1-1-2006

Numerical solution of boundary value problems using a variational formulation

Virendra I Jaiswal
University of Nevada, Las Vegas

NUMERICAL SOLUTION OF BOUNDARY VALUE PROBLEMS

USING A VARIATIONAL FORMULATION

by

Virendra I. Jaiswal

Bachelor of Science
Maharaja Sayaji-Rao University, Baroda
2000

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Mathematical Sciences
Department of Mathematical Sciences
College of Sciences

Graduate College
University of Nevada, Las Vegas
December 2005

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval

The Graduate College
University of Nevada, Las Vegas

November 14, 2005

The Thesis prepared by

Virendra I. Jaiswal

Entitled

Numerical Solution of Boundary Value Problem Using a Variational Formation

is approved in partial fulfillment of the requirements for the degree of

Master of Science

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

ABSTRACT

Numerical Solution of Boundary Value Problems using a Variational Formulation

by

Virendra I. Jaiswal

Dr. Rohan Dalpatadu, Examination Committee Chair
Associate Professor of Mathematics
University of Nevada, Las Vegas

In this thesis unique solutions of certain Boundary Value Problems are approximated by first converting them into their variational formulation and obtaining linear systems of equations by either using finite element method or discretization, then using the Gauss-Seidel iterative method to solve the resulting systems.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my Graduate Thesis Advisor, Dr. Rohan Dalpatadu, for all of the help and guidance that he has given me throughout my studies at UNLV. I would also like to thank Dr. Satish Bhatnagar for getting me interested in attending graduate school.

I would also like to thank Dr. Singh, Dr.Xin Li, Dr.Ding, and all of my other professors for devoting their time to teaching, for knowledge is truly the best gift that anyone could receive.

Also, I would like to thank Nancy, LaVerne Boe and all of the Mathematics Department staff for making graduate school not only interesting but also fun.

Last, but not least I would like to thank my parents and my friends for all of the love and support that they have given me throughout the years

CHAPTER 1

INTRODUCTION

The Finite Element Method can be described as follows: suppose that the problem to be solved is given in variational form, and we are required to find the function u, which minimizes a given functional I. This would lead to a differential equation for u (The Euler-Lagrange Equation), and normally an exact solution would be quite difficult to obtain, and so an approximation would be necessary. On the other hand, we may have a boundary value problem and we will try to minimize a suitable functional, to obtain the solution to the boundary value problem. The Rayleigh-Ritz-Galerkin idea is to choose a finite number of "Trial" functions $\phi_1, \phi_2, \ldots, \phi_N$ and among all their linear combinations $\sum C_N \phi_N$, to find the one which is minimizing. This procedure yields a finite number of simultaneous algebraic equations, whose solution would give us the approximation we seek. Theoretically, the minimizing process automatically seeks out the combination which is closest to the solution.

The procedure starts by subdividing the structure into smaller pieces, usually sub interval in 1-Dimension and triangles or rectangles in 2-Dimension, then the "Trial" functions are given a simple form-usually they are polynomials of at most third degree.. The accuracy can be increased by retaining the same polynomials and refining the subdivision, instead of including more and more complex "Trial" functions.

This paper will illustrate the setting of two algorithms, using the Finite Element Method, with piecewise linear polynomials, and rectangular elements (for the partial differential equation), to approximate the solutions of a linear boundary value problem in one dimension and an elliptical boundary value problem in two dimensions.

The Problem

The problem is to approximate the solution of

$$Lu = f \tag{1.3.1}$$

Where L is a linear operator acting on a certain class of functions, those functions which are twice continuously differentiable and satisfy certain boundary conditions. Under certain conditions, we find that L is a one-to-one operator onto a certain class of functions. The problem is to determine L^{-1}, in order to find $u = L^{-1}f$.

If (v, v) is the associated inner product, then the functional

$$I(v) = (Lv, v) - 2(f, v) \tag{1.3.2}$$

is related to (1.3.1) in the following way:

$I(v)$ is minimized at $v = u$, only if the first variation vanishes there, and the condition for this is the Euler- Lagrange equation $Lu = f$.

The problems of inverting L and minimizing I are equivalent; they produce the same solution u. In our problem we will minimize I obtain the solution u.

The Finite Element Method

In the finite element method we determine approximations of the form

$$U(t) = \sum_{i=0}^{N} C_i(t)\phi_i(t)$$

This method involves three basic steps:

1. Choose a finite-dimensional space S. For example, subdivide the domain into a union of elements by the uniform mesh $0 = x_0 < x_1 < \ldots < x_N = 1$, where $x_i = i\Delta x$. Then let

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{\Delta x}, x_{i-1} \le x \le x_i \\ \dfrac{x_{i+1} - x}{\Delta x}, x_i \le x \le x_{i+1} \\ 0 \end{cases}$$

The ϕ_i are the so-called hat functions. The space S is then choose to be the space spanned by $\{\phi_i(x) \mid 0 \le i \le N\}$. This choice yields a space with desirable approximation theoretic properties.

2. Approximate $u(t)$ by

$$U(t) = \sum_{i=0}^{N} C_i(t)\phi_i(t)$$

3. Solve the set of linear equation generated in step 2.

The Finite Difference Method

An elementary approach to finite difference method is provided by Taylor's theorem, which we state as follows:

Let $\phi \in C^{n+1}[a,b]$, where $C^{n+1}[a,b]$ denotes the class of

functions that are $n+1$ times continuously differentiable on the interval [a, b]. Then there exists a number $\xi, a < \xi < b$ such that,

$$\phi(b) = \sum_{i=0}^{n} \frac{\phi^{(i)}(a)}{i!}(b-a)^i + R_n$$

where $$R_n = \frac{\phi^{(n+1)}(\xi)(b-a)^{n+1}}{(n+1)!}$$

From this theorem it is easy to justify the following three approximations to $d\phi/dx(a)$:

1. Forward difference : $\frac{\phi(a+h)-\phi(a)}{h}$
2. Backward difference: $\frac{\phi(a)-\phi(a-h)}{h}$
3. Centered difference: $\frac{\phi(a+h)-\phi(a-h)}{2h}$, where $h>0$

Indeed, if ϕ is sufficiently smooth, then 1 and 2 approximate $d\phi/dx$ with an error of $O(h)$, while 3 approximates $d\phi/dx$ with an error of $O(h^2)$. We say that 1 or 2 is a first order, and 3 is a second-order, approximation of $d\phi/dx$.

The second derivative $d^2\phi/dx^2$ can be approximated similarly using the formula

4. Centered difference: $\frac{\phi(a+h)-2\phi(a)+\phi(a-h)}{h^2}$

which is also a second order as $h \to 0$. These approximations are sufficient for our purpose, although it is worth mentioning that higher-order finite difference formulas can be derived as well as finite difference approximation of higher-ordered derivatives.

If L_h is a finite difference approximation to a differential operator L and

$$(L_h - L)[\phi](a) \to 0 \qquad \text{as} \quad h \to 0,$$

then $L_h[\phi]$ is said to be a consistent finite difference approximation to $L[\phi]$ at a.

For example,

$$L_h[\phi] \equiv \frac{\phi(a+h) - \phi(a)}{h}$$

is a consistent approximation to $L[\phi] \equiv d\phi / dx$ at a.

Finite difference methods for solving initial – boundary value problems such as to determine approximations at a finite number of points in the domain and involve four basic steps:

1. Subdivide the domain, for example by the uniform mesh $x_0 < x_1 < \ldots < x_N$, where the mesh points are $x_j = j\Delta x$ and the mesh gauge is $\Delta x = 1/N$.
2. Approximate the differential equation at each mesh point x_j by replacing derivatives by appropriately chosen finite difference approximations.
3. Impose the boundary and initial conditions on the system generated in step 2
4. Solve the finite difference equations generated in step 2 and 3.

Hence we replace a differential equation and any auxiliary conditions by a system of linear algebraic equations. The solution of the latter constitutes an approximation at mesh point to the former.

CHAPTER 2

A PROBLEM IN ONE DIMENSION

The Ritz Method

Definition 2.1.1:- $\|\bullet\|_n$ Norm

The $\|\bullet\|_n$ norm, for a real function f from $[0, \pi]$ having n derivatives is given by

$$\|f\|_n^2 = \int_0^\pi \left[(f^{(n)})^2 + \ldots + (f')^2 + f^2\right]. \qquad (2.1.1)$$

Definition 2.1.2

A function f: $[0,\pi] \to \mathbb{R}$ is said to have finite energy with respect to the $\|\bullet\|_n$ norm, if $\|f\|_n^2 < \infty$.

The linear differential equation

$$-(pu')' + qu = f \qquad p \geq \min p > 0,\ q \geq 0 \qquad (2.1.2)$$

with the boundary conditions

$$u(0) = 0, u(\pi) = 0 \qquad (2.1.3)$$

where p has a piecewise continuous first derivative on $[0, \pi], q$ is continuous on $[0, \pi]$ and, f is piecewise continues and has finite energy with respect to the $\|\bullet\|_n$ norm, has a unique solution in $[0, \pi]$.

Definition 2.1.3

H_B^2 is the space of real valued functions u on $[0, \pi]$, such that u satisfies (2.1.3) u is twice differentiable and has finite energy with respect to $\|\bullet\|_2$ norm.

Definition 2.1.4

H_B^1 is the space of real valued continuous functions v on [0, π], such that v is piecewise differentiable, $v(0) = 0$, and v has finite energy with respect to the $\|\bullet\|_1$ norm, H_B^1 is called the space of admissible functions.

Definition 2.1.5

H^0 is the space of real valued functions f on [0, π], such that f has finite energy with respect to the $\|\bullet\|_0$ norm.

Theorem

L in (1.3.1) is a one –to – one mapping from H_B^2 onto H^0, so that for each f $\in$ H^0, (2.1.2) has a unique solution u in H_B^2 if L is the operator defined by

$$Lu = -(pu')' + qu, \quad u \in H_B^2 \tag{2.1.4}$$

and p', f are continuous on [0, π].

