writing Eq (12), in vector form

$$\bar{r}^{(k)} = b - Lx^{(k+1)} - Ux^{(k)} - Dx^{(k)} \tag{14}$$

But the residual vector r is defined by

$$r^{(k+1)} = b - Lx^{(k+1)} - Ux^{(k+1)} - Dx^{(k+1)} \tag{15}$$

therefore, using Eqs (14) and (15), the vector $r^{(k+1)}$ can be written as

$$r^{(k+1)} = \bar{r}^{(k)} + Ux^{(k)} - Ux^{(k+1)} + Dx^{(k)} - Dx^{(k+1)}$$

$$r^{(k+1)} = \bar{r}^{(k)} - (D + L)(x^{(k+1)} - x^{(k)}) \tag{16}$$

Substituting Eq (13) in to Eq (16), we get

$$\begin{aligned} r^{(k+1)} &= \bar{r}^{(k)} - (U + D)\omega D^{-1} \bar{r}^{(k)} \\ r^{(k+1)} &= [(1-\omega)I - \omega U D^{-1}] \bar{r}^{(k)} \end{aligned} \tag{17}$$

From relations (13) and (17), we see that if ω is not very close to zero then a test type

$$\frac{\left\| x^{(k+1)} - x^{(k)} \right\|}{\left\| x^{(k)} \right\|} < \varepsilon_1$$

Or equivalent to the type of

$$\frac{\|r\|}{\|b\|} < \varepsilon_2 .$$

2.3.3. Condition on the convergence of Relaxation Method

Now, consider Eq (10)

$$x^{(k+1)} = (I - \omega E)^{-1} \{(1-\omega)I + \omega F\} x^{(k)} + \omega (I - \omega E)^{-1} D^{-1} b$$

$$X^{(k+1)} = T_\omega X^{(k)} + V_\omega$$

From this method, the rate of convergence, therefore is determined by the spectra radius of the matrix T_ω

We begin by listing, a few qualitative results about $\delta(T_w)$ in order to discuss the convergence of the relaxation methods.

The following theorem shows that in relaxation methods only parameters ω with $0 < \omega < 2$ at best, leads to convergent method. Now, before going to see the Theorems, let us consider a simple example.

Example: The linear system Ax = b given by

$$4x_1 + 3x_2 = 24$$
$$3x_1 + 4x_2 - x_3 = 30$$
$$-x_2 + 4x_3 = -24$$

has the solution $(3,4,-5)^T$. Gauss-Seidel and SOR method with $\omega = 1.25$ will be used to solve this system, using $x^{(0)} = (1, 1, 1)^T$ for both methods. The equations for the gauss-seidel method are

$$x_1^{(k+1)} = -0.75x_2^{(k)} + 6,$$
$$x_2^{(k+1)} = -0.75x_1^{(k+1)} + 0.25x_3^{(k)} + 7.5$$
$$x_3^{(k+1)} = 0.25x_2^{(k+1)} - 6, \text{ for each } k = 1, 2, \ldots$$

and the equations for the relaxation method with $\omega = 1.25$ are

$$x_1^{(k+1} = -0.25x_1^{(k)} - 0.9375x_2^{(k)} + 7.5$$
$$x_2^{(k+1)} = -0.9375x_1^{(k+1)} - 0.25x_2^{(k)} + 0.3125x_2^{(k)} + 9.375$$
$$x_3^{(k+1)} = 0.3125x_2^{(k+1)} - 0.25x_3^{(k)} - 7.5$$

The first seven iterates for each method are listed in the following Tables.

k	0	1	2	3	4	5	6	7
$x_1^{(k+1)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_2^{(k+1)}$	1	-3.81250	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_3^{(k+1)}$	1	-5.04687	-5.029297	-5.018312	-5.011444	-5.007153	-5.004470	-5.002794

Table (2): Gauss-Seidel method

k	0	1	2	3	4	5	6	7
$x_1^{(k+1)}$	1	6.312500	2.6223140	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k+1)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k+1)}$	1	-6.650146	-4.600238	-5.096686	-4.973489	-5.005713	-4.998282	-5.0003486

Table (3): Relaxation method

In order for the iterates to be accurate to seven decimal places the Gauss-Seidel method 34 iterations, as opposed to 14 iterations for the relaxation method with ω =1.25.

The obvious question to ask is how the appropriate value of ω is chosen. Although no complete answer to this question for nxn linear system, the following results can be used in certain situations.

Theorem 3.1: For arbitrary matrices A one has $\delta(T_\omega) \geq |\omega - 1|$

Proof: I- ω L is a lower triangular matrix with 1 as diagonal elements, so that

$\det(I - \omega L) = 1$, for all ω

For the characteristic polynomial $P(\lambda)$ of T_ω it follows that

$$P(\lambda) = \det(\lambda I - T_\omega) = \det((I - \omega L)(\lambda I - T_\omega))$$

$$= \det((\lambda + \omega - 1)I - \omega\lambda L - \omega U)$$

The constant term P(0) of $P(\lambda)$ is equal to the product of the eigenvalues $\lambda_i(T_\omega)$:

$$\prod_{i=1}^{n} \lambda_i(T_\omega) = P(0) = \det((\omega-1)I - \omega U) = (\omega-1)^n$$

Thus $P(T_\omega) = \max_i |\lambda_i.(T_\omega)| \geq (\omega-1)$

Note: For matrices A with $L \geq 0, U \geq 0$, only relaxation can give faster than Gauss-Seidel Method.

Theorem3.2: For a positive definite matrices A one has $\delta(T_\omega) < 1$, *for* $0 < \omega < 2$.

In particular, the Gauss-Seidel method ($\omega = 1$) converges for definite matrices.

Proof: Let $0 < \omega < 2$, and A is positive definite. Then $U = L^H$, in the composition of $A = D - L - U$ of A.

For matrix $T_\omega = T$ in (1)

One $T = \frac{1}{\omega} D - L$, and the matrix

$$T + T^H - A = \frac{1}{\omega} D - L + \frac{1}{\omega} D - U - (D - L - U)$$

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

$$= (\frac{2}{\omega} - 1) D$$

is positive definite, since the diagonal elements of positive definite matrix A are positive and $(\frac{2}{\omega} - 1) > 0$.

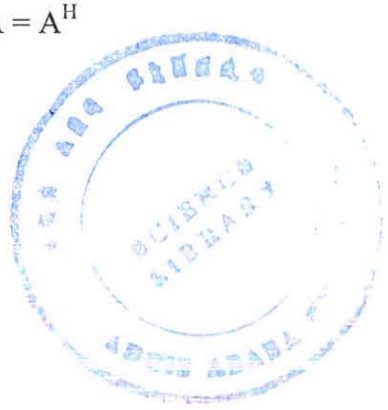
We first show that the eigenvalues λ of $A^{-1}(2T - A)$ all lie in the interior of the right half plane, $\mathrm{Re}\,\lambda > 0$.

