2005

A numerical method for solving double integral equations

Afshin Tiraie
San Jose State University

A NUMERICAL METHOD FOR SOLVING

DOUBLE INTEGRAL EQUATIONS

A Thesis

Presented to

The Faculty of the Department of Mathematics

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

by

Afshin Tiraie

December 2005

APPROVED FOR THE DEPARTMENT OF MATHEMATICS

Mohammad Saleem

Dr. Mohammad Saleem

Igor Malyshev

Dr. Igor Malyshev

Samih Obaid

Dr. Samih Obaid

APPROVED FOR THE UNIVERSITY

Thea I. Williamson 12/21/05

ABSTRACT

A NUMERICAL METHOD FOR SOLVING
DOUBLE INTEGRAL EQUATIONS

by Afshin Tiraie

Integral equations are of special interest in physics and applied mathematics, and since in general, they have no analytical solution, numerical approaches to solving them have great importance. Multi-integral equations, specifically, are very difficult to solve, and numerical methods for solving them are limited. In this thesis, we introduce a numerical scheme which could be used in solving various types of double-integral equations. After converting the equation to a discrete form, we use the centers of the cells to evaluate the double integral and thus the solution. Examples are provided, and errors are computed by comparing the numerical and exact solutions.

ACKNOWLEDGMENT

I need to thank Dr. Mohammad Saleem, my advisor, who introduced numerical calculus to me. I am also grateful to Dr. Igor Malyshev and Dr. Samih Obaid; in the busiest time of the semester, they read the manuscripts thoroughly, and provided valuable suggestions.

TABLE OF CONTENTS

INTRODUCTION

Integral equations are of great importance in mathematics, and arise in various scientific researches. Differential equations – which appear almost in every discipline – under suitable conditions may be transformed into integral equations. Although one-dimensional integral equations have been studied in detail ([1], [2], and [4]), and for solving them numerically, various efficient methods have been devised, multi-dimensional integral equations have not been treated greatly, and there is not plenty of literature about them. In one dimension, there are several efficient methods for approximating the integral of functions [5]. Although, usually, these methods are employed to solve one-dimensional integral equations, they are not easily generalized to higher dimensions. And, probably, this is the reason for rendering multi-dimensional integral equations. (One exception is [6] where a class of Volterra integral equations has been treated through specific methods.)

In this thesis, through solving a particular equation, we introduce a numerical method which may be applied to solving double integral equations in general. The method constructs systems of equations by assuming the equality of the value of the unknown function at some point with the average of the values of the function at neighboring points. The systems of linear equations, then, are solved to estimate the value of the function at the selected points. This scheme is analogous to the Crank-Nicholson method for solving the heat equation which has been presented in [7].

CHAPTER 1

A Double Integral Equation and Its Numerical Solution

In this chapter, we derive a numerical method to solve the equation

$$f(x,y) = g(x,y) + \int_c^y \int_a^b f(s,t)G(s,t)dsdt \tag{1.1}$$

for f over the rectangle $R = \{(x,y) \mid a \le x \le b,\ c \le y \le d\}$, where g and G are known, G is continuous, and f is assumed to be continuous over R. The reader notes that the double integral may be expressed as a function of y alone. This, however, will not alter the essence of the problem, and will not simplify it.

Let m and n be large positive integers, $h = \dfrac{b-a}{n}$, and $k = \dfrac{d-c}{m}$. Let $\{x_0, x_1, \ldots, x_i, \ldots, x_n\}$ and $\{y_0, y_1, \ldots, y_j, \ldots, y_m\}$ be partitions of $[a,b]$ and $[c,d]$ where $x_i = a + ih$ and $y_j = c + jk$. Then

$$f(x_i, y_j) = g(x_i, y_j) + \int_{y_0}^{y_j} \int_{x_0}^{x_n} f(s,t)G(s,t)dsdt \tag{1.2}$$

is a discrete form of (1.1) where f is required at each point (x_i, y_j) of the rectangle R.