Furthermore, there exists a constant C such that for a given f, the corresponding u satisfies

$$\|u\|_2 \le C\|f\|_0. \tag{2.1.5}$$

Proposition 2.1.1

The functional I(v) in (1.3.2.), for v$\in$ H_B^2, is uniquely minimized by the solution u to (2.1.2.) and (2.1.3.). Further I (v) can be expressed in the form

$$I(v) = \int_0^{\pi} [p(v')^2 + qv^2 - 2fv], \quad v \in H_B^2 \tag{2.1.6}$$

Proof:

Let ε be real and v$\in$ H_B^2. Then

$I(u+\varepsilon v) = \{L(u+\varepsilon v),\ u+\varepsilon v\} - 2(f,\ u+\varepsilon v)$

$= (Lu, u) - 2(f, u) + 2\varepsilon (Lu, u) - 2\varepsilon (f, v) + \varepsilon^2 (Lv, v)$

$= I(u) + \varepsilon^2 (Lv, v) + 2\varepsilon (Lu - f, v)$

$= I(u) + \varepsilon^2 (Lv, v)$

(Note that, $(Lu, v) = (Lv, v) = \int_0^{\pi} [pu'v' + quv]$, since $v \in H_B^2$, by use of integration by parts.)

$$(Lv, v) = \int_0^{\pi} [p(v')^2 + qv^2]$$

Since $(Lv, v) \geq 0$, with quality if and only if $v \equiv 0$, we must have the first part of the proposition.

Now $I(v) = (Lv, v) - 2(f, v)$

$$= \int_0^{\pi} [p(v')^2 + qv^2 - 2fv],$$

Since I(v) no longer contains second derivatives, we try to minimize it over the space H_B^1.

Proposition 2.1.2

The function u in H_B^1 which minimizes I (v) over H_B^1 will satisfy the boundary condition $u(\pi) = 0$ and will be the solution to (2.1.2.) and (2.1.3.), if it is twice differentiable.

Proof:

For any real ε and $v \in H_B^1$,

$I(u) \leq I(u + \varepsilon v)$

$$= I(u) + 2\varepsilon \int_0^{\pi} [pu'v' + pqv - fv] + \varepsilon^2 \int_0^{\pi} [p(v')^2 + qv^2].$$

Since ε is arbitrary, we must have

$$0 = \int_0^{\pi} [pu'v' + pqv - fv]$$

$$= \int_0^{\pi} [-(pu')' + qu - f]v \; + pu'v\,|_0^{\pi}$$

$$= \int_0^{\pi} [-(pu')' + qu - f]v \; + \; p(\pi)\, u'(\pi)\, v(\pi).$$

This will hold only if both $u(\pi) = 0$ and $-(pu')' + qu = f$, since v is arbitrary in H_B^1

Remark 1

The quadratic I(v) taken over H_B^2 is represented by a parabolic in infinite dimension, which has "holes" on the surface. By extending H_B^2 into the closed space H_B^1, we have simply filled in these "holes".

Remark 2

When the coefficient P(x) is discontinuous at $x=x_0$, the solution u is no longer in H_B^2. The minimum value of I(v) would have been the same in both spaces, but within the space H_B^2, there was no function u for which I attained this minimum.

Remark 3

Let N be a positive integer and $h = \frac{\pi}{N}$. Then S^h defined below is a finite dimensional subspace of H_B^1, and its elements (denoted by v^h) are called trial functions. S^h is also referred to as the trial space

Definition 2.1.6

A Ritz approximation is the function $u^h \in S^h$ which minimizes I(v) over S^h. Let S^h be the space of functions which are linear over each subinterval [(j-1)h, jh] and continuous at the nodes 0 and jh, for j= 1,2, … , N, zero at x= 0. For j = 1,2, … , N define $\phi_j^h \in S^h$ as $\phi_j^h(ih) = \delta_{ij}$. Then for $v^h \in S^h$,

$$v^h = \sum_{j=1}^{N} q_j \phi_j^h(x) \qquad (2.1.7)$$

where $q_1, \ldots, q_N$ are real constants.

Notice that q_j is the value of v^h at the j^{th} node x = jh, and $(\phi_i^j, \phi_j^h) = 0$ if $|\, i-j \,| > 1$.

In our problem, if we take p and q as constant, $v^h = \sum_{j=1}^{N} q_j \phi_j^h$ and computer the second degree terms of the integral $I(v^h)$ over one subinterval at a time, we obtain

$$\int_{(j-1)h}^{jh} [(v^h)']^2 = \frac{(q_j - q_{j-1})^2}{h}$$

$$\int_{(j-1)h}^{jh} (v^h)^2 = (v^h)^2 = \frac{h(q_j^2 - q_j q_{j-1} + q_{j-1}^2)}{3}$$

and

$$\int_0^{\pi} [(p(v^h)')^2 + q(v^h)^2 \doteq \sum_{j=1}^{N} [\frac{p(q_j - q_{j-1})^2}{h} + \frac{hq}{3}(q_j^2 - q_j q_{j-1} + q_{j-1}^2)] \qquad (2.1.8)$$

where $q_0 = 0$.

This is not a particularly convenient form of the result. We would prefer the matrix form $q^T K q$, where $q = (q_1, \ldots, q_N)^T$, because it is the matrix K which we need. The reason being $I(v^h)$ is quadratic in $q = (q_1, \ldots, q_N)^T$ and the minimum for $I(v^h) = q^T K q - 2F^T q$ occurs at the vector Q, determined by KQ = F. This is the system we shall have to solve, and we need to know only the matrix K and the vector F.

Since we will be using a numerical qudrature to evaluate the integral over each subinterval, we can take the average values of the functions p(x), q(x) and f(x) over each subinterval.

Then for j = 1,2, … , N

$$\int_0^{\pi} (p(v^h)')^2 \doteq \frac{p(j)}{h}(q_j - q_{j-1})^2$$

$$= [q_{j-1}, q_j] \bullet \frac{p(j)}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} q_{j-1} \\ q_j \end{bmatrix}$$

where p(j) is the average of p(x) on [(j-1)h, jh].

we can write this as

$$\int_{(j-1)h}^{jh} (p(v^h)')^2 \doteq [q_{j-1}, q_j]\, k_1^j \begin{bmatrix} q_{j-1} \\ q_j \end{bmatrix}.$$

The matrix k_1^j is the element stiffness matrix. For j = 1,2, … , N

$$\int_{(j-1)h}^{jh} q(v^h)^2 \doteq \frac{hq(j)}{3}(q_j^2 - q_j q_{j-1} + q_{j-1}^2)$$

$$= [q_{j-1}, q_j] \frac{hq(j)}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} q_{j-1} \\ q_j \end{bmatrix}$$

where q(j) is the average value of q(x) on [(j-1)h, jh].

We can write this as

$$\int_{(j-1)h}^{jh} q(v^h)^2 \doteq [q_{j-1}, q_j] k_0^j \begin{bmatrix} q_{j-1} \\ q_j \end{bmatrix}. \qquad (2.1.10)$$

The matrix k_0^j is the element mass matrix, thus the matrix k_1 associated with $\int_0^{\pi} (p(v^h)')^2$ is

$$k_1 = \frac{1}{h} \begin{bmatrix} p(1) & & \\ & & \\ & & \end{bmatrix} + \begin{bmatrix} p(2) & -p(2) & \\ -p(2) & p(2) & \\ & & \end{bmatrix} + \cdots\cdots + \begin{bmatrix} & & \\ & p(N) & -p(N) \\ & -P(N) & p(N) \end{bmatrix}$$

$$= \frac{1}{h}\begin{bmatrix} p(1)+p(2) & -p(2) & 0..... & & & \\ -p(2) & p(2)+p(3) & -p(3) & 0..... & & \\ 0 & -p(3) & p(3)+p(4) & -p(4) & 0.. & \\ & \ddots & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \ddots & 0 \\ & & .....0 & -p(N-1) & p(N-1)+p(N) & -p(N) \\ & & & .....0 & -p(N) & p(N) \end{bmatrix}$$

Similarly, the matrix k_0 associated with $\int_0^{\pi} q(v^h)^2$ is

$$k_0 = \frac{6}{h}\begin{bmatrix} 2q(1)+2q(2) & q(2) & 0..... & & \\ q(2) & 2q(2)+2q(3) & q(3) & 0..... & \\ 0 & q(3) & 2q(3)+2q(4) & q(4) & \\ & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & 0 \\ & .....0 & q(N-1) & 2q(N-1)+2q(N) & q(N) \\ & & .....0 & q(N) & 2q(N) \end{bmatrix}$$

The relation between the matrices k_0, k_1 and the interregna's are:

$$\int_0^{\pi} (p(v^h)')^2 \doteq q^T K_1 q \qquad (2.1.11)$$

and

$$\int_0^{\pi} q(v^h)^2 \doteq q^T K_0 q \qquad (2.1.12)$$

If K= $K_0 + K_1$, then

$$\int_0^{\pi}[(p(v^h)')^2 + q(v^h)^2] \doteq q^T K q \qquad (2.1.13)$$

The matrix K_1 is the global stiffness matrix and K_0 is the mass matrix.

To compute the integral $\int_0^{\pi} fv^h$, we replace f by its average value f(j) in each subinterval

$[(j-1)^h, j^h]$ to obtain $\int_{(j-1)h}^{jh} fv^h \doteq \frac{hf(j)}{2}(q_{j-1} + q_j)$.

If the vector F is given by $F= (f_1,f_2, \dots , f_N)^T$ where $f_j = (f(j) + f(j+1)) \cdot \frac{h}{2}$

for j = 1,2, … ,N-1

and $f_N = f(N)$, then the relation between F and $\int_0^{\pi} fv^h$ is given by :

$$\int_0^{\pi} fv^h = F^T q. \qquad (2.1.14)$$

combining (2.1.13) and (2.1.14) we obtain

$$I(v^h) = q^T Kq - F^T q. \qquad (2.1.15)$$

This will have a minimum at q = Q only if

$$KQ = F. \qquad (2.1.16)$$

An equation (2.1.16) is referred to as the finite element equation. The matrix K which was generated is positive definite and therefore invertible. Since p>0, and $q \geq 0$,

$$q^T Kq = \int_0^{\pi} [p(\Sigma q_j \phi_j')^2 + q(\Sigma q_j \phi_j)^2]$$

can be zero only if $\Sigma q_j \phi_j'$ is identically zero, and this happens only if $q_j = 0$ for each j.