Indeed, if x is an eigenvector for λ, then $A^{-1}(2T - A)x = \lambda x$

$$X^H(2T - A)x = \lambda x^H A x$$

Taking the conjugate complex of the last relation, gives because $A = A^H$

$$X^H(2T^H - A)x = \bar{\lambda} x^H A x$$



By definition, it follows that

$$X^H(T+T^H-A)x = \mathrm{Re}\,\lambda\, x^H Ax$$

But now, A and $T + T^H - A$ are postive definite and thus $\mathrm{Re}\,\lambda > 0$. For the matrix

$$Q := A^{-1}(2T-A) = 2A^{-1}T - I \text{ one has } (Q-I)(Q+I)^{-1} = T_\omega$$

Here, we observe that T is non-singular matrix and therefore T^{-1} and $(Q+I)^{-1}$ exists.

Now, if μ is an eigenvalue of T_ω and x is the corresponding eigenvector, then from

$(Q-I)(Q+I)^{-1}x = T_\omega x = \mu x$

It follows, for the vector $y := (Q+I)^{-1}x \neq 0$, that $(Q-I)y = \mu(Q+I)y$,

$(1-\mu)Qy = (1+\mu)y$, since $y \neq 0$, we must have $\mu \neq 0$

$Qy = \dfrac{1+\mu}{1-\mu}y$ i.e $\lambda = \dfrac{1+\mu}{1-\mu}$ is an eigenvalue of $Q = A^{-1}(2T-A)$

Hence $\mu = \dfrac{\lambda-1}{\lambda+1}$ .For $|\mu|^2 = \mu\bar{\mu}$, one obtains

$$|\mu|^2 = \frac{1-2\mathrm{Re}\,\lambda+|\lambda|^2}{1+\mathrm{Re}\,\lambda+|\lambda|^2} \text{ and since } \mathrm{Re}\,\lambda > 0, \text{ for } 0<\omega<2$$

This implies $|\mu|<1$.

Therefore, $\delta(T_\omega) < 1$.

Definition: A matrix A which, relative to the decomposition $(A = D - L - U)$

Where $E = D^{-1}L$, $F = D^{-1}U$, $J = E + F$ and $T := (I-E)^{-1}F$, assuming $a_{ii} \neq 0$ for $i = 1,2,\ldots,n$.

$A = D(I - E - F)$, has the property that the eigenvalues of the matrices

$$J(\alpha) = \alpha E + \alpha^{-1}F$$

For $\alpha \neq 0$ are independent of α, is called consistently ordered.

Theorem 3.3: Let A be a consistently ordered and $\omega \neq o$. Then:

a) With μ, also $-\mu$ is an eigenvalue of $J = E + F$

b) If μ is an eigenvalue of J and

$$(\lambda+\omega-1)^2 = \lambda\omega^2\mu^2 \qquad (1)$$

then λ is an eigenvalue of T_ω

Proof:

a) Since A is consistently ordered, the matrix J(-1) = -E –F = -J has the same eigenvalue as J(1) = E + F

b) Because det(I - ω E) = 1, for all ω, one has

$$\det(\lambda I - T_\omega) = \det[(I - \omega E)(\lambda I - T_\omega)]$$

$$= \det[\lambda I - \lambda \omega E - (1 - \omega)I - \omega F]$$

$$= \det((\lambda + \omega - 1)I - \lambda \omega E - \omega F) \qquad (2)$$

Now, let μ be an eigenvalue of J = E + F and λ be the solution of (1). Then $\lambda + \omega - 1 = \sqrt{\lambda}\omega\mu$ or $\lambda + \omega - 1 = -\sqrt{\lambda}\omega\mu$. Because of (a), we can assume with out loss of generality that

$$\lambda + \omega - 1 = \sqrt{\lambda}\omega\mu$$

If $\lambda = 0$, then $\omega = 1$, so that by (2), det (0.I – T_1) = det (- F) = 0.

i.e, λ is an eigenvalue of T_ω. If $\lambda \neq 0$, it follows from (2) that

$$\det(\lambda I - T_\omega) = \det[(\lambda + \omega - 1)I - \sqrt{\lambda}\omega(\sqrt{\lambda}E + \frac{1}{\sqrt{\lambda}}F)]$$

$$= (\sqrt{\lambda}\omega)^n \det[\mu I - (\sqrt{\lambda}E + \frac{1}{\sqrt{\lambda}}F)]$$

$$= (\sqrt{\lambda}\omega)^n \det[\mu I - (E + F)] = 0$$

Since the matrix $J(\sqrt{\lambda}) = (\sqrt{\lambda}E + \frac{1}{\sqrt{\lambda}}F)$ has the same eigenvalue as J = E + F and μ is an eigenvalue of J. Therefore, det (λ I - T_ω) = 0, and λ is an eigenvalue of T_ω. //

To obtain the Optimal ω Value for a relaxation method, we proceed as follows

Suppose A is a block tri-diagonal matrix of order n. Assume that A is written as

$$A = \begin{bmatrix} D_1 & U_1 & & & & & \\ L_2 & D_2 & U_2 & & & & \\ & L_3 & D_3 & . & & & \\ & & . & . & . & & \\ & & & . & . & . & \\ & & & & .. & . & U_{p-1} \\ & & & & & L_p & D_p \end{bmatrix}$$

Where each D_i is a diagonal sub-matrix of order n_i such that

$\sum_{i=1}^{p} n_i = n$. Then $\delta(T_{Gs}) = \delta^2(T_J)$, where T_{Gs} and T_J are the block matrices obtained by decomposing the matrix A for Gauss-Seidel and Jacobi method respectively.

Theorem 3.4: If A is a block, tri-diagonal and if all the eigenvalues of the matrix corresponding to Jacobi method are real then Jacobi and relaxation method ($0<\omega<2$) for block matrices converge if the value of ω that minimizes $\delta(T_\omega)$ is

$$\omega = \frac{2}{(1+\sqrt{1-\delta(T_{Gs})})} = \frac{2}{(1+\sqrt{1-\delta^2(T_J)})} \quad \text{and} \quad \delta(T_\omega) = \omega - 1$$

it can be verified that the variation of the spectra radius of the matrix T as the function of ω will be look like the following.

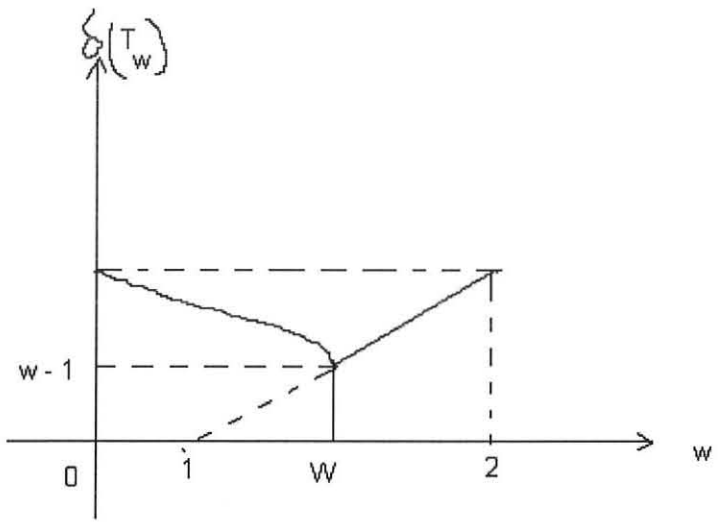


Fig (1) spectra radius of T_W

Proof: The eigenvalue μ_i of the matrix T_J, by assumption are real, and

$$-\delta(T_J) \le \mu_i \le \delta(T_J) < 1.$$

For fixed $\omega \in (0,2)$ [by Theorem (3.1) it suffices to consider this domain] to each μ_i there belong two eigenvalues $\lambda_i^1(\omega,\mu_i), \lambda_i^2(\omega,\mu_i)$ *of* T_ω, which are obtained by solving