When $j = 0$, for any i

$$f(x_i, y_0) = g(x_i, y_0) + \int_{y_0}^{y_0} \int_{x_0}^{x_n} f(s,t)G(s,t)dsdt = g(x_i, y_0). \tag{1.3}$$

When $j = 1$, we have

$$f(x_0, y_1) = g(x_0, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,, \tag{1.4}$$

$$f(x_1, y_1) = g(x_1, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,, \tag{1.5}$$

$$f(x_2, y_1) = g(x_2, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,, \tag{1.6}$$

$$\vdots$$

$$f(x_{n-2}, y_1) = g(x_{n-2}, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,, \tag{1.7}$$

$$f(x_{n-1}, y_1) = g(x_{n-1}, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,, \tag{1.8}$$

and

$$f(x_n, y_1) = g(x_n, y_1) + \int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt\,. \tag{1.9}$$

Subtracting the corresponding sides of (1.5) – (1.9) from (1.4) – (1.8) accordingly, we obtain

$$f(x_0, y_1) - f(x_1, y_1) = g(x_0, y_1) - g(x_1, y_1), \tag{1.10}$$

$$f(x_1, y_1) - f(x_2, y_1) = g(x_1, y_1) - g(x_2, y_1), \tag{1.11}$$

$$f(x_2, y_1) - f(x_3, y_1) = g(x_2, y_1) - g(x_3, y_1), \tag{1.12}$$

$$\vdots$$

$$f(x_{n-2}, y_1) - f(x_{n-1}, y_1) = g(x_{n-2}, y_1) - g(x_{n-1}, y_1), \tag{1.13}$$

and

$$f(x_{n-1}, y_1) - f(x_n, y_1) = g(x_{n-1}, y_1) - g(x_n, y_1)\,. \tag{1.14}$$

For the n+1 unknowns $f(x_0,y_1), f(x_1,y_1), f(x_2,y_1),\ldots,$ and $f(x_n,y_1)$, (1.10) – (1.14) provide n linear equations; seeking another equation, we approximate $\int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt$.

Letting $\xi_1 = \frac{x_0+x_1}{2}$, and $\psi_1 = \frac{y_0+y_1}{2}$, and assuming that $f(\xi_1,\psi_1)$ is the average of the values of f at (x_0,y_0), (x_1,y_0), (x_1,y_1), and (x_0,y_1), that is the value of f at the center of the rectangle with vertices (x_0,y_0), (x_1,y_0), (x_1,y_1), and (x_1,y_0), namely at (ξ_1,ψ_1), equals the average of the values of f at the vertices, we have

$$f(\xi_1,\psi_1) = \frac{f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)+f(x_1,y_1)}{4}. \tag{1.15}$$

Letting $\xi_i = \frac{x_{i-1}+x_i}{2}$ for i = 2, 3, …, n, similarly, we could write

$$f(\xi_2,\psi_1) = \frac{f(x_1,y_0)+f(x_1,y_1)+f(x_2,y_0)+f(x_2,y_1)}{4}, \tag{1.16}$$

$$f(\xi_3,\psi_1) = \frac{f(x_2,y_0)+f(x_2,y_1)+f(x_3,y_0)+f(x_3,y_1)}{4}, \tag{1.17}$$

$$\vdots$$

$$f(\xi_{n-2},\psi_1) = \frac{f(x_{n-3},y_0)+f(x_{n-3},y_1)+f(x_{n-2},y_0)+f(x_{n-2},y_1)}{4}, \tag{1.18}$$

$$f(\xi_{n-1},\psi_1) = \frac{f(x_{n-2},y_0)+f(x_{n-2},y_1)+f(x_{n-1},y_0)+f(x_{n-1},y_1)}{4}, \tag{1.19}$$

and

$$f(\xi_n,\psi_1) = \frac{f(x_{n-1},y_0)+f(x_{n-1},y_1)+f(x_n,y_0)+f(x_n,y_1)}{4}. \tag{1.20}$$