Once the entries for the matrix K and the vector F have been computed, we use Gaussian Elimination with back substitution to determine Q. Gaussian Elimination can be performed without row interchanges and the procedure is numerically stable since the matrix K is symmetric, positive definite and tridiagonal.

Since the values $q_1,\ldots,q_N$ coincide with the values of v at the nodes j^h, j = 1,2, ... , N, these would be the approximations to the solution of our boundary value problem. The approximating function is

$$U^h = \sum_{j=1}^{N} q_j \phi_j^h$$

Analysis of The Error

Definition 2.2.1

If the functional I(v) is written in the form

$$I(v) = a(v,v) - 2(f,v) \qquad (2.2.1)$$

Then a(v ,v) is said to be the energy of the functional, and the symmetric form

$$a(v,w) = \int_0^{\pi} [pv'w' + qvw] \qquad (2.2.2)$$

is the energy inner product.

Proposition 2.2.1

Since u minimizes I(v) over H_B^1 and S^h is any closed subspace of H_B^1. Then

(a) The minimum of the functional $I(v^h)$ and the minimum $a(u - v^h, u - v^h)$ for $v^h \in S^h$, are achieved by the same function u^h; i.e.,

$$a(u - u^h, u - u^h) = \min a(u - v^h, u - v^h), \quad v^h \in S^h$$

$$= \min I(v^h), \qquad v^h \in S^h$$

$$= I(u^h). \tag{2.2.3}$$

(b) The error $u - u^h$ is orthogonal to S^h; i.e.,

$$a(u - u^h, v^h) = 0, \qquad v^h \in S^h \tag{2.2.4}$$

(c) The minimizing function uh satisfies

$$a(u^h, v^h) = (f, v^h) \qquad v^h \in S^h \tag{2.2.5}$$

In particular,

$$a(u, v) = (f, v) \qquad v \in H_B^1 \tag{2.2.6}$$

Proof :

(c) If u^h minimizes I (v) over S^h, then for any ε and $v^h \in S^h$,

$$I(u^h) \le I(u^h + \varepsilon v^h)$$

$$= a(u^h + \varepsilon v^h, u^h + \varepsilon v^h) - 2(f, u^h + \varepsilon v^h)$$

$$= I(u^h) + 2\varepsilon[a(u^h, v^h) - (f, v^h)] + \varepsilon^2 a(v^h, v^h).$$

Since ε is arbitrary, we must have

$$a(u^h, v^h) = (f, v^h)$$

In particular, a (u, v) = (f, v).

(b) Take $v = v^h$ in (2.2.6) and subtract (2.2.5) from this to obtain

$$a(u, v^h) - a(u^h, v^h) = 0.$$

Therefore, $a(u - u^h, v^h) = 0$.

(a) $a(u - u^h - v^h, u - u^h - v^h) = a(u - u^h, u - u^h) - 2\, a(u - u^h, v^h) + a(v^h, v^h)$.

since $a(v^h, v^h) \ge 0$, we have by applying (2.2.4) ,

$$a(u - u^h, u - u^h) \le a(u - u^h - v^h, u - u^h - v^h)$$

with equality if and only if v^h is identically zero. Thus u^h is the minimizing function in

(2.2.3) and (a) is proved.

The existence and uniqueness of u^h follows from the fact that S^h and H_B^1 are closed and if $a(v_N - v_M, v_N - v_M) \to 0$ as M, N $\to \infty$, then there is a $v \in S^h$ such that $a(v_N - v, v_N - v) \to 0$ as N $\to \infty$.

Corollary.

The energy in the error equals the error in the energy; i.e.,

$$a(u - u^h, u - u^h) = a(u, u) - a(u^h, u^h). \tag{2.2.7}$$

Furthermore,

$$a(u^h, u^h) \le a(u, u). \tag{2.2.8}$$

Proposition 2.2.2

If $u'' \in H^0$, then

$$[\![u - u_I]\!] \le \frac{1}{\pi^2} h^2 [\![u']\!]_0 \tag{2.2.9}$$

$$[\![u' - u_I']\!] \le \frac{1}{\pi} h\, [\![u'']\!]_0 \tag{2.2.10}$$

$$a(u - u_I, u - u_I) \le [\frac{h^2}{\pi^2} \max p + \frac{h^4}{\pi^4} \max q][\![u'']\!]^2_0 \tag{2.2.11}$$

where, $u_I(x) = \sum_1^N u(jh)\phi_j^h(x)$.

proof: See Strang and Fix [2], page 45.

Corollary.

The error $e^h = u - u^h$ in the finite element method satisfies

$$a(e^h, e^h) \le c_1 h^2 [\![u'']\!]_2^2 \le c_2 h^2 [\![f]\!]_2^0 \tag{2.2.12}$$

where the leading terms for c_1 and c_2 are max $\frac{p}{\pi^2}$ and min p respectively.

Remark 1

The above corollary gives us a bound of order $O(h^2)$ for the error in energy. In practice, thins bound is completely realistic, even for crude meshes h = $\frac{\pi}{2}$ or $\frac{\pi}{4}$.

Remark 2

In the case where f is discontinuous and u does not have a second derivative at some point, the error in energy is not usually $0(h^2)$. In most cases it can be shown to be of O (h).

Proposition 2.2.3

For any solution u $\in H_B^1$, the finite element method converges in the energy norm; i.e.,

$$H_B^1 a(e^h, e^h) \to 0 \quad \text{as} \quad h \to 0 .$$

Proof :

Since H_B^1 was constructed by completing H_B^2, there is a sequence v_n in H_B^2 converging to u in the energy norm. For each fixed N, the finite element approximations v_N^h converge to v_N as $h \to 0$, by the above proposition and corollary. Therefore, choosing N large enough and then h small, we can find a function v_N^h in S^h, which is arbitrarily close to u. Since the projection u^h will be even closer, the sequence u^h must converge to u.

Proposition 2.2.4

The piecewise linear approximation u^h, derived by the Ritz method, satisfies

$$[\![u - u^h]\!]_0 \le \rho C^2 h^2 [\![u'']\!]_0 \le \rho^2 C^2 h^2 [\![f]\!]_0 \qquad (2.2.13)$$

where ρ and C are constants.

Proof: See Strang and Fix [2], pages 48,49.

Proposition 2.2.5

If f is replaced by its linear interpolate $f_I = \sum_{j=1}^{N} f(jh)\phi_j^h$, then the induced error

$u^h - u^{\sim h}$ satisfies $$a(u^h - u^{\sim h}, u^h - u^{\sim h}) \le \frac{K\rho^2 h^4}{\pi^4} [\![f'']\!]_0^2 \qquad (2.2.14)$$

where ρ and K are constants.

Proof :

The exact solution $u - \tilde{u}$, corresponding to the data $f - f_I$ is bounded by

$$[\![u - u_I]\!]_2 \le \rho [\![f - f_I]\!]_0 \le \frac{\rho^2 h^2}{\pi 2} [\![f'']\!]_0 \qquad (2.2.15)$$

Therefore, it follows that

$$\begin{aligned} a(u^h - u^{\sim h}, u^h - u^{\sim h}) &\le K [\![u^h - u^{\sim h}]\!]_1^2 \\ &\le K [\![u^h - u^{\sim h}]\!]_2^2 \\ &\le \frac{K\rho^2 h^4}{\pi^4} [\![f'']\!]_0^2 . \end{aligned}$$

Corollary

If f is replaced by its average value f (j), over each interval, then the induced error with respect to the energy norm is O (h^4).

Conclusion

From the last corollary (2.2.12) and (2.2.13.) we can conclude that (2.1.16.) will yield a process with error $[\![u^h - u^{\sim h}]\!]_0$ of order O(h^2).

The Algorithm

To approximate the solution to the boundary value problem

$$-(pu')' + q_0 u = f \qquad p \geq 0 \text{ min } p > 0,\ q \geq 0$$

$$u(0) = 0, u(\pi) = 0 \ ,$$

where p has a piecewise continuous first derivative on [0 , π], f is piecewise continuous on [0 , π] and has finite energy, select an integer N > 0.

Step – 1. Set h = $\frac{\pi}{N}$ and x_j = jh for j = 0,1, ... , N .

Step – 2. Compute the average values p(1), p(2),q(1), q(2),f(1) and f(2) on the subintervals [x_0 , x_1] and [x_1 , x_2]. Set

$$k_{11} = \frac{[p(1) + p(2)]}{h} + \frac{h[q(1) + q(2)]}{3},$$

$$k_{12} = \frac{hq(2)}{6} - \frac{p(2)}{h},$$

$k_{1j} = 0$, for j = 3,4, ..., N,

$$f = \frac{h[f(1) + f(2)}{2}.$$

Step – 3 For each i = 2,3, ... , N-1 compute p(i + 1), q (i + 1) , f (i + 1) and set

$k_{ij} = 0$, for j = 1, 2, ... , i -2 and j = i+2, ..., N,

$$k_{i,i-1} = k_{i-1,i},$$

$$k_{ii} = \frac{[p(i) + p(i+1)]}{h} + \frac{h[q(i) + q(i+1)]}{3},$$

$$k_{i,i+1} = \frac{hq(i+1)}{6} - \frac{p(i+1)}{h},$$

$$f_i = \frac{h[f(i) + f(i+1)]}{2}.$$

Step – 4 Set $k_{N,j} = 0$, for j = 1,2, ... , N-2,

$k_{N,N-1} = k_{N-1,N}$,

$$k_{N,N} = \frac{p(N)}{h} - \frac{hq(N)}{3},$$

$$f_N = \frac{hf(N)}{2}.$$

Step – 5 The matrix K = (k_{ij}) and the vector F (f_1, ..., f_N) are known. Solve the system KQ = F by Gaussian Elimination with back substitution to obtain the vector Q = (q_1, ... , q_N) ^{T}.