the quadratic equation (1) in the Theorem (3.3) in λ. Geometrically, $\lambda_i^1(\omega)$, $\lambda_i^2(\omega)$ are obtained as abscissae of the points of intersection of the straight line

$$g_\omega(\lambda) = \frac{\lambda + \omega - 1}{\omega}$$

with the parabola $m_i(\lambda) = \pm\sqrt{\lambda}\mu_i$. The line $g_\omega(\lambda)$ has the slop $\frac{1}{\omega}$ and passes through the point (1,1).

If it does not intersect the parabola $m_i(\lambda), then\ \lambda_i^1, \lambda_i^2$ are conjugate complex number with modulus $|\omega - 1|$, as obtained immediately from (1) of Theorem (3.4). Thus,

$$\delta(T_\omega) = \max_i(|\lambda_i^1(\omega)|, |\lambda_i^2(\omega)|) = \max(|\lambda^1(\omega)|, |\lambda^2(\omega)|)$$

the $\lambda^1(\omega), \lambda^2(\omega)$ being obtained by intersection of $g_\omega(\lambda)$ with $m(\lambda) = \pm\sqrt{\lambda}\mu$, where $\mu = \delta(J) = \max_i|\mu_i|$. By solving the quadratic equation (1) of Theorem (3.3), with $\mu = \delta(J)$. Then the optimal value of ω is become

$$\omega = \frac{2}{(1 + \sqrt{1 - \delta^2(T_J)})}$$

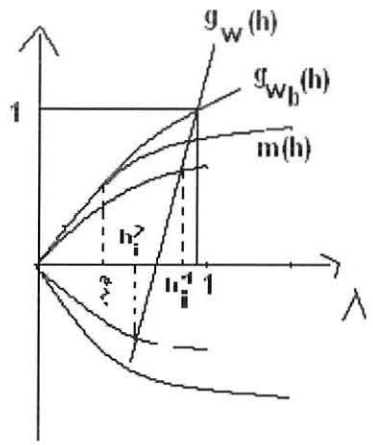


Figure (2): Determination of ω

Example: The linear system Ax = b given by

$$4x_1 + 3x_2 = 24$$
$$3x_1 + 4x_2 - x_3 = 30$$
$$-x_2 + 4x_3 = -24$$

Now the matrix A is given by

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

This matrix is positive definite and tri-diagonal, so we apply the above theorem

Since $T_J = D^{-1}(L + U)$

$$T_J = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}$$

We have

$$T_J - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix}$$

So $\det(T_J - \lambda I) = -\lambda(\lambda^2 - 0.625)$

Thus,

$$\delta(T_J) = \sqrt{0.625}$$

$$\text{and} \quad \omega = \frac{2}{1+\sqrt{1-\delta^2(T_J)}} = \frac{2}{1+\sqrt{1-0.625}} = 1.24$$

This explains the comparatively rapid convergence by using $\omega = 1.25$

2.3.4. Number of Iterations Necessary in Reducing the Error by a Factor of ε

Here it is necessary to see the speed of convergence,

Let x be the exact solution of the matrix equation Ax = b. We know then that the error at the k^{th} iteration is given by

$$e^k = x^k - x$$

Now let define the quantity α such that $\alpha^k = \frac{\|e^k\|}{\|e^0\|}$ (1)

Since we know that $e^k = T^k e^0$, and therefore, taking the norm on both sides we get

$$\|e^k\| \leq \|T^k\| \|e^0\| \quad (2)$$

From equations (1) and (2), it follows immediately that $\|T^k\|$ is an upper bounds of α^k.

i.e $\alpha^k \leq \|T^k\|$ (3)

And taking the k^{th} root on both sides we obtain the inequality

$$\alpha \leq (\|T^k\|^{\frac{1}{k}})$$

let us define the number $R(k,T) = - - \ln(\|T^k\|^{\frac{1}{k}})$

as the average rate of convergence in k iterations

In order to reduce the error by ε , we find the number k of iterations such that $\alpha^k < \varepsilon$. For this eq (3), it is sufficient to choose k satisfying the inequality $\|T^k\| < \varepsilon$

This is provided that $k \geq \frac{-\ln(\varepsilon)}{-\ln(\|T^k\|^{\frac{1}{k}})} = \frac{-\ln(\varepsilon)}{R(k,T)}$

Note that for any arbitrary matrix A the calculation of the denominator will be very clumsy. Finally, we know that for Hermitian matrix $\|T\|_2 = \delta(T)$. using this result we can easily verify that $R(k,T) = -\ln \delta(T)$

And we call this the asymptotic rate of convergence.

2. 4. Conjugate Gradient Method

Let us first see, how to find solution by optimization method.

Consider the linear system of equations Ax = b, where A is symmetric and positive definite matrix and x^* is the exact solution vector. Note that if A is not positive definite then we can make it to be a symmetric and positive definite matrix by pre-multiplying by A^T and solve the new matrix equation

$$A^TAx = A^Tb$$

Define the quadratic form E(.) by

$$E(x) = \tfrac{1}{2}(x-x^*)^TA(x-x^*) = (1/2)x^TAx - x^Tb + (1/2)(x^*)^TAx^* \quad (1)$$

Since the residue vector r is given by $r = b - Ax$ and x^* is the solution of the matrix equation $Ax = b$, we get $x = A^{-1}(b-r)$ and $x^* = A^{-1}b$. This implies $E(x) = (1/2)r^TA^{-1}r$.

The quadratic functional E(x) associated to a symmetric and positive definite matrix A has a unique minimum value and it is obtained by making the gradient of E(x) to zero. But from Eq.(1), we obtain the gradient to be

$$\nabla E(x) = \left(\frac{\partial E(x)}{\partial x_1}, \frac{\partial E(x)}{\partial x_2}, \ldots, \frac{\partial E(x)}{\partial x_n}\right)^T = -(b - Ax) = -r \quad (2)$$

Thus, the minimum of the quadratic functional E(x) corresponds to the point x such that $Ax - b = 0$. In other words, the exact solution x^* of the system $Ax = b$ corresponds to the vector minimizing the quadratic functional E(x). That is the solution of the matrix equation $Ax = b$ is equivalent to the minimization of the quadratic functional

$E(x) = (1/2)r^TA^{-1}r$.

Suppose E(x) is continuous and continuously differentiable function in a neighborhood of $x^{(k)} \varepsilon R^n$. Consider the Taylor series expansion of E(x) at $x^{(k)}$, namely:

$$E(x^{(k)} + \Delta x) = E(x^{(k)}) + \Delta x.\nabla E(x^{(k)}) + O(\|\Delta x\|) \quad (3)$$

If $\|\Delta x\|$ is very small, then we can neglect the higher order terms.