As h and k are small – since n and m are large – we have

$$\int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt \approx$$

$$f(\xi_1,\psi_1)G(\xi_1,\psi_1)hk + f(\xi_2,\psi_1)G(\xi_2,\psi_1)hk + f(\xi_3,\psi_1)G(\xi_3,\psi_1)hk + \ldots +$$

$$f(\xi_{n-2},\psi_1)G(\xi_{n-2},\psi_1)hk + f(\xi_{n-1},\psi_1)G(\xi_{n-1},\psi_1)hk + f(\xi_n,\psi_1)G(\xi_n,\psi_1)hk\,,$$

and using (1.14) – (1.19),

$$\int_{y_0}^{y_1}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt \approx$$

$$\frac{f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)+f(x_1,y_1)}{4}G(\xi_1,\psi_1)hk +$$

$$\frac{f(x_1,y_0)+f(x_1,y_1)+f(x_2,y_0)+f(x_2,y_1)}{4}G(\xi_2,\psi_1)hk +$$

$$\frac{f(x_2,y_0)+f(x_2,y_1)+f(x_3,y_0)+f(x_3,y_1)}{4}G(\xi_3,\psi_1)hk +$$

$$\vdots$$

$$\frac{f(x_{n-3},y_0)+f(x_{n-3},y_1)+f(x_{n-2},y_0)+f(x_{n-2},y_1)}{4}G(\xi_{n-2},\psi_1)hk +$$

$$\frac{f(x_{n-2},y_0)+f(x_{n-2},y_1)+f(x_{n-1},y_0)+f(x_{n-1},y_1)}{4}G(\xi_{n-1},\psi_1)hk +$$

$$\frac{f(x_{n-1},y_0)+f(x_{n-1},y_1)+f(x_n,y_0)+f(x_n,y_1)}{4}G(\xi_n,\psi_1)hk\,. \qquad (1.21)$$

Employing (1.21) in (1.9), we reach

$$f(x_n,y_1) \approx g(x_n,y_1) + \frac{hk}{4}\{$$

$$[f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)+f(x_1,y_1)]G(\xi_1,\psi_1) +$$

$$[f(x_1,y_0)+f(x_1,y_1)+f(x_2,y_0)+f(x_2,y_1)]G(\xi_2,\psi_1) +$$

$$[f(x_2,y_0)+f(x_2,y_1)+f(x_3,y_0)+f(x_3,y_1)]G(\xi_3,\psi_1) +$$

$$\vdots$$

$$[f(x_{n-2},y_0)+f(x_{n-2},y_1)+f(x_{n-1},y_0)+f(x_{n-1},y_1)]G(\xi_{n-2},\psi_1) +$$

$$[f(x_{n-2},y_0)+f(x_{n-2},y_1)+f(x_{n-1},y_0)+f(x_{n-1},y_1)]G(\xi_{n-1},\psi_1)+$$

$$[f(x_{n-1},y_0)+f(x_{n-1},y_1)+f(x_n,y_0)+f(x_n,y_1)]G(\xi_n,\psi_1)\}.$$

Assuming $\approx$ equality, and collecting the unknowns, this becomes

$$f(x_n,y_1)=g(x_n,y_1)+\frac{hk}{4}\{$$

$$[f(x_0,y_0)+f(x_1,y_0)]G(\xi_1,\psi_1)+[f(x_1,y_0)+f(x_2,y_0)]G(\xi_2,\psi_1)+$$

$$[f(x_2,y_0)+f(x_3,y_0)]G(\xi_3,\psi_1)+\ldots+[f(x_{n-3},y_0)+f(x_{n-2},y_0)]G(\xi_{n-2},\psi_1)+$$

$$[f(x_{n-2},y_0)+f(x_{n-1},y_0)]G(\xi_{n-1},\psi_1)+[f(x_{n-1},y_0)+f(x_n,y_0)]G(\xi_n,\psi_1)$$

$$+$$

$$f(x_0,y_1)G(\xi_1,\psi_1)+f(x_1,y_1)[G(\xi_1,\psi_1)+G(\xi_2,\psi_1)]+f(x_2,y_1)[G(\xi_2,\psi_1)+G(\xi_3,\psi_1)]+$$

$$\ldots+f(x_{n-2},y_1)[G(\xi_{n-2},\psi_1)+G(\xi_{n-1},\psi_1)]+f(x_{n-1},y_1)[G(\xi_{n-1},\psi_1)+G(\xi_n,\psi_1)]+$$

$$f(x_n,y_1)G(\xi_n,\psi_1)\}.$$

For reasons that will appear below, we add $f(x_0,y_0)-g(x_0,y_0)$ – which equals 0 – to $g(x_n,y_1)$ in the left hand side of (1.21); thus, we arrive at