Step – 6 The procedure is complete, and the approximate solution is

$$\phi(x) = \sum_{j=1}^{N} q_j \phi_j^h(x).$$

Inhomogeneous Boundary Conditions

In this section we retain (2.1.2.) with the same restrictions on p, q and f, but change (2.1.3.) to the inhomogeneous conditions.

$$u(0) = g, \quad u'(\pi) + \alpha u(\pi) = b \qquad \alpha \geq 0. \tag{2.4.1.}$$

In our new admissible space H_B^1 , all functions satisfy $u(0) = g$, Therefore the difference between any admissible $v_0 = v_1 - v_2 = 0$. Let v_0 denote the space of these functions v_0.

Since the boundary condition at x = π is of a new kind, the functional I(v) is redefined as

$$\mathrm{I(v)} = \int_0^{\pi} [p(v')^2 + qv^2] + \alpha p(\pi) v^2(\pi) - 2\int_0^{\pi} fv - 2bp(\pi)v(\pi). \tag{2.4.2}$$

The new condition has introduced a boundary condition both in the linear term and the energy of the functional

$$a(v,v) = \int_0^{\pi} [p(v')^2 + qv^2] + \alpha p(\pi)v^2(\pi). \qquad (2.4.3)$$

Proposition 2.4.1

The solution u to (2.1.2.) and (2.4.1.) uniquely minimizes I(v) over the new admissible space H_B^1.

Proof: Let ε and $v_0 \in v_0$ be given . Then

$$I(u+v_0\varepsilon) = I(u) + 2\varepsilon[\int_0^{\pi} (pu'v'_0 + quv_0 - fv_0) + \alpha p(\pi)u(\pi)v_0(\pi) - bp(\pi)v_0(\pi)]$$

$$+ \varepsilon^2[\int_0^{\pi} (p(v_0')^2 + qv_0^2 + \alpha p(\pi)v_0(\pi)^2]$$

$$= I(u) + 2\varepsilon[\int_0^{\pi} (-(pu')' + qu - f)v_0 + p(\pi)(u'(\pi) + \alpha u(\pi) - b)v_0(\pi)]$$

$$+\varepsilon^2[\int_0^{\pi} (p(v_0')^2 + 2v_0^2) + \alpha p(\pi)v_0(\pi)^2]$$

$$= I(u) + \varepsilon^2[\int_0^{\pi} (p(v_0')^2 + qv_0^2) + \alpha p(\pi)v_0(\pi)^2].$$

Therefore I(u) $\leq I(u+v_0\varepsilon)$ with equality if and only if v_0 is identically zero.

Since H_B^1 is not a vector space in this case, we simply ask for S^h to have the same form. The trial functions v^h need not lie in H_B^1, but the difference of any tow trial functions must be in the space v_0. These differences $v_1^h - v_2^h = v_0^h$ form a finite dimensional space S_0^h, which is a subspace of v_0.

For linear elements, the trial space S^h will be all piecewise linear functions witch satisfy $v^h(0) = g$. S_0^h is the same piecewise linear trial space introduced in Section 1 of the chapter, and every v^h in S^h has the form

$$v^h(x) = g\phi_0^h(x) + \sum_{j=1}^{N} q_j \phi_j^h(x). \qquad (2.4.4)$$

$I(v^h)$ is a quadratic in the unknown $q_1, \ldots, q_N$, and it minimization leads again to a linear tridiagonal system KQ = F . In the interior of the interval i.e., for all but the first and the last rows of the matrix this system will be identical to the one in Sections 1 and 2 of this chapter. The first equation of the system (with coefficients p = q = 1) is

$$\frac{(-g + 2q_1 - q_2)}{h} + h\frac{(g + rq_1 + q_2)}{b} = \int_0^h f\phi_1^h .$$

Shifting the terms involving g to the right hand side, the first row of K is exactly as before, whereas the right side has changed by the term $\frac{g}{h} - \frac{h}{b}$.

At the other end, the new terms in $I(v^h)$ are

$$\alpha p(\pi)(v^h(\pi))^2 - 2bp(\pi)v^h(\pi) = \alpha p(\pi)q_N^2 - 2bp(\pi)q_N .$$

Therefore in the last equation, after deleting the factor 2, there is an extra $bp(\pi)$ in the right hand, and an extra $\alpha p(\pi)$ in the entry K_{NN}.

The error estimates for this problem will be the same as in Section 2, i.e.,

$a(u - u^h, u - u^h) = 0(h^2)$ and $[\![u - u^h]\!]_0 = 0(h^2)$

CHAPTER 3

A PROBLEM IN TWO DIMENSION

The Problem

We consider the second-order partial differential equation,

$$\mathrm{Lu} = f, \tag{3.1.1}$$

where

$$Lu \equiv -\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} + cu \tag{3.1.2}$$

This equation is to hold on a bounded region Ω, whose boundary $\partial\Omega$ is piecewise smooth. We assume in (3.1.1) and (3.1.2) that on $\bar{\Omega} \equiv \Omega \cup \partial\Omega$, the functions c, $f \in C(\bar{\Omega}), a_{ij} \in C^1(\bar{\Omega})$, and c(x, y) ≥ 0. Furthermore, we suppose that for each $(x_1, x_2) \in C$ and for any real numbers t_1, t_2 there exists a positive constant μ such that

$$\sum_{i,j=1}^{2} a_{ij}(x_1, x_2) t_i t_j \geq \mu \sum_{i=1}^{2} t_1^2 \tag{3.1.3}$$

this is the condition that L is uniformly elliptic in Ω.

Throughout this chapter we identify $x_1 \equiv x$ and $x_2 \equiv y$ so that (3.1.2) may also be written as

$$Lu \equiv -\sum_{i=1}^{2} \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{2} a_{ij} f \frac{\partial u}{\partial x_j}\right) + cu. \tag{3.1.4}$$

Another commonly used notation is $Lu \equiv -\nabla \bullet A\nabla u + cu$.

To develop a discretization of (3.1.1) in which the derivatives are approximated by difference quotients, we assume that the problem domain Ω is the interior of a polygon which has been partitioned in to rectangular cells by a mesh or gird $\prod_{\Omega}$. This is a rectangular network of lines that are parallel to the coordinate axes and intersect the boundary of Ω only at points where they themselves intersect.

We can regard Ω as being embedded in a rectangle Ω^* in such a way that the line of $\prod_{\Omega}$

$$\{x = x_i\}_{i=0}^{M_1} \cup \{y = y\}_{j=0}^{M_2}$$

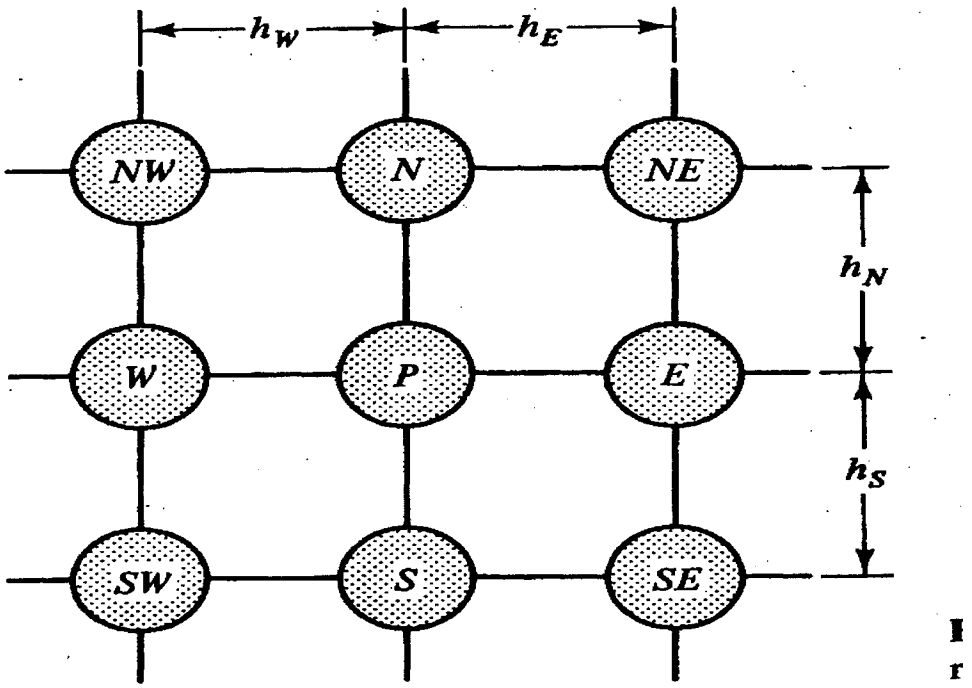
partition Ω^* as well as Ω. This permits an easy reference frame for the mesh which partitions that nonrectangular region Ω .The points of intersection of that members of $\prod_{\Omega}$ are called mash points, and the mesh gauge h is defined as the maximum of the set. $\{|x_i - x_{i-1}|, |y_j - y_{j-1}|\}$ .The mesh points which belong respectively to Ω and $\partial\Omega$ form two sets denoted by Ω_h and $\partial\Omega_h$. Also, $\tilde{\Omega}_h \equiv \Omega_h \cup \partial\Omega_h$.

Our problem is to construct a mesh function $U : \tilde{\Omega}_h \to R$ such that U is a "good" approximation to u on $\tilde{\Omega}_h$. This mesh function is determined by solving a system of finite difference equations. These equations are obtained from (3.1.1) by approximating the differential equation at each point by an algebraic equation in which the derivatives have been replace by appropriate difference approximations.

Obviously, the boundary conditions are going to affect the equations that are associated with boundary points. We defer consideration of this issue for the present, and first discuss the mechanics of approximating the derivatives. We adopt the standard

compass designations of neighbors and denote a typical mesh point with coordinates(x, y) by P and its neighbors by compass abbreviations (Figure 3.1).

Figure 3.1



The following theorem presents the desired approximations to the derivatives in (3.1.2) as well as an asymptotic $(h \to 0)$ behavior of the associated truncation error.