Suppose the criterion

$$\Delta x.\nabla E(x^{(k)}) < 0 \quad (4)$$

is satisfied. Then from Eq. (3), we obtain $E(x^{(k)} + \Delta x) < E(x^{(k)})$.

Definition: A direction Δx satisfying the condition (4) is called a descent direction for the function E(x) at the point $x^{(k)}$.

The general form of a descent methods defined by

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)} \tag{5}$$

where $p^{(k)}$ is a search direction at the point $x^{(k)}$ for E(x) and $\alpha^{(k)}$ is a scalar whose value remains to be defined.

In fact, the value of the step $\alpha^{(k)}$ is chosen such that the minimum of $E(x^{(k+1)})$ is attained at $\alpha^{(k)}$ in the descent direction $p^{(k)}$. Now from Eq.(1), we obtain

$$E(x^{(k+1)}) = (1/2)(x^{(k)} + \alpha^{(k)}p^{(k)} - x^*)^T A(x^{(k)} + \alpha^{(k)}p^{(k)} - x^*) \tag{6}$$

As a function of $\alpha^{(k)}$, $E(x^{(k+1)})$ will attain a minimum if

$$\frac{\partial E}{\partial \alpha^{(k)}}(x^{(k+1)}) = (p^{(k)})^T A(x^{(k)} + \alpha^{(k)} p^{(k)} - x^*) = 0$$

i.e $(p^{(k)})^T (Ax^{(k)} - b + A\alpha^{(k)} P^{(k)}) = 0$

this implies $(p^{(k)})^T(-r^{(k)} + A\alpha^{(k)} p^{(k)}) = 0$

The value of $\alpha^{(k)}$ is chosen such that it satisfies the condition $(\frac{\partial E}{\partial \alpha^{(k)}}) = 0$, and hence gives the minimum value in the direction of $p^{(k)}$. Thus, we obtain

$$\alpha^{(k)} = \frac{(p^{(k)})^T r^{(k)}}{(p^{(k)})^T A p^{(k)}} \tag{7}$$

The general formula of the descent method for the solution of the matrix equation Ax = b will be written as

$$x^{(k+1)} = x^{(k)} + \frac{(p^{(k)})^T r^{(k)}}{(p^{(k)})^T A p^{(k)}} \cdot p^{(k)} \tag{8}$$

where $r^{(k)} = b^{(k)} - Ax^{(k)}$

2.4.1. Gradient Method

Suppose we choose the search direction $p^{(k)}$ as, $p^{(k)} = -\nabla E(x^{(k)})$, i.e we choose $p^{(k)}$ the most rapidely descending direction in a neighbourhood of $x^{(k)}$. Then from Eq. (2), we see that $p^{(k)}$ is the same as the vector $r^{(k)}$, and the Eq.(8) can be written in the following form

Gradient algorithm (Method of steepest descent) for Ax = b

$$\left.\begin{aligned} x^{(k+1)} &= x^{(k)} + \frac{(r^{(k)})^T . r^{(k)}}{(r^{(k)})^{(k)} . Ar^{(k)}} . r^{(k)} \\ &\text{where } r^{(k)} = b - Ax^{(k)} \\ &\text{Ter}\min\text{ate the direction if} \\ &\frac{\left\| r^{(k+1)} - r^{(k)} \right\|}{\left\| r^{(k)} \right\|} < \varepsilon \\ k &= 0,1,2,\ldots,k\max \end{aligned}\right\} \quad (9)$$

The algorithm (9) is an iterative method leading to the solution of the linear system Ax = b. The gradient will converge but its convergence is slow. The optimal local strategy of finding the steepest direrction is not a good one to search the global minimum.

In the conjugate gradient method, we take the orthogonal directions in the sense of the matrix. i.e we choose the search directions as A-conjugate directions.

2.4.2. Choice of conjugate direction

Definition: The direction $p^{(1)}, p^{(2)}, \ldots$ are called A- conjugate if they satisfy the condition

$$(p^{(k)})^T A p^{(k-1)} = 0, \text{ for all } k \quad (10)$$

we search the vector $p^{(k)}$ in the plane formed by the direction vectors $p^{(k-1)}$ and $r^{(k)}$ which are orthogonal. Then by definition, the search direction $p^{(k)}$ is taken as

$$p^{(k)} = r^{(k)} + \beta^{(k)} p^{(k-1)} \quad (11)$$

and the scalar $\beta^{(k)}$ is chosen, in which it minimizes $E(x^{(k+1)})$. Substituting Eq.(11) in to Eq.(6), and differentiating with respect to $\beta^{(k)}$ we obtain

$$\begin{aligned} \frac{\partial E}{\partial \beta^{(k)}}(x^{(k+1)}) &= \alpha^{(k)} (p^{(k-1)})^T A[x^{(k)} - x + \alpha^{(k)}(r^{(k)} + \beta^{(k)} p^{(k-1)})] \\ &= -\alpha^{(k)} (p^{(k-1)})^T r^{(k)} + \alpha^{(k)} (p^{(k-1)}) A \alpha^{(k)} (r^{(k)} + \beta^{(k)} p^{(k-1)}) \end{aligned} \quad (12)$$

Since the value of $\beta^{(k)}$, minimizing $E(x^{(k+1)})$ should satisfy the equation

$$\frac{\partial E}{\partial \beta^{(k)}}(x^{(k+1)}) = 0$$

This implies

$$-(p^{(k-1)})^T r^{(k)} + \alpha^{(k)}(p^{(k-1)})^T A r^{(k)} + \alpha^{(k)}\beta^{(k)}(p^{(k-1)})^T A p^{(k-1)} = 0$$

But $(p^{(k-1)})^T r^{(k)} = 0$, and ,therefore, the optimal value of $\beta^{(k)}$ should be

$$\beta^{(k)} = \frac{-(p^{(k-1)})^T A r^{(k)}}{(p^{(k-1)})^T . A(p^{(k-1)})} \tag{13}$$

Also, from Eq. (13)

$\beta^{(k)} (P^{(k-1)})^T . Ap^{(k-1)} + (p^{(k-1)})^T + Ar^{(k)} = 0$ implies

$(p^{(k-1)})^T A(r^{(k)} + \beta^{(k)} p^{(k-1)}) = 0$

Now, by Eq.(11), we see that

$$(p^{(k-1)})^T Ap^{(k)} = 0 \tag{14}$$

which means the vectors is A-conjugate. Since $r^{(k)}$ and $p^{(k-1)}$ are orthogonal from (11),

$(r^{(k)})^T p^{(k)} = (r^{(k)})^T r^{(k)} + \beta^{(k)} (r^{(k)})^T . p^{(k-1)}$

implies $(r^{(k)})^T . p^{(k)} = (r^{(k)})^T . r^{(k)}$

Conjugate Gradient Algorithm for the solution of Ax = b

1. *let* $A, b, k\max, x^{(0)}$ *and* ε *be given*
2. $r^{(0)} = b - Ax^{(0)}, \quad p^{(0)} = r^{(0)}$
3. $\alpha^{(k)} = \dfrac{(p^{(k)^T}) . r^{(k)}}{(p^{(k)})^T . Ap^{(k)}}$

   $x^{(k+1)} = x^{(k)} + \alpha^{(k)} . p^{(k)}$

   $r^{(k+1)} = b - Ax^{(k+1)}$

   $\beta^{(k)} = \dfrac{(r^{(k+1)})^T . r^{(k+1)}}{(r^{(k)})^T . r^{(k)}}$

   and $p^{(k+1)} = r^{(k+1)} + \beta^{(k)} . p^{(k)}$
4. *Ter* min *ate the iteration process*

   if $\dfrac{\|r\|}{\|b\|} < \varepsilon$

k = 0,1,2,…,Kmax **(15)**

The conjugate algorithm applied to solve a matrix equation Ax = b of the order n converges in at the most n iterations. However, because of rounding errors in the calculation of conjugate directions, we do not obtain the exact solution in n iteration.