$$f(x_n,y_1)=g(x_n,y_1)+f(x_0,y_0)-g(x_0,y_0)+\frac{hk}{4}\{$$

$$[f(x_0,y_0)+f(x_1,y_0)]G(\xi_1,\psi_1)+[f(x_1,y_0)+f(x_2,y_0)]G(\xi_2,\psi_1)+$$

$$[f(x_2,y_0)+f(x_3,y_0)]G(\xi_3,\psi_1)+\ldots+[f(x_{n-3},y_0)+f(x_{n-2},y_0)]G(\xi_{n-2},\psi_1)+$$

$$[f(x_{n-2},y_0)+f(x_{n-1},y_0)]G(\xi_{n-1},\psi_1)+[f(x_{n-1},y_0)+f(x_n,y_0)]G(\xi_n,\psi_1)$$

$$+$$

$$f(x_0,y_1)G(\xi_1,\psi_1)+f(x_1,y_1)[G(\xi_1,\psi_1)+G(\xi_2,\psi_1)]+f(x_2,y_1)[G(\xi_2,\psi_1)+G(\xi_3,\psi_1)]+$$

$$\ldots+f(x_{n-2},y_1)[G(\xi_{n-2},\psi_1)+G(\xi_{n-1},\psi_1)]+f(x_{n-1},y_1)[G(\xi_{n-1},\psi_1)+G(\xi_n,\psi_1)]+$$

$$f(x_n,y_1)G(\xi_n,\psi_1). \tag{1.22}$$

Multiplying both sides by -4, moving the unknowns to the left, and arranging them in the ascending order of x's indices, we find

$$hkG(\xi_1,\psi_1)f(x_0,y_1)+hk[G(\xi_1,\psi_1)+G(\xi_2,\psi_1)]f(x_1,y_1)+$$

$$hk[G(\xi_2,\psi_1)+G(\xi_3,\psi_1)]f(x_2,y_1)+\ldots+hk[G(\xi_{n-2},\psi_1)+G(\xi_{n-1},\psi_1)]f(x_{n-2},y_1)$$

$$+hk[G(\xi_{n-1},\psi_1)+G(\xi_n,\psi_1)]f(x_{n-1},y_1)+4[hkG(\xi_n,\psi_1)-1]f(x_n,y_1)$$

$$=$$

$$-\{4[g(x_n,y_1)+f(x_0,y_0)-g(x_0,y_0)]+hk\{$$

$$[f(x_0,y_0)+f(x_1,y_0)]G(\xi_1,\psi_1)+[f(x_1,y_0)+f(x_2,y_0)]G(\xi_2,\psi_1)+$$

$$[f(x_0,y_0)+f(x_1,y_0)]G(\xi_1,\psi_1)+[f(x_1,y_0)+f(x_2,y_0)]G(\xi_2,\psi_1)+$$

$$[f(x_2,y_0)+f(x_3,y_0)]G(\xi_3,\psi_1)+\ldots+[f(x_{n-3},y_0)+f(x_{n-2},y_0)]G(\xi_{n-2},\psi_1)+$$

$$[f(x_{n-2},y_0)+f(x_{n-1},y_0)]G(\xi_{n-1},\psi_1)+[f(x_{n-1},y_0)+f(x_n,y_0)]G(\xi_n,\psi_1)\}\}\,. \quad (1.23)$$

We could now form a system of $n+1$ linear equations using (1.9) – (1.13) and (1.23) as shown on the next page.