THEOREM: Suppose that $u \in C^4(\tilde{\Omega})$ and $P \in \Omega$. Then as $h \to 0$,

$$u_x(p) = \frac{h_w}{h_E(h_E + h_w)}u(E) + \frac{(h_E - h_w)}{h_E h_w}u(p) - \frac{h_E}{h_E(h_E + h_w)}u(w) + O(h^2) \qquad (3.1.5)$$

$$u_y(p) = \frac{h_s}{h_N(h_N + h_s)}u(N) + \frac{(h_N - h_s)}{h_N h_S}u(p) - \frac{h_N}{h_s(h_N + h_s)}u(s) + O(h^2) \qquad (3.1.6)$$

$$u_{xx}(p)=\frac{2}{h_E(h_E+h_w)}u(E)-\frac{2}{h_w h_E}u(p)+\frac{2}{h_w(h_E+h_w)}u(w)+O((h_w-h_E)+h^2) \tag{3.1.7}$$

$$u_{yy}(p)=\frac{2}{h_N(h_N+h_s)}u(N)+\frac{2}{h_N h_S}u(p)-\frac{2}{h_s(h_N+h_s)}u(s)+O((h_s-h_N)+h^2) \tag{3.1.8}$$

$$u_{xy}(p)=\begin{cases} -\dfrac{h_E}{h_w(h_E+h_w)}\left[-\dfrac{h_N}{h_s(h_N+h_s)}u(sw)+\dfrac{(h_N-h_s)}{h_N h_S}u(w)+\dfrac{h_s}{h_N(h_N+h_s)}u(Nw)\right] \\ +\dfrac{(h_E-h_w)}{h_E h_w}\left[-\dfrac{h_N}{h_s(h_N+h_s)}u(s)+\dfrac{(h_N-h_s)}{h_N h_S}u(p)+\dfrac{h_s}{h_N(h_N+h_s)}u(N)\right] \\ +\dfrac{h_w}{h_E(h_E+h_w)}\left[-\dfrac{h_N}{h_s(h_N+h_s)}u(sE)+\dfrac{(h_N-h_s)}{h_N h_S}u(E)+\dfrac{h_s}{h_N(h_N+h_s)}u(NE)\right] \\ +O((h_E-h_w)+(h_N-h_s)+h^2) \end{cases} \tag{3.1.9}$$

for proof of this Theorem see Forsythe and Wasow [1960].

Now let $M(\Omega_h)$ and $M(\tilde{\Omega}_h)$ denote the sets of grid functions whose domains are, respectively, Ω_h and $\tilde{\Omega}_h$. When the order terms in Theorem are ignored, the right-hand sides of (3.1.5) – (3.1.9) define difference operator $u_x, u_y, u_{xx}, u_{yy}, u_{xy}$ from $M(\Omega_h)$ to $M(\tilde{\Omega}_h)$. These correspond to the partial derivatives $u_x, u_y, u_{xx}, u_{yy}, u_{xy}$. With these definitions, we define the finite difference operator $L_h : M(\tilde{\Omega}_h) \to M(\Omega_h)$ by the condition that for $p \in \Omega_h$,

$$(L_h U)P \equiv -\left[a_{11}U_{xx}+(a_{12}+a_{21})U_{xy}+a_{22}U_{yy}+(a_{11}^{(1,0)}+a_{21}^{(0,1)})U_x+(a_{22}^{(0,1)}+a_{12}^{(1,0)})U_y\right] \tag{3.1.10}$$

where $$a_{ij}^{(p,q)}=\frac{\partial^{p+q}a_{ij}}{\partial x^p \partial y^q}$$

and the right-hand side is evaluated at P, we note that (3.1.10) may also be written in the form

$$(L_h U)P = \sum_Q A(P,Q)U(Q) \tag{3.1.11}$$

where the only possible nonzero couplings A(P ,Q) are given by the formulas

$$\text{A (P, P)} = \begin{cases} a_{11}\dfrac{2}{h_E h_w} - (a_{12}+a_{21}) - \dfrac{(h_E - h_w)(h_N - h_s)}{h_E h_w h_N h_s} + a_{22}\dfrac{2}{h_N h_s} + c \\ -(a_{11}^{(1,0)} + a)\dfrac{(h_E - h_w)}{h_E h_w} - (a_{22}^{(0,1)} + a_{12}^{(1,0)})\dfrac{(h_N - h_s)}{h_N h_S} \end{cases}$$

$$\text{A (P, E)} = -a_{11}\frac{2}{h_E(h_E + h_w)} - (a_{12}+a_{21})\frac{(h_N - h_s)}{h_N h_S h_E(h_E + h_w)} - (a_{11}^{(1,0)} + a_{21}^{(0,1)})\frac{h_w}{h_E(h_E + h_w)}$$

$$\text{A (P, W)} = -a_{11}\frac{2}{h_E(h_E + h_w)} + (a_{12}+a_{21})\frac{h_E(h_s - h_N)}{h_N h_S h_E(h_E + h_w)} + (a_{11}^{(1,0)} + a_{21}^{(0,1)})\frac{h_E}{h_W(h_E + h_w)}$$

$$\text{A (P, N)} = -a_{22}\frac{2}{h_s(h_s + h_N)} - (a_{12}+a_{21})\frac{h_s(h_E - h_W)}{h_w h_S h_E(h_N + h_s)} - (a_{22}^{(0,1)} + a_{12}^{(1,0)})\frac{h_s}{h_W(h_E + h_w)}$$

$$\text{A (P, S)} = -a_{22}\frac{2}{h_s(h_s + h_N)} + (a_{12}+a_{21})\frac{h_N(h_E - h_W)}{h_w h_S h_E(h_N + h_s)} + (a_{22}^{(0,1)} + a_{12}^{(1,0)})\frac{h_N}{h_s(h_s + h_N)}$$

$$\text{A (P, NW)} = (a_{12}+a_{21})\frac{h_E h_s}{h_w h_N(h_s + h_N)(h_w + h_E)}$$

$$\text{A (P, SW)} = -(a_{12}+a_{21})\frac{h_E h_N}{h_w h_s(h_s + h_N)(h_w + h_E)}$$

$$\text{A (P, SW)} = -(a_{12}+a_{21})\frac{h_w h_s}{h_N h_E(h_s + h_N)(h_w + h_E)}$$

$$\text{A (P, SE)} = (a_{12}+a_{21})\frac{h_w h_N}{h_s h_E(h_s + h_N)(h_w + h_E)}$$

In these formulas, the coefficient functions a_{ij} as well as their derivatives are evaluated at the point P. The derivatives $a_{11}^{(1,0)}, a_{12}^{(1,0)}, a_{21}^{(0,1)}$ and $a_{22}^{(0,1)}$ can also be approximated using the appropriate formulas from Theorem, Thus yielding couplings which depend only on functional values $a_{ij}(p)$ and $c(p)$.

Boundary Conditions

The finite difference equation

$$(L_h U)P = f(P), \qquad p \in \Omega_h \tag{3.2.1}$$

is equivalent to a rectangular linear system of equations which may be written in matrix form as

$$\tilde{A}\tilde{U} = K \tag{3.2.2}$$

The equation associated with the mesh point P "approximates" the partial differential equation (3.1.1) at the point P. Dirichlet boundary conditions are easily accommodated. For example, if $p \in \partial\Omega_h \cap \partial\Omega_1$, and we simply set

$$U(P) = g_1(P) \tag{3.2.3}$$

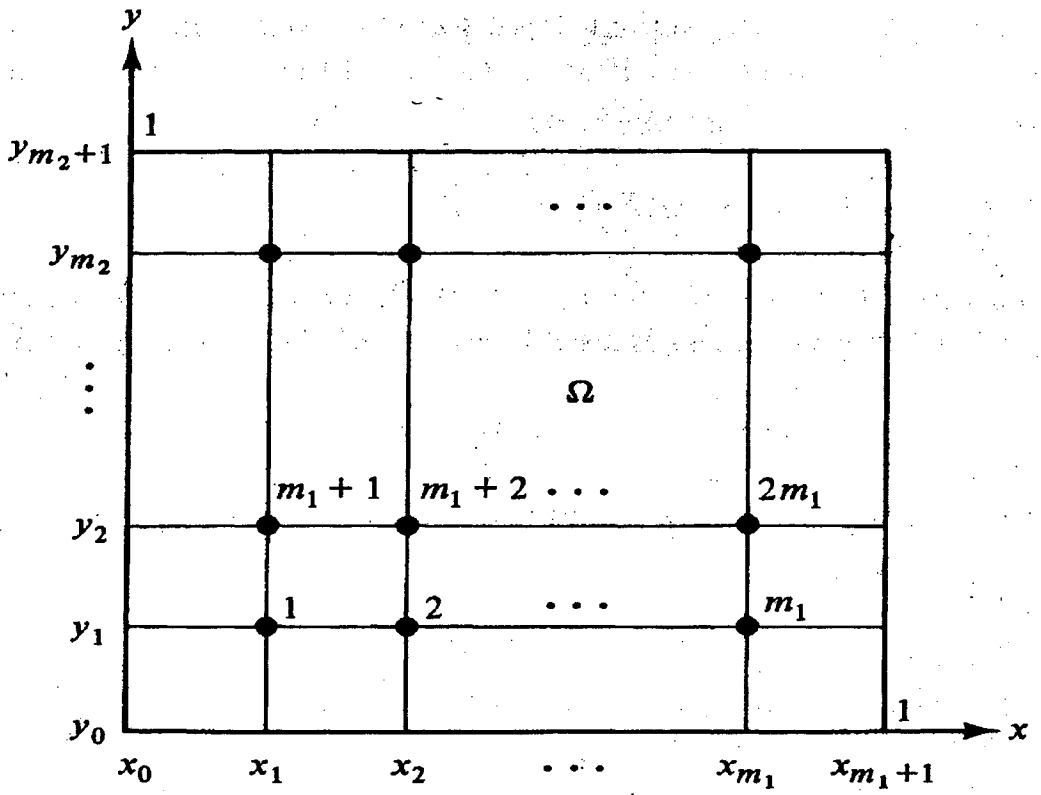
obviously (3.2.3) can be used to eliminate the unknowns in (3.2.2) that corresponds to mash points on $\partial\Omega$.