Chapter 3

Application to difference methods

In order to illustrate how the iterative methods describe can be applied, we consider the Dirichlet boundary value problem.

$$\left.\begin{aligned} -U_{XX}-U_{YY}&=f(x,y),\ 0<x,y<1 \\ U(x,y)&=0 \text{ for } (x,y)\varepsilon\,\alpha\Omega \end{aligned}\right\} \qquad (1)$$

for the unit square $\Omega := \{x,y \mid 0 < x, y < 1\} \subseteq R^2$ with boundary $\alpha\Omega$

We assume f(x,y) continuous on $\Omega \cup \alpha\Omega$. Since the various methods for the solution of boundary-value problems are compared on this problem (1) is also called the Model problem. To solve (1) by means of difference method, one covers $\Omega \cup \alpha\Omega$ with grid size $\Omega_h \cup \alpha\Omega_h$.

$$\Omega_h := \{(x_i, y_j) \mid i, j = 1,2,\ldots,N\}$$

$$\alpha\Omega_h := \{(x_i,0),(x_i,1),(0,y_j),(1,y_j) \mid i,j. = 0,1,2,\ldots,N+1\}$$

where $x_i = ih$, $y_j = jh$, $i,j = 0,1,2,\ldots,N+1$

$h := 1/(N+1)$, $N > 0$ an integer.

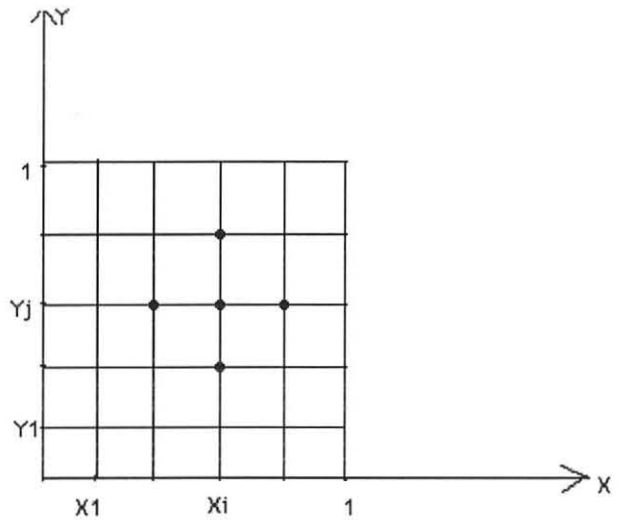


Figure (1) : The grid size Ω_h

With the further abbreviation : $U_{ij} = U(x_i,y_j)$, $i,j = 0,1,2,\ldots,N+1$

The differential operator :

$-U_{xx}-U_{yy}$ can be replaced for all $(x_i,y_j) \in \Omega_h$ by the difference operator .

$$(4U_{ij} - U_{i-1,j} - U_{i+1,j} - U_{i,j-1} - U_{i,j+1})/h^2 \qquad (2)$$

Up to the error J_{ij} .

The unknowns U_{ij} , $1 < i,j < N$ [because of the boundary conditions the $U_{ij} = 0$ are known for $(x_i,y_j) \in \alpha\Omega_h$],therefore obey a system of linear equations of the form .

$$4U_{ij} - U_{i-1,j} - U_{i+1,j} - U_{i,j-1} - U_{i,j+1} = h^2 f_{ij} + h^2 J_{ij}, (x_i,y_j) \in \alpha\Omega_h \qquad (3)$$

with f_{ij}: = $f(x_i,y_j)$.Here the errors J_{ij} of course depends on the mesh size h. Under approximate differentiability assumptions for the exact solution U, it is easy to see that $J_{ij} = o(h^2)$. For sufficiently small h one can expect that the solution Z_{ij} , $i,j = 1,2,\ldots,N$ of the linear system of equations

$$4Z_{ij} - Z_{i-1,j} - Z_{i+1,j} - Z_{i,j-1} - Z_{i,j+1} = h^2 f_{ij} \text{ ,} i,j = 1,2,\ldots,N$$

$$Z_{oj} = Z_{N+1,j} = Z_{i0} = Z_{i,N+1} = 0 \quad \text{for } i,j = 0,1,\ldots,N+1 \qquad (4)$$

Obtained from (3) by omitting the error J_{ij}, agrees approximately with U_{ij} .

To every grid point (x_i,y_j) of Ω_h there belongs exactly one component Z_{ij} of the right-hand sides $h^2 f_{ij}$ row-wise (see figure 1) in to vectors.

$Z = [z_{11},z_{21},\ldots,z_{N1},z_{12},\ldots,z_{N2},\ldots,z_{1N},\ldots,z_{NN}]^T$

$b = h^2[f_{11},\ldots,f_{N1},\ldots,f_{1N},\ldots,f_{NN}]^T$

then equation (4) is equivalent to a system of linear equations of the form

$Az = b$, with the $N^2 x N^2$ matrix

$$A = \begin{bmatrix}
4 & -1 & & & -1 & & & & & & & & & \\
-1 & \cdot & \cdot & & & \cdot & & & & & & & & \\
 & \cdot & \cdot & \cdot & & & \cdot & & & & & & & \\
 & & \cdot & \cdot & -1 & & & -1 & & & & & & \\
-1 & & & -1 & 4 & & & & \cdot & & & & & \\
 & \cdot & & & & 4 & -1 & & & \cdot & & & & \\
 & & \cdot & & & -1 & \cdot & \cdot & & & \cdot & & & \\
 & & & \cdot & & & \cdot & \cdot & \cdot & & & \cdot & & \\
 & & & & -1 & & & \cdot & \cdot & -1 & & & -1 & \\
 & & & & & \cdot & & -1 & 4 & & & & & \cdot \\
 & & & & & & \cdot & & & \cdot & & & & -1 \\
 & & & & & & & \cdot & & & \cdot & & & \\
 & & & & & & & & \cdot & & & 4 & -1 & \\
 & & & & & & & & & -1 & & -1 & \cdot & \cdot \\
 & & & & & & & & & & \cdot & & \cdot & \cdot & -1 \\
 & & & & & & & & & & & -1 & & -1 & 4
\end{bmatrix} \qquad (5)$$

$$A = \begin{bmatrix}
A_{11} & A_{12} & o & o & \cdot & \cdot & & \cdot & & 0 \\
A_{21} & \cdot & \cdot & & & & & & & 0 \\
0 & \cdot & \cdot & \cdot & & & & & & \cdot \\
0 & & \cdot & \cdot & \cdot & & & & & \cdot \\
\cdot & & & \cdot & \cdot & \cdot & & & & \cdot \\
\cdot & & & & \cdot & \cdot & & \cdot & & 0 \\
\cdot & & & & & \cdot & & \cdot & & A_{N-1,N} \\
0 & \cdot & \cdot & \cdot & 0 & 0 & A_{N,N-1} & & & A_{NN}
\end{bmatrix}$$

A is partitioned in a natural way in to blocks A_{ij} of order N, which are induced by partitioning of the point (x_i,y_j) of Ω_h in to horizontal rows (x_1,y_j), (x_2,y_j), … , (x_N,y_j) .