$$
\begin{cases}
f(x_0,y_1)-f(x_1,y_1)=g(x_0,y_1)-g(x_1,y_1) \\
f(x_1,y_1)-f(x_2,y_1)=g(x_1,y_1)-g(x_2,y_1) \\
f(x_2,y_1)-f(x_3,y_1)=g(x_2,y_1)-g(x_3,y_1) \\
\vdots \\
f(x_{n-2},y_1)-f(x_{n-1},y_1)=g(x_{n-2},y_1)-g(x_{n-1},y_1) \\
f(x_{n-1},y_1)-f(x_n,y_1)=g(x_{n-1},y_1)-g(x_n,y_1) \\
\\
hkG(\xi_1,\psi_1)f(x_0,y_1)+hk[G(\xi_1,\psi_1)+G(\xi_2,\psi_1)]f(x_1,y_1)+ \\
hk[G(\xi_2,\psi_1)+G(\xi_3,\psi_1)]f(x_2,y_1)+\ldots+hk[G(\xi_{n-2},\psi_1)+G(\xi_{n-1},\psi_1)]f(x_{n-2},y_1)+ \\
hk[G(\xi_{n-1},\psi_1)+G(\xi_n,\psi_1)]f(x_{n-1},y_1)+4[hkG(\xi_n,\psi_1)-1]f(x_n,y_1) \\
= \\
-\{4[g(x_n,y_1)+f(x_0,y_0)-g(x_0,y_0)]+hk\{ \\
[f(x_0,y_0)+f(x_1,y_0)]G(\xi_1,\psi_1)+[f(x_1,y_0)+f(x_2,y_0)]G(\xi_2,\psi_1)+ \\
[f(x_2,y_0)+f(x_3,y_0)]G(\xi_3,\psi_1)+\ldots+[f(x_{n-3},y_0)+f(x_{n-2},y_0)]G(\xi_{n-2},\psi_1)+ \\
[f(x_{n-2},y_0)+f(x_{n-1},y_0)]G(\xi_{n-1},\psi_1)+[f(x_{n-1},y_0)+f(x_n,y_0)]G(\xi_n,\psi_1)\}\}.
\end{cases}
\tag{1.24}
$$

We now tackle the general case where $j > 0$, $f(x_i, y_{j-1})$ is known, and $f(x_i, y_j)$ is required for every i. What follows is parallel to what we saw above. We have

$$f(x_0,y_j)=g(x_0,y_j)+\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt, \tag{1.25}$$

$$f(x_1,y_j)=g(x_1,y_j)+\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt, \tag{1.26}$$

$$f(x_2,y_j)=g(x_2,y_j)+\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt, \tag{1.27}$$

$$\vdots$$

$$f(x_{n-2},y_j)=g(x_{n-2},y_j)+\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt, \tag{1.28}$$

$$f(x_{n-1},y_j)=g(x_{n-1},y_j)+\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt, \tag{1.29}$$

and

$$f(x_n, y_j) = g(x_n, y_j) + \int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt. \tag{1.30}$$

Next, (1.25) – (1.30) yield

$$f(x_0, y_j) - f(x_1, y_j) = g(x_0, y_j) - g(x_1, y_j), \tag{1.31}$$

$$f(x_1, y_j) - f(x_2, y_j) = g(x_1, y_j) - g(x_2, y_j), \tag{1.32}$$

$$f(x_2, y_j) - f(x_3, y_j) = g(x_2, y_j) - g(x_3, y_j), \tag{1.33}$$

$$\vdots$$

$$f(x_{n-2}, y_j) - f(x_{n-1}, y_j) = g(x_{n-2}, y_j) - g(x_{n-1}, y_j), \tag{1.34}$$

and

$$f(x_{n-1}, y_j) - f(x_n, y_j) = g(x_{n-1}, y_j) - g(x_n, y_j). \tag{1.35}$$

For the n+1 unknowns $f(x_0, y_j), f(x_1, y_j), f(x_2, y_j), \ldots,$ and $f(x_n, y_j)$, (1.31) – (1.35) provide n linear equations; seeking another equation, we approximate $\int_{y_0}^{y_k}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt$.

But $\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt = \int_{y_0}^{y_{j-1}}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt + \int_{y_{j-1}}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt$, and

$\int_{y_0}^{y_{j-1}}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt = f(x_0, y_{j-1}) - g(x_0, y_{j-1})$; therefore

$$\int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt = f(x_0, y_{j-1}) - g(x_0, y_{j-1}) + \int_{y_{j-1}}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt. \tag{1.36}$$