The M×M system of linear equations that results from (3.2.2) after (3.2.3) has been used to eliminate the boundary unknowns may write as

$$AU = K \tag{3.2.4}$$

if $\tilde{\Omega}$ is the rectangle of Figure 3.2 and the unknowns are given the "lexicographical" ordering shown in this figure, then $M = m_1 m_2$ and the coefficient matrix A may be partitioned as a block-tridiagonal matrix.

Figure 3.2



$$A=\begin{bmatrix} A_{11} & A_{12} & & & \\ A_{21} & A_{22} & A_{23} & & \\ .. & .... & ..... & .... & \\ & & ... & .... & . \\ & & & A_{m_2 m_{2-1}} & A_{m_2 m_2} \end{bmatrix} \qquad (3.2.5)$$

where each nonzero block is itself an $m_1 \times m_1$ tri-diagonal matrix. In fact, with this partitioning we have $U = (U_1, U_2, ..., U_{m_2})^T$, where $U_j = (U_{1j}, U_{2j}, ..., U_{m_1,j})^T$ and $U_{ij} = U(P_{ij})$, $P_{ij} = (x_i, y_j)$ Furthermore, if we denote the element in row i and column k of matrix A_{pq} by $[A_{pq}]_{ik}$ and let $J = \{(i,k) \mid i = 1, ..., m_1; k = i-1, i, i+1; k \neq 0, m_1 + 1\}$, then for j = 1,2,..., m_2,

$$\left[A_{jj}\right]_{ik} = \begin{cases} A(P_{ij}, P_{kj}), \\ 0 \end{cases}$$

$$\left[A_{jj+1}\right]_{ik} = \begin{cases} A(P_{ij}, P_{kj+1}), \\ 0 \end{cases}$$

$$\left[A_{jj-1}\right]_{ik} = \begin{cases} A(P_{ij}, P_{kj-1}), \\ 0 \end{cases}$$

The source term K in (3.2.4) is given by $K = (K_1, K_2, ..., U_{m_2})^T$, where for $2 \le j \le m_2 - 1$

$$[K_j]_i = \begin{cases} f_{1j} - \sum_{k=j-1}^{j+1} A(P_{1j}, P_{0k}) U_{0k}, i = 1 \\ f_{ij}, 2 \le i \le m_1 - 1 \\ f_{m1,j} - \sum_{k=j-1}^{j+1} A(P_{m1,j}, P_{m1+1,k}) U_{m1+1,k}, i = m_1 \end{cases}$$

while

$$[K_1]_i = \begin{cases} f_{11} - \sum_{k=j-1}^{j+1} A(P_{11}, P_{0k}) U_{0k} - \sum_{k=1}^{2} A(P_{11}, P_{k0}) U_{k0}, i = 1 \\ f_{ij} - \sum_{k=0}^{2} A(P_{11}, P_{k0}) U_{k0}, 2 \le i \le m_1 - 1 \\ f_{m1,j} - \sum_{k=j-1}^{j+1} A(P_{m1,j}, P_{m1+1,k}) U_{m1+1,k} - \sum_{k=m_1-1}^{m_1} A(P_{m_1,1}, P_{k0}) U_{k0}, i = m_1 \end{cases}$$

and K_{m_2} is defined analogously,

Note that if $a_{12} = a_{21} = 0$, the off-diagonal blocks in (3.2.5) reduce to diagonal matrices. Moreover, when the mesh gauge h is sufficiently small, it can be shown that A is nonsingular, and consequently system (3.2.4) has a unique solution.

The Variational Formulation

In this section the variational formulation of elliptical boundary value problems used to generate finite difference equations. The mesh lines of $\prod_{\Omega}$ partition Ω into a union of rectangles, and it facilitates the discussion if an auxiliary or dual system $\prod_{\Omega}^{'}$ of horizontal and vertical mesh lines is constructed in such a way that the lines of $\prod_{\Omega}^{'}$ fall halfway between the lines of $\prod_{\Omega}$ A sub region, R bounded by mesh lines of $\prod_{\Omega}^{'}$ and possibly $\partial\Omega$ is called a cell. Each cell is either a rectangle and contains exactly one mesh point.

For the second-order elliptic problem with Dirichlet boundary conditions, the variational formulation asks for the minimization over

$S = \{w \in H^1(\Omega) \mid w(P) = g_1(p), P \in \partial\Omega_1\}$ of the functional

$$I[w] = \frac{1}{2}\sum_{R}\iint_{R}\left(\sum_{i,j=1}^{2} a_{ij}\frac{\partial w}{\partial x_i}\frac{\partial w}{\partial x_j} + cw^2 - 2fw\right)dxdy \qquad (3.3.1)$$

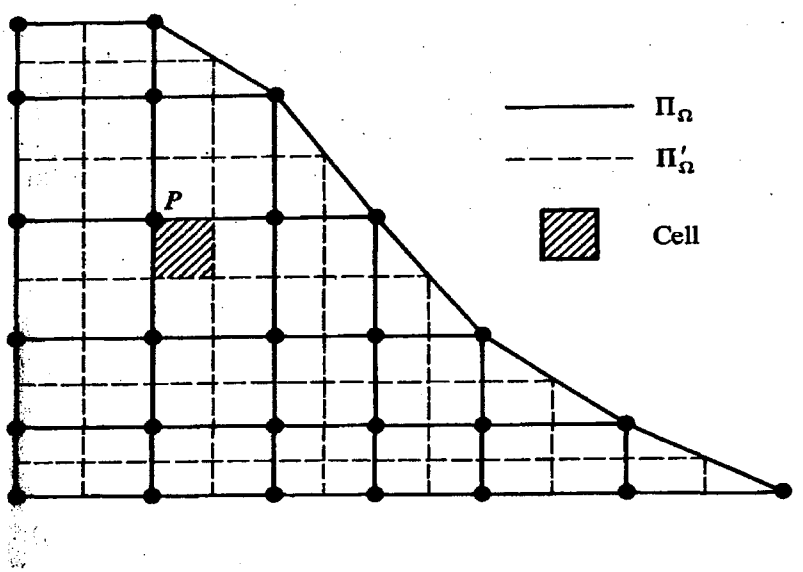
where the sum is over all cells R determined by $\prod_{\Omega}^{'}$ and $\partial\Omega$.

Let u and its derivatives be approximated by constants over each cell; for example in the shaded cell of Figure 3.3.

$$u \approx U(P), \qquad u_x \approx \frac{U(E)-U(P)}{h_E}, \qquad u_y \approx \frac{U(S)-U(P)}{h_s} \quad \text{and so on.}$$

Systematic substitution of such approximations in to $I[u]$ gives rise to a function J whose arguments are the grid function values $\{U(P) \mid P \in \Omega_k\}$. The variation problem is then

Figure 3.3



replaced by the finite dimensional minimization problem: Find U* $\in M(\tilde{\Omega}_h)$ such that

$$J(U^*) = \min_{S_h} J(U) \tag{3.3.2}$$

where, $S_h = \{U \in M(\tilde{\Omega}_h) \mid U(P) = g_1(P), P \in \partial\Omega_h \cap \partial\Omega_1\}$.

Since this is a conventional minimization problem for a function of several variables, a minimum can only be attained at a point where that gradient of J is zero. Hence (3.3.2) leads to the system of equations

$$\frac{\partial J}{\partial U(P)} = 0, \qquad P \in \tilde{\Omega}_h - \partial\Omega_1 \tag{3.3.3}$$

formally, this is the finite difference system generated by the variational formulation of the boundary value problem.

The condition (3.3.3) is equivalently written as

$$(L_h U)P = \sum_Q A(P,Q)U(Q) = S(P), \qquad P \in \tilde{\Omega}_h - \partial\Omega_1$$

where the only possible nonzero couplings are

$$\mathrm{A(P,P)} = \begin{cases} [a_{11}(P)+a_{11}(W)]\dfrac{h_s+h_N}{4h_w} + [a_{11}(P)+a_{11}(E)]\dfrac{h_s+h_N}{4h_E} \\ +[a_{22}(P)+a_{22}(S)]\dfrac{h_E+h_w}{4h_s} + [a_{22}(P)+a_{22}(N)]\dfrac{h_E+h_w}{4h_N} \\ +c(P)\dfrac{(h_E+h_w)(h_s+h_N)}{4} \end{cases}$$

$$A(P,E) = -[a_{11}(P)+a_{11}(E)]\frac{h_s+h_N}{4h_E}$$

$$A(P,W) = -[a_{11}(P)+a_{11}(W)]\frac{h_s+h_N}{4h_W}$$

$$A(P,S) = -[a_{22}(P)+a_{22}(S)]\frac{h_W+h_E}{4h_S}$$

$$A(P,N) = -[a_{22}(P)+a_{22}(N)]\frac{h_W+h_E}{4h_N}$$

$$A(P,SW) = -\frac{1}{4}[a_{12}(W)+a_{12}(S)]$$

$$A(P,NE) = -\frac{1}{4}[a_{12}(N)+a_{12}(E)]$$

$$A(P,SE) = \frac{1}{4}[a_{12}(E)+a_{12}(S)]$$

$$A(P,NW)=\frac{1}{4}[a_{12}(N)+a_{12}(W)] \qquad \text{and}$$

$$S(P)=f(P)\frac{(h_W+h_E)(h_N+h_S)}{4}$$

The couplings above were obtained by approximating each function u, a_{kj},c and f in a cell of $\prod_{\Omega}'$ by its value at the mesh point $\prod_{\Omega}$ which determines the cell. For example, if R is the shaded cell in Figure 3.3.

$$\int_R a_{11}(x,y)dxdy \approx a_{11}(P)\frac{h_E h_S}{4}$$

other quadratures could just as easily have been used.

Conclusion

In the One-Dimension case, using piecewise linear elements in the finite element method we obtain approximation with error of order $O(h^2)$, and we have confirmed that these error are indeed of order $O(h^2)$.