The matrix A is quite sparse. In each row, at most five elements are different from zero. For this reason, in the execution of an iteration step $Z^{(i)} \rightarrow Z^{(i+1)}$ of, say the Jacobi method or the Gauss-Seidel method one requires only about $5N^2$ operations
[1 operation = 1 multiplication or division + 1 addition], but if the matrix A were dense N^4 operations per iteration step would be required. Let compare this expenditure with that of direct method for solving Az = b .

If we compute a triangular decomposition

$A = LL^T$ of A, say cholesky method, then L would be an N^2xN^2 lower triangular matrix of the form .

$$L = \begin{bmatrix} * & & & & & & \\ . & . & & & & & \\ . & & . & & & & \\ .* & & & . & & & . \\ & . & & & . & & \\ & & . & & & . & \\ & & & . & & & . \\ & & & & * & . \; . & * \end{bmatrix}$$

The computation of L alone requires approximation $(¼)N^2$ operations. Since the Jacobi method requires approximately $(N+1)^2$ iterations[over relaxation method : N+1 iterations] in order to obtain a result accurate to 2 decimal places, the cholesky method would be less expensive than Jacobi method. The main difficult with cholesky method, however, lies in fact today's memory capacities the storage of the approximately N^3 non-vanishing elements of L requires too much spaces. Here lies the advantage of iterative methods. Their storage requirement is only of the order of magnitude N^2.

3.1. Block Iterative Method

As, the example of the previous section shows, the matrices A arising in the application of difference methods to partial differential equations often exhibit a natural block structure.

$$A = \begin{bmatrix} A_{11} & . & . & . & A_{1N} \\ . & & & & . \\ . & & & & . \\ . & & & & .. \\ A_{N1} & . & . & . & A_{NN} \end{bmatrix}$$

Where A_{ii} are square matrices. If, in addition, all A_{ii} are non-singular, it seems natural to introduce block iterative methods relative to the given partition α of A, in the following way.

Let consider the decomposition

$$A = D_\alpha - E_\alpha - F_\alpha$$

$L_\alpha := (D_\alpha)^{-1}E_\alpha$ and , $U_\alpha := (D_\alpha)^{-1}F_\alpha$.

$$D_\alpha = \begin{bmatrix} A_{11} & 0 & . & . & o \\ 0 & A_{22} & & & . \\ . & & & & . \\ . & & & & 0 \\ 0 & . & . & 0 & A_{NN} \end{bmatrix} \qquad E_\alpha = -\begin{bmatrix} 0 & . & . & . & 0 \\ A_{21} & 0 & & & . \\ . & & . & & . \\ . & & & . & . \\ N1 & . & . & A_{N,N-1} & 0 \end{bmatrix}$$

$$F_\alpha = -\begin{bmatrix} o & A_{12} & & & A_{1N} \\ . & . & & & \\ . & & . & & \\ . & & & . & A_{N,N-1} \\ 0 & . & . & . & 0 \end{bmatrix}$$

Now, we obtain the block Jacobi method (block total- step method) for the solution of $Ax = b$ by choosing $M = D_\alpha$ and $N = E_\alpha + F_\alpha$

Thus the iteration algorithm

$D_\alpha x^{(k+1)} = (E_\alpha + F_\alpha)x^{(k)}$

This implies $A_{ii}x_i^{(k+1)} = bi - \sum_{j \neq i} A_{ij}x_j^{(k)}$, $k = 0,1,\ldots$ $i = 1,2,\ldots,N$ (1)

Here, the vector $x^{(k)}$, be are of course partitioned similar to A. In each step $x^{(k)} \rightarrow x^{(k+1)}$ of the method ,we must solve n systems of linear equations of the form $A_{ii}z = y$, $i = 1,2,\ldots,N$. This is accomplished by first obtaining, a triangular decomposition $A_{ii} = L_iU_i$ of the A_{ii}, and the reducing $A_{ii}z = y$ to the solution of two triangular systems of equations

$$L_iw = y, \; U_iz = w.$$

For the efficiency of the method it is essential that the A_{ii} be simply structured matrices for which the triangular decomposition is easily carried out. This is the case,e.g.,for the matrix A in (5) of (3.1) of the model problem. Hence, A_{ii} are positive definite tri-diagonal NxN matrices.

$$A_{ii} = \begin{bmatrix} 4 & -1 & & & \\ -1 & . & . & & \\ & . & . & . & \\ & & . & . & -1 \\ & & & -1 & 4 \end{bmatrix}$$

The rate of convergence of (1) determined by the spectra radius of $\delta(J_\alpha)$ of the matrix

$$J_\alpha = D^{-1}(E_\alpha + F_\alpha) = L_\alpha + U_\alpha$$

Similarly, we can define, a block Gauss-Seidel Method

(block single-step method),through the choice

$$M = D_\alpha - E_\alpha \quad \text{and} \quad N = F_\alpha$$

Or explicitly

$$A_{ii}x_i^{(k+1)} = b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{N} A_{ij}x_j^{(k)} \;, \; i = 1,2,\ldots,N, \quad k = 0,1,2,\ldots \qquad (2)$$

Here, again systems of equation with the matrices A_{ii} need to be solved in each iteration step.

As, in section (2.3), we can also introduce block relaxation methods through the choice

$$M = (1/\omega) D_\alpha (I - \omega L_\alpha)$$

Let $\bar{x}_i^{(k+1)}$ be the solution of (2); then

$$x_i^{(k+1)} = \omega(\bar{x}_i^{(k+1)} - x_i^{(k)}) + x_i^{(k)}, \; i = 1,2,\ldots,N$$

Now, of course

$$T_\alpha(\omega) = (I - \omega L_\alpha)^{-1}[(1 - \omega)I + \omega U_\alpha]$$

One expects intuitively that the block methods will converge faster with increasing coarseness of the block partition α of A.

For the coarsest partition α of A in to a single Block, e.g., the iterative method converges after just one step. It is then equivalent to direct method. The reduction in the number of iterations is compensated, to certain extent, by larger computational work for each individual iteration step.