Letting $\psi_j = \dfrac{y_{j-1} + y_j}{2}$, and assuming that $f(\xi_1, \psi_j)$ is the average of the values of f at (x_0, y_{j-1}), (x_1, y_{j-1}), (x_1, y_j), and (x_0, y_j), we could write

$$f(\xi_1, \psi_j) = \frac{f(x_0, y_{j-1}) + f(x_0, y_j) + f(x_1, y_{j-1}) + f(x_1, y_j)}{4}. \tag{1.37}$$

Similarly, we have

$$f(\xi_2, \psi_j) = \frac{f(x_1, y_{j-1}) + f(x_1, y_j) + f(x_2, y_{j-1}) + f(x_2, y_j)}{4}, \tag{1.38}$$

$$f(\xi_3, \psi_j) = \frac{f(x_2, y_{j-1}) + f(x_2, y_j) + f(x_3, y_{j-1}) + f(x_3, y_j)}{4}, \tag{1.39}$$

$$\vdots$$

$$f(\xi_{n-2}, \psi_j) = \frac{f(x_{n-3}, y_{j-1}) + f(x_{n-3}, y_j) + f(x_{n-2}, y_{j-1}) + f(x_{n-2}, y_j)}{4}, \tag{1.40}$$

$$f(\xi_{n-1}, \psi_j) = \frac{f(x_{n-2}, y_{j-1}) + f(x_{n-2}, y_j) + f(x_{n-1}, y_{j-1}) + f(x_{n-1}, y_j)}{4}, \tag{1.41}$$

and

$$f(\xi_n, \psi_j) = \frac{f(x_{n-1}, y_{j-1}) + f(x_{n-1}, y_j) + f(x_n, y_{j-1}) + f(x_n, y_j)}{4}. \tag{1.42}$$

Then, we use the approximation

$$\int_{y_{j-1}}^{y_j} \int_{x_0}^{x_n} f(s,t) G(s,t)\, ds dt \approx$$

$$\frac{f(x_0, y_{j-1}) + f(x_0, y_j) + f(x_1, y_{j-1}) + f(x_1, y_j)}{4} G(\xi_1, \psi_j) hk +$$

$$\frac{f(x_1, y_{j-1}) + f(x_1, y_j) + f(x_2, y_{j-1}) + f(x_2, y_j)}{4} G(\xi_2, \psi_j) hk +$$

$$\frac{f(x_2,y_{j-1})+f(x_2,y_j)+f(x_3,y_{j-1})+f(x_3,y_j)}{4}G(\xi_3,\psi_j)hk+$$

$$\vdots$$

$$\frac{f(x_{n-3},y_{j-1})+f(x_{n-3},y_j)+f(x_{n-2},y_{j-1})+f(x_{n-2},y_j)}{4}G(\xi_{n-2},\psi_j)hk+$$

$$\frac{f(x_{n-2},y_{j-1})+f(x_{n-2},y_j)+f(x_{n-1},y_{j-1})+f(x_{n-1},y_j)}{4}G(\xi_{n-1},\psi_j)hk+$$

$$\frac{f(x_{n-1},y_{j-1})+f(x_{n-1},y_j)+f(x_n,y_{j-1})+f(x_n,y_j)}{4}G(\xi_n,\psi_j)hk\,,$$

(1.30), and (1.36) to arrive at

$$f(x_n,y_j)\approx g(x_n,y_j)+f(x_0,y_{j-1})-g(x_0,y_{j-1})+\frac{hk}{4}\{$$

$$[f(x_0,y_{j-1})+f(x_0,y_j)+f(x_1,y_{j-1})+f(x_1,y_j)]G(\xi_1,\psi_j)+$$

$$[f(x_1,y_{j-1})+f(x_1,y_j)+f(x_2,y_{j-1})+f(x_2,y_j)]G(\xi_2,\psi_j)+$$

$$[f(x_2,y_{j-1})+f(x_2,y_j)+f(x_3,y_{j-1})+f(x_3,y_j)]G(\xi_3,\psi_j)+$$

$$\vdots$$

$$[f(x_{n-2},y_{j-1})+f(x_{n-2},y_j)+f(x_{j-1},y_k)+f(x_{n-1},y_j)]G(\xi_{n-2},\psi_j)+$$

$$[f(x_{n-2},y_{j-1})+f(x_{n-2},y_j)+f(x_{n-1},y_{j-1})+f(x_{n-1},y_j)]G(\xi_{n-1},\psi_j)+$$

$$[f(x_{n-1},y_{j-1})+f(x_{n-1},y_j)+f(x_n,y_{j-1})+f(x_n,y_j)]G(\xi_n,\psi_j)\}\,.$$