It also has confirmed in The Two-Dimension case, using finite (forward) difference method the error is $O(h)$. This can be improved by using finite (centered) difference and better approximation for the function on the cells.

APPENDIX I

Results

Table I – Example 2.1

$u(t) = t * e^t$, $f(t) = (t-1)e^t, r(t) = t, p(t) = t + 4$

t	$u(t)$	$C(t)$	E_h	$E_{h/2}$
1.0625	3.074445	3.243464	-0.16902	-0.04226
1.1250	3.465243	3.76861	-0.30337	-0.07584
1.1875	3.893661	4.303348	-0.40969	-0.10242
1.2500	4.362927	4.855537	-0.49261	-0.12315
1.3125	4.876527	5.431793	-0.55527	-0.13882
1.3750	5.438228	6.037861	-0.59963	-0.14991
1.4375	6.052098	6.678893	-0.62679	-0.1567
1.5000	6.722531	7.359647	-0.63712	-0.15928
1.5625	7.454267	8.08465	-0.63038	-0.1576
1.6250	8.252427	8.858318	-0.60589	-0.15147
1.6875	9.122534	9.68506	-0.56253	-0.14063
1.7500	10.07055	10.56936	-0.49881	-0.1247
1.8125	11.1029	11.51584	-0.41294	-0.10323
1.8750	12.22653	12.52934	-0.30281	-0.0757
1.9375	13.44891	13.61496	-0.16605	-0.04151

Table II

Example 2.2

$$u(t) = t - t^2, f(t) = 40t^2, p(t) = 2, r(t) = t$$

t	$u(t)$	$C(t)$	E_h	$E_{h/2}$
1.062500	-0.066406	-0.091279	0.024873	0.006218
1.125000	-0.140625	-0.257248	0.116623	0.029156
1.187500	-0.222656	-0.405302	0.182646	0.045662
1.250000	-0.312500	-0.540647	0.228147	0.057037
1.312500	-0.410156	-0.566999	0.156843	0.039211
1.375000	-0.515625	-0.787041	0.271416	0.067854
1.437500	-0.628906	-0.902728	0.273822	0.068456
1.500000	-0.750000	-1.015501	0.265501	0.066375
1.562500	-0.878906	-1.126430	0.247524	0.061881
1.625000	-1.015625	-1.236316	0.220691	0.055173
1.687500	-1.160156	-1.345770	0.185614	0.046404
1.750000	-1.312500	-1.455260	0.142760	0.035690
1.812500	-1.472656	-1.665156	0.192500	0.048125
1.875000	-1.640625	-1.775753	0.135128	0.033782
1.937500	-1.816406	-1.887298	0.070892	0.017723

Table III

Example 3.1

$$u(x, y) = 10(x^2 - y^2)$$

x	y	$u(x,y)$	$C(x,y)$	E_h	$E_{h/2}$
1.10000	1.05000	1.07500	1.06000	-0.01500	-0.00750
1.10000	1.10000	0.00000	0.01017	0.01017	0.00508
1.10000	1.15000	-1.12500	-1.11323	0.01176	0.00588
1.20000	1.05000	3.37500	3.38444	0.00943	0.00472
1.20000	1.10000	2.30000	2.30923	0.00923	0.00462
1.20000	1.15000	1.17500	1.18338	0.00838	0.00419

Table IV

Example 3.2

$$u(x, y) = e^{x} \cos y$$

x	y	$u(x,y)$	$C(x,y)$	E_h	$E_{h/2}$
1.0500	1.1000	1.4948	1.5093	-0.0145	-0.0073
1.1000	1.1000	1.3627	1.3525	0.0102	0.0051
1.1500	1.1000	1.2272	1.2164	0.0108	0.0054
1.0500	1.2000	1.6520	1.6426	0.0094	0.0047
1.1000	1.2000	1.5060	1.5145	-0.0085	-0.0043
1.1500	1.2000	1.3562	1.3479	0.0084	0.0042

Table V

Example 3.3

$$u(x, y) = e^x \cos y + e^y \cos x$$

x	y	$u(x,y)$	$C(x,y)$	E_h	$E_{h/2}$
1.10000	1.05000	2.79101	2.80226	0.01125	0.00563
1.10000	1.10000	2.72536	2.73462	0.00926	0.00463
1.10000	1.15000	2.65971	2.66831	0.00860	0.00430
1.20000	1.05000	2.68749	2.70035	0.01286	0.00643
1.20000	1.10000	2.59458	2.69333	0.09875	0.04938
1.20000	1.15000	2.50006	2.59570	0.09564	0.04782

Table VI

Example 3.4

$$u(x, y) = e^x \sin y + e^y \sin x$$

x	y	$u(x,y)$	$C(x,y)$	E_h	$E_{h/2}$
1.10000	1.05000	5.15264	5.15724	0.00460	0.00230
1.10000	1.10000	5.35467	5.35888	0.00421	0.00211
1.10000	1.15000	5.55670	5.55899	0.00229	0.00115
1.20000	1.05000	5.54339	5.57249	0.02910	0.01455
1.20000	1.10000	5.75891	5.77339	0.01448	0.00724
1.20000	1.15000	5.97404	5.98007	0.00603	0.00301

APPENDIX II

C++ PROGRAMS

Main Program

```
// The following program solves the boundary value problem (BVP)
//                                   -(pu')'+ qu = f
// with boundary conditions u(0) = 0, u( π ) = 0 using finite element method
// The above BVP is change to become a problem of solving
//The linear system of equations  Lu = f
// Function t*e^t


#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>

int order;
Void read (double a[20][20],double b[20])

{  double r1,r2,r3,h,fb,fa,a0,ri,e;
        clrscr ();
        h=0.0625;
        a0=1;
        e=2.718281;
        fa=2.718281;
        fb=14.778103;

 /* cal value of aj */

for(int i=1;i<16;i++)
   { for(int j=1;j<16;j++)
        {  a[i][j]=0;
           r1= a0+(j-1)*h;
           r2= a0+(h*j);
           r3= a0+(j+1)*h;
           if (i==j)
           {    a[i][j-1]= (r1*(-1))+((r1+4)*(h*h/6));
```

```
                a[i][j] = ((r2*2))+((r2+4)*(2*h*h/3));
                a[i][j+1]=(r3*(-1))+((r3+4)*(h*h/6));
                j++;
            }
        }
        ri=a0+(h*i);
        b[i] = (ri-1)*pow(e,ri)*h*h;
        cout<<"\n value of j is : "<<ri<<"\t And value of bi is "<<b[i];
                if(i==1)
                b[i]=b[i]-(a[i][0]*fa);
                if(i==15)
                b[i]=b[i]-(a[i][16]*fb);
    }
}

void Gauss_s(double a[20][20],double b[20],double x[20],int n)
   {
      for(int k=0;k<n;k++)
        {  for(int i=1;i<=15;i++)
              {  x[i]=b[i];
                 for(int j=i-1;j<=i+1;j++)
                  { if (j!=i)
                     x[i] = x[i]-(a[i][j]*x[j]);
                   }
              }
        }
    }


void main()
  {
      double a[20][20],b[20],x[20],ej[20],f[20];
      double r,h,e;
      int n;
      FILE *s1;
      s1= fopen ("fun_tet.txt","w");
      clrscr();
      read(a,b);

      //printing matrix

      fprintf(s1,"\n Given matrix is : \n");

for(int i=1;i<16;i++)
      { for(int j=1;j<16;j++)
                {    fprintf(s1,"\t %f ",a[i][j]);
```

```
                }
            fprintf(s1,"\t\t %f \n",b[i]);
        }
        cout<<"\n Enter total no. of ittration :";
        cin>>n;
                                /* Normalize Matrix  */

fprintf(s1,"\n Given matrix is : \n");
      for( i=1;i<=15;i++)
        {  for(int j=1;j<=15;j++)
             { if (j!=i)
                    a[i][j] = a[i][j]/a[i][i];
                    cout<<"\t "<<a[i][j];
                    fprintf(s1,"\t %f ",a[i][j]);
             }
             b[i]=b[i]/a[i][i];
             cout<<"\t"<<b[i];
             fprintf(s1,"\t\t %f \n",b[i]);
             cout<<"\n";
        }

        Gauss_s(a,b,x,n);
        cout<<"\n Approximated Soln. of the given system of Equation by using
                Gauss-Shield is : \n";

        fprintf (s1,"\n Approximated Soln. of the given system of Equation by using
                  Gauss-Shield method is : ");

    for(i=1;i<=15;i++)
      {  cout<<"\n x["<<i<<"]"<<x[i];
                  fprintf(s1, "\n");
                  fprintf(s1, " %f  ",x[i]);
       }

/* calculating error.*/

       fprintf(s1,"\n Value of X   F(x)      C(x)      Error.");
       h=0.0625;
       e=2.718281;
       for( i=1;i<=15;i++)
            { r= 1+(h*i);
             f[i]= r*pow(e,r);
             ej[i]=f[i]-x[i];
             cout<<"\n Value of X is :  " << f[i]<<" Cal Value is : "<<x[i]<<" And
                    Error is :"<<ej[i];
             fprintf (s1,"\n ");
```

```
                fprintf(s1,"%f   %f   %f   %f", r,f[i],x[i],ej[i]);
               }
        fclose(s1);
        cin.get();
}
```