For the most common partitions, in which A is block tri-diagonal and the diagonal blocks usually tri-diagonal, however, the computational work involved in block methods is equal to that in ordinary methods. In these cases, block methods bring real advantages

3.2. Applications

Example 1: Study of heat distribution on sides of an oven and solve by Gauss-Seidel Algorithm.

Statements of the Problem: Let us consider the domain of the problem to be the vertical cross-section of an oven as shown in the Fig.(1) . Let Γ_e *and* Γ_e denote the interior and exterior sides of the oven respectively. Then, for any point (x,y) in the domain of the problem, the temperature T(x,y) at this point in the continuous system satisfies the Laplace equation.

$$\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0$$

along with the following boundary conditions:

$$T(x,y) = \theta_e \quad \text{on } \Gamma_e \quad \text{and}$$

$$T(x,y) = \theta_i \quad \text{on } \Gamma_i$$

1) Write an algorithm to calculate the temperature T (x, y) at all the interior grids points corresponding to sub-division of the oven as shown in Fig.(2). Assume that the length of sides of the region is L_1, L_2, L_3, and L_4

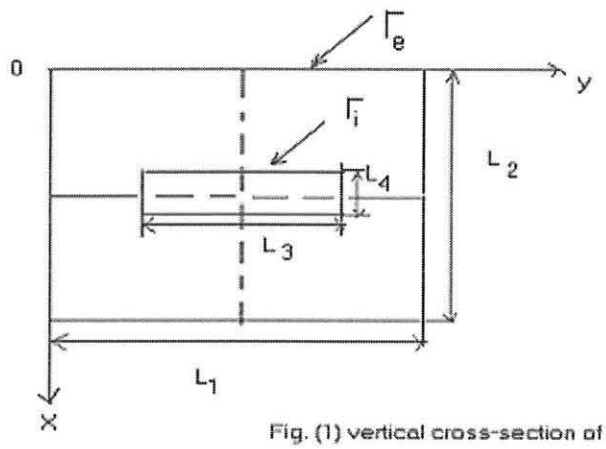


Fig. (1) vertical cross-section of an oven

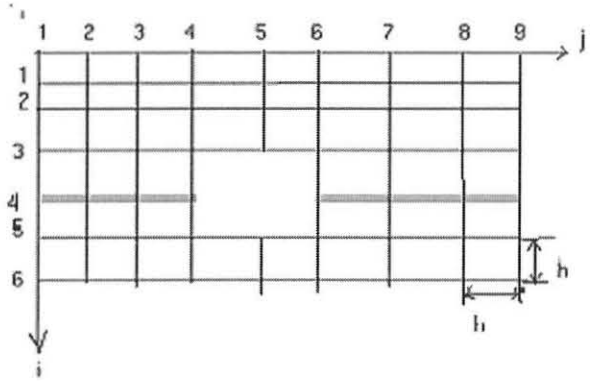


Fig. (2) The domain with 26 interior grid points

Method of the solution:

Let us consider the continuous Problem (P_1) defined by: Find T(x,y) such that

$$\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0 \qquad (1)$$

Satisfying the bounder conditions

$$T(x,y) = \theta_e \quad \text{on } \Gamma_e \quad \text{and}$$

$$T(x,y) = \theta_i \quad \text{on } \Gamma_i$$

Now, corresponding to the above subdivisition Fig.(2) of the domain with 26 interior points, the problem (P_1) can be approximated by a discrete problem (P_2), leading to the solution of the following systems

$$\frac{T_{i+1,j} + T_{i,j+1} - 4T_{i,j} + T_{i-1,j} + T_{i,j-1}}{h^2} = 0, \quad for \quad all \quad (i,j) \varepsilon N. \quad (2)$$

$T_{ij} = \theta_e$ on Γ_e *and* $T_{ij} = \theta_i$ *on* Γ_i

Where N denotes the set of interior points. i.e $\{ (i,j) \mid (i,j) \notin (\Gamma e, \Gamma_i)\}$

And from (2) we obtain the following systems of equations, namely

$T_{32} + T_{23} - 4T_{22} + T_{21} + T_{12} = 0$, when i=2,j=2

$T_{33} + T_{24} - 4T_{23} + T_{22} + T_{13} = 0$, when i=2, j=3

$T_{42} + T_{33} - 4T_{32} + T_{31} + T_{22} = 0$, when i=3,j=2

$T_{43} + T_{34} - 4T_{33} + T_{32} + T_{23} = 0$, when i=3,j=3

.

.

.

$T_{78} + T_{67} - 4T_{68} + T_{69} + T_{58} = 0$, when i=6,j=8

Note that $T_{21} = T_{12} = T_{13} = T_{31} = \ldots = T_{69} = \theta_e$

And $T_{34} = T_{35} = T_{44} = \ldots = T_{56} = \theta i_i$

Rewriting the above equations in matrix for, we get

$$\begin{bmatrix} 4 & -1 & -1 & 0 & . & . & . & 0 \\ -1 & 4 & 0 & -1 & 0 & .. & .. & 0 \\ -1 & 0 & 4 & -1 & 0 & -1 & & .0 \\ 0 & -1 & -1 & 4 & 0 & & -1 & 0 \\ . & . & & & & & & . \\ . & . & & & & & & . \\ . & . & & & & & & . \\ 0 & 0 & 0 & 0 & .. & . & -1 & 4 \end{bmatrix} \begin{bmatrix} T_{22} \\ T_{23} \\ T_{32} \\ T_{33} \\ . \\ . \\ . \\ T_{67} \end{bmatrix} = \begin{bmatrix} 2\theta_e \\ \theta_e \\ \theta_e \\ \theta_i \\ \\ \\ \\ 2\theta_i \end{bmatrix} \qquad (3)$$

Now, we can compute the temperature by using the following in put values

$$\theta_e = 50, \theta_i = 1150, \quad L_1 = 60, \quad L_2 = 80, \quad L_3 = L_4 = 20$$

In summary, the continuous problem (p_1) defined by (1) is approximated by the discrete problem (p_2) defined by the system (2)

For the subdivision as shown in Fig.(2) the approximated linear system (2) leads to the matrix equation (3), and one should solve a matrix equation of order 26. While solving (2) (or(3)) by Gauss-seidel method. Note that at each row, one of the variables is a function of the other. During calculation, if we separate the central point at each of the molecule then the Gauss-seidel method will lead to the following simple algorithm.

$$T_{ij} = (T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1})/4 \qquad (4)$$

Remark: It is important to note that in algorithm (4) it is not necessary to remember explicitly the matrix equation (3). It is sufficient to store the temperature T_{ij} at each of the grid points.

Finally, we remark that the values of the indices corresponding to the sides of the domain are determined from Fig. (3)

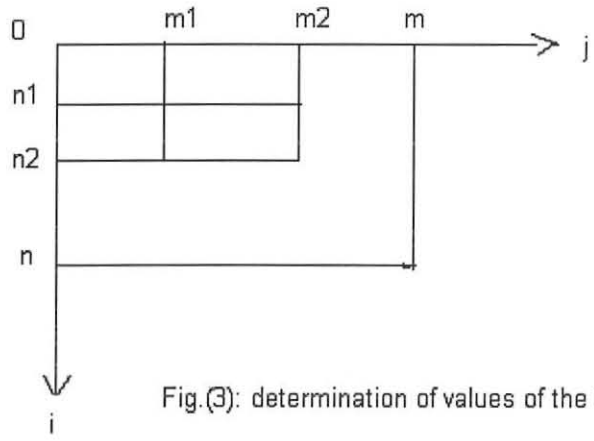


Fig.(3): determination of values of the indices corresponding to the sides of the domain.