Subroutine – Gauss Seidel Method

```
// program    for gauss seidel method
// solve the linear system of equations

#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
int order;

void read(double a[][20])
 {
   for(int i=1;i<=order;i++)
      for(int j=1;j<=order;j++)
         {  a[i][j]=0;
            if(i==j)
              {   if(i>1)
                  {   j--;
                      cout<<"\n Enter Value of ["<<i<<"] ["<<j<<"] th element of matrix : ";
                      cin>>a[i][j];
                       j++;
                  }
                  cout<<"\n Enter Value of ["<<i<<"] ["<<j<<"] th element of matrix : ";
                  cin>>a[i][j];
                  if(i<order)
                    {   j++;
                      cout<<"\n Enter Value of ["<<i<<"] ["<<j<<"] th element of matrix : ";
                      cin>>a[i][j];
                    }
                  }
              }
          }

void Gauss_s(double a[20][20],double b[20],double x[20],int n)
{ for(int k=0;k<n;k++)
          {  for(int i=1;i<=order;i++)
```

```
                {  x[i]=b[i];
                   for(int j=1;j<=order;j++)
                   { if (j!=i)
                     x[i] = x[i]-(a[i][j]*x[j]);
                   }
                 }
            }
    }
void main ( )
{
        double a[20][20],b[20],x[20],ej[20],f[20];
        double r,h;
        int n;
        FILE *s1;
        s1= fopen("out1.txt","w");
        clrscr();
        cout<<"\n Enter order of squre matrix  : ";
        cin>>order;
        read(a);
        for(int i=1; i<=order;i++)
        {  cout<<"\n Enter[" <<i<<"] th element of colum vector B :";
           cin>>b[i];
         }
        cout<<"\n Enter total no. of ittration :";
        cin>>n;

        /* Normalize Matrix  */

        fprintf(s1,"\n Given matrix is : \n");
        for(int i=1;i<=order;i++)
          {     for(int j=1;j<=order;j++)
                  { if (j!=i)
                          a[i][j] = a[i][j]/a[i][i];
                          cout<<"\t "<<a[i][j];
                          fprintf(s1,"\t %f ",a[i][j]);
                  }
                  b[i]=b[i]/a[i][i];
                  cout<<"\t"<<b[i];
                  fprintf(s1,"\t\t %f \n",b[i]);
                  cout<<"\n";
          }


        Gauss_s(a,b,x,n);

        cout<<"\n Approximated Soln. of the given system of Equction by using
```

```
                Gauss-shield is : \n";

        fprintf (s1,"\n Appromimated Soln. of the given system of Equaction by using
                Gauss-Shield method is : ");

        for( int i=1;i<=order;i++)
        {   cout<<"\n x["<<i<<"]"<<x[i];
            fprintf(s1, "\n");
          fprintf(s1, " %f  ",x[i]);
           }

    fclose(s1);
   cin.get();

}
```

Subroutine – Cal_Fun

```
// This is program for calculate funciton value

#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>

void main( )
 {  double x=0;
   double y;
   double a[30][30];
   FILE *s1;
        s1= fopen("data1.txt","w");
            clrscr();
     for (int i=0;i<=4;i++)
            {   y=0;
                for( int j=0;j<=4;j++)
                  { cout<<"\n" <<i << "   "<<j;
                    fprintf(s1,"\n %f  %f",x,y);
                    a[i][j]= exp(x)*cos(y);
                     cout<<"\n  Value of ["<<i<<"] ["<<j<<"] th element of matrix :
                          <<a[i][j];
                     fprintf(s1,"\t %f ",a[i][j]);
                     y=y+.1;
```

```
                    }
                  x=x+.2;
                fprintf(s1,"\n ");
              }
        fclose(s1);
cin.get();
}
```

APPENDIX III GAUSS SEIDEL METHOD

The Gauss-Seidel Method

We are considering an iterative solution to the linear system

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

where A is an $N \times N$ sparse matrix, x and b are vectors of length N and we are solving for x . Iterative solvers are an alternative to direct methods that attempt to calculate an exact solution to the system of equations. Iterative methods attempt to find a solution to the system of linear equations by repeatedly solving the linear system using approximations to the vector. Iterations continue until the solution is within a predetermined acceptable bound on the error.

Common iterative methods for general matrices include the Gauss-Jacobi and Gauss-Seidel, while conjugate gradient methods exist for positive definite matrices. Critical in the choice and use of iterative methods is the convergence of the technique. Gauss-Jacobi uses all values from the previous iteration, while Gauss-Seidel requires that the most recent values be used in calculations. The Gauss-Seidel method generally has better convergence than the Gauss-Jacobi method, although for dense matrices, the Gauss-Seidel method is inherently sequential. Better convergence means less iteration, and a faster overall algorithm, as long as the strict precedence rules can be observed. The

convergence of the iterative method must be examined for the application along with algorithm performance to ensure that a useful solution to $Ax = b$ can be found.

The Gauss-Seidel method can be written as:

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^k \right) \qquad (2)$$

where, x_j^k is the i^{th} unknown in during the k^{th} iteration i = 1,2,….,n and k = 0,1,….

$x_i^{(0)} = 0$ is the initial guess for the i^{th} unknown ,

a_{ij} is the coefficient of A in the i^{th} row and j^{th} column,

b_i is the i^{th} value in b.

or, $$x^{k+1} = (D+L)^{-1}[b - Ux^{(k)}] \qquad (3)$$

where, $x^{(k)}$ is the k^{th} iterative solution to x, k = 0,1,2,……

$x^{(0)}$ is the initial guess at x, D is the diagonal of A, L is the of strictly lower triangular portion of A, U is the of strictly upper triangular portion of A, and b is right-hand-side vector.

The representation in equation 2 is used in the development of the parallel algorithm, while the equivalent matrix-based representation in equation 3 is used below in discussions of available parallelism.

The Convergence of Gauss-Seidel Method

The Gauss- Seidel method converges to the solution of $Ax = b$ if

$$r = \max_i \sum_{\substack{j=1 \\ j \neq 1}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$$

(that is, A is strictly diagonally dominant.)

PROOF : Assuming that if the method converges then $x_m \to x$, we have for the component form and its limit

$$x_i^{(m+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(m+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(m)} + \frac{f_i}{a_{ii}}$$

$$x_i = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j + \frac{f_i}{a_{ii}}$$

which yields after subtraction and setting $e_m = x_m - x$,

$$e_i^{(m+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} e_j^{(m+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} e_j^{(m)}$$

we now prove by induction on i, the component index, that $[\![e_{m+1}]\!]_\infty \leq r [\![e_m]\!]_\infty$, m = 0,1,2,.....

$$e_1^{(m+1)} = -\sum_{j=2}^{n} \frac{a_{1j}}{a_{11}} e_j^m$$

so;

$$\left| e_1^{(m+1)} \right| \leq \sum_{j=2}^{n} \left| \frac{a_{1j}}{a_{11}} \right| \bullet \left| e_j^{(m)} \right|$$

$$\leq [\![e_m]\!]_\infty \sum_{j=2}^{n} \left| \frac{a_{1j}}{a_{11}} \right|$$

$$\leq r [\![e_m]\!]_\infty$$

Assume that $\left|e_1^{(m+1)}\right| \le r\|e_m\|_\infty$ for k = 1,2,3,…,i-1

$$\left|e_1^{(m+1)}\right| \le \sum_{j=1}^{i-1}\left|\frac{a_{ij}}{a_{ii}}\right|\left|e_j^{(m+1)}\right| + \sum_{j=i+1}^{n}\left|\frac{a_{ij}}{a_{ii}}\right|\left|e_j^{(m)}\right|$$

$$\le r\|e_m\|_\infty \sum_{j=1}^{i-1}\left|\frac{a_{ij}}{a_{ii}}\right| + \|e_m\|_\infty \sum_{j=i+1}^{n}\left|\frac{a_{ij}}{a_{ii}}\right|$$

$$< \|e_m\|_\infty \sum_{\substack{j=1 \\ j\neq i}}^{i-1}\left|\frac{a_{ij}}{a_{ii}}\right|$$

$$\le r\|e_m\|_\infty .$$

Hence $\|e_{m+1}\|_\infty \le r\|e_m\|_\infty$, since all the components of e_{m+1} are bounded by this and $\|e_m\|_\infty \le r^m\|e_0\|_\infty \to 0$ as $m \to \infty$. The characteristic equation for Gauss-Seidel is

$$\det\left[\lambda I - (D-L)^{-1}U\right] = 0$$

$$\Rightarrow \left[\lambda(D-L) - U\right] = 0$$

or in displayed form

$$\begin{bmatrix} a_{11}\lambda & a_{12} & \cdots\cdots & a_{1n} \\ a_{21}\lambda & a_{22}\lambda & \cdots\cdots & \cdots\cdot \\ \vdots & & & a_{n-1,n} \\ a_{n1}\lambda & & a_{n,n-1}\lambda & a_{n,n}\lambda \end{bmatrix} = 0$$

REFERENCES

[1] BURDEN , RICHARD L., FAIRES, J. DOUGLAS, and REYNOLDS, ALBERT C. (1978). Numerical analysis. Prindle Weber & Schmidt

[2] STRANG, GILBERT, and FIX, GEORGE J. (1973), An analysis of the finite element method. Prentice-Hall. Englewood Cliffs, N.J.

[3] VARGA, RICHAD S. (1962), Matrix iterative analysis. Prientice-Hall, Engleweed Cliffs, N.J.

[4] WACHSPRESS, EUGENE L. (1966), Iterative solution of elliptic systems. Prientice-Hall, Engleweed Cliffs, N.J.,.

[5] CHARLES A. HALL; THOMAS A. PORSCHING (1990), Numerical Analysis of Partial Differential Equations. Prientice-Hall, Engleweed Cliffs, N.J.

[6] GEORGE LINDFIELD , JOHN PENNY (2000) Numerical Methods using Mat-Lab Prentice-Hall, Upper Saddle River, N.J.

VITA

Graduate College
University of Nevada, Las Vegas

Virendra I. Jaiswal

Local Address:
1555 E. Rochelle Ave; Unit # 171,
Las Vegas, NV 89119.

Home Address:
A-1, Sai Duplex,
Opp; Sarad-Kayna High School,
Vishawamitry Rd,
Baroda. 390011,
Gujarat INDIA.

Degrees:
Bachelor of Science, Mathematics, 2000
Mahraja Sayaji-Rao University, Baroda.

Thesis Title:
Numerical Solution of Boundary Value Problem Using A Variational Formation

Thesis Examination Committee:
Chairperson, Dr. Rohan J. Dalpatadu, Ph.D.
Committee Member, Dr. Ashok K. Singh, Ph.D.
Committee Member, Dr. Xin Li, Ph.D.
Graduate Faculty Representative, Dr. Sahjendra Singh, Ph.D.