Conclusions:

The method converges in 34 iterations for the initial arbitrary guess $\theta = 100$ at the interior points of the domain (Fig.(2)). A better choice of the initial guess will lead to reduction in the number of iteration.

Example 2: Calculation of the deflection of a plate under loading by Gauss-Seidel method.

Statement of the problem:

Consider a square plate simple supported on the boundary under the load function g (x, y). Fig.(4)

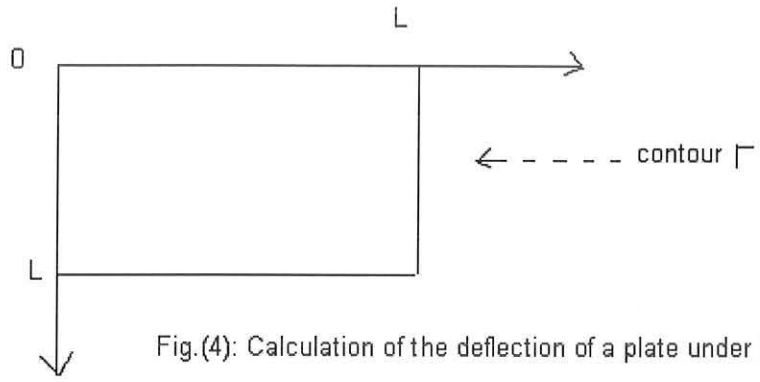


Fig.(4): Calculation of the deflection of a plate under loading

It can be verified that the moment w(x,y) and the displacement u(x,y) along the axis OZ perpendicular to the XOY plane satisfy the poisson equation.

$$\frac{\partial w(x,y)}{\partial x^2} + \frac{\partial^2 w(x,y)}{\partial y^2} = g(x,y)/D \qquad (1)$$

with w(x,y) = 0 on Γ and ,

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = w(x,y) \qquad (2)$$

such that u(x,y) = 0 on Γ

Where $D = Et^2/(12(1-v^2))$, it is rigidy of the plate.

t = thickness of the plate (mm)

v = 0.3, the poisson coefficients (dimensionless)

E = the young's modulus (d_a .N/mm^2)

L = length of the plate (mm)

g = load function (d_aN)

1) Write the algorithm to evaluate the deflection U and the moment w at each grid point of the following subdivision in Fig.(5)

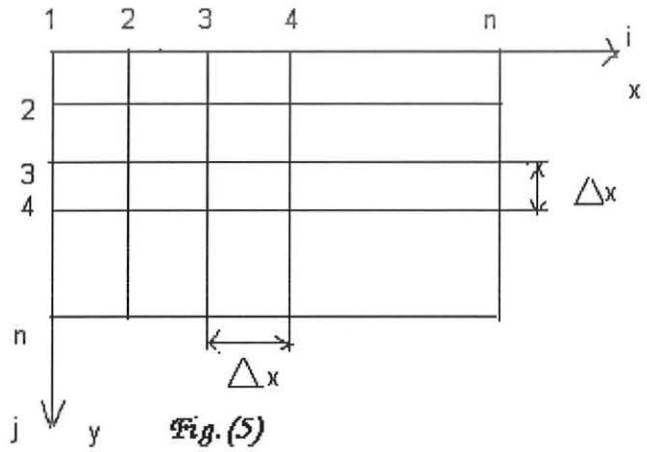


Fig.(5)

Method of the solution:

Let (x_i,y_i) be the grid points of the domain (see Fig.(4)). Then $x_i = (i-1)h$, and $y_i = (i-1)h$, for all $i = 1,2,...,n$.

Let us denote $w_{ij} = w(x_i,y_j)$

$g_{ij} = g(x_i,y_j)$

$u_{ij} = u(x_i,y_j)$

Then, corresponding to Eq.(1) and (2) are approximated as

$$(w_{i,j+1} + w_{i,j-1} - 4w_{ij} + w_{i-1,j} + w_{i+1,j})/h^2 = g_{ij}/D \qquad (3)$$

$w_{ij} = 0$ on Γ for all $i,j = 2,3,...,n-1$

$$(u_{i,j+1} + u_{i,j-1} - 4u_{ij} + u_{i-1,j} + u_{i+1,j})/h^2 = w_{ij} \qquad (4)$$

$u_{ij} = 0$ on Γ for all $i,j = 2,3,...,n-1$

The solution of Eqs.(1) and (2) at the grid point (x_i,y_j) (see Fig.(4)) are approximated by solving successively the linear systems (3) and (4) .

The algorithm for the solution of (3) by Gauss-Seidel method will be written as

$$w^{(n+1)}_{ij} = \tfrac{1}{4}[\,(w^{(n)}_{i,j+1} + w^{(n+1)}_{i,j-1} + w^{(n+1)}_{i-1,j} + w^{(n)}_{i+1,j} - h^2 g_{ij})/D\,]$$

for all $i,j = 2,3,...,n-1$

$w_{ij} = 0$ on Γ

Similarly, for the solution of (4) the algorithm will become

$$u^{(n+1)}_{ij} = 1/4(u^{(n)}_{i,j+1} + u^{(n+1)}_{i,j-1} + u^{(n+1)}_{i-1,j} + u^{(n)}_{i+1,j} - h^2 w_{ij})$$

for all $i,j = 2,3,...,n-1$

$u_{ij} =$ on Γ

CONCLUSION ON ITERATIVE METHODS

Iterative methods are generally preferred for large systems of linear equations defined by $Ax = b$, where the matrix A is sparse, because they do not modify the matrix A and, indeed, in a number of applications A is sparse. More precisely the matrix A appears with certain structure like tri diagonal, pent diagonal, etc which allows us not to store explicitly A but assures practical convergence.

For problems of small size, the Gauss-Seidel method is preferred over Jacobi method because it needs less memory size and often converges rapidly.

Generally, the convergence of relaxation method is faster than that of Gauss-Seidel method even if the optimal factor ω is adjusted experimentally.

References

1. **Anthony Ralston and Philip Rabinowitz. (2001)**
   A first course in numerical analysis
   Second edition, McGraw-Hill, New York, USA
2. **James L.buchann and Petter R.Turner (1992)**
   Numerical method and analysis.
   McGraw-Hill, Inc. New York, USA.
3. **A.Gourdin and M.Boumarat** (1996)
   Applied numerical methods
   Prentice-Hall, India.
4. **Kendall E.Atkinson (1978)**
   An introducation to numerical analysis.
5. **Richard L. Burden, J.Douglas Faires, Albert, C.Reynolds (1981)**
   Numerical Analysis
6. **Store, R.Bulirsch (1980)**
   Introduction to numerical Analysis
   Springer-Verlag