Assuming $\approx$ equality, and collecting the unknowns, this becomes

$$f(x_n,y_j)=g(x_n,y_j)+f(x_0,y_{j-1})-g(x_0,y_{j-1})+\frac{hk}{4}\{$$

$$[f(x_0,y_{j-1})+f(x_1,y_{j-1})]G(\xi_1,\psi_j)+[f(x_1,y_{j-1})+f(x_2,y_{j-1})]G(\xi_2,\psi_j)+$$

$$[f(x_2,y_{j-1})+f(x_3,y_{j-1})]G(\xi_3,\psi_j)+\ldots+[f(x_{n-3},y_{j-1})+f(x_{n-2},y_{j-1})]G(\xi_{n-2},\psi_j)+$$

$$[f(x_{n-2},y_{j-1})+f(x_{n-1},y_{j-1})]G(\xi_{n-1},\psi_j)+[f(x_{n-1},y_{j-1})+f(x_n,y_{j-1})]G(\xi_n,\psi_j)$$

$$+$$

$$f(x_0,y_j)G(\xi_1,\psi_j)+f(x_1,y_j)[G(\xi_1,\psi_j)+G(\xi_2,\psi_j)]+f(x_2,y_j)[G(\xi_2,\psi_j)+G(\xi_3,\psi_j)]+$$
$$\ldots+f(x_{n-2},y_j)[G(\xi_{n-2},\psi_j)+G(\xi_{n-1},\psi_j)]+f(x_{n-1},y_j)[G(\xi_{n-1},\psi_j)+G(\xi_n,\psi_j)]+$$
$$f(x_n,y_j)G(\xi_n,\psi_j)\}. \qquad (1.43)$$

But (1.43) is just (1.22) with j-1 added to the indices of y and ψ. It follows that merely adding j-1 to the indices of y and ψ in (1.23), we could reach

$$hkG(\xi_1,\psi_j)f(x_0,y_j)+hk[G(\xi_1,\psi_j)+G(\xi_2,\psi_j)]f(x_1,y_j)+$$
$$hk[G(\xi_2,\psi_j)+G(\xi_3,\psi_j)]f(x_2,y_j)+\ldots+hk[G(\xi_{n-2},\psi_j)+G(\xi_{n-1},\psi_j)]f(x_{n-2},y_j)+$$
$$hk[G(\xi_{n-1},\psi_j)+G(\xi_n,\psi_j)]f(x_{n-1},y_j)+4[hkG(\xi_n,\psi_j)-1]f(x_n,y_j)$$
$$=$$
$$-\{4[g(x_n,y_j)+f(x_1,y_{j-1})-g(x_1,y_{j-1})]+hk\{$$
$$[f(x_0,y_{j-1})+f(x_1,y_{j-1})]G(\xi_1,\psi_j)+[f(x_1,y_{j-1})+f(x_2,y_{j-1})]G(\xi_2,\psi_j)+$$
$$[f(x_2,y_{j-1})+f(x_3,y_{j-1})]G(\xi_3,\psi_2)+\ldots+[f(x_{n-3},y_{j-1})+f(x_{n-2},y_{j-1})]G(\xi_{n-2},\psi_j)+$$
$$[f(x_{n-2},y_{j-1})+f(x_{n-1},y_{j-1})]G(\xi_{n-1},\psi_j)+[f(x_{n-1},y_{j-1})+f(x_n,y_{j-1})]G(\xi_n,\psi_j)\}\} \qquad (1.44)$$

Employing (1.31) – (1.35) and (1.44), we could now form the system

$$
\left\{
\begin{array}{c}
f(x_0,y_j)-f(x_1,y_j)=g(x_0,y_j)-g(x_1,y_j) \\
f(x_1,y_j)-f(x_2,y_j)=g(x_1,y_j)-g(x_2,y_j) \\
f(x_2,y_j)-f(x_3,y_j)=g(x_2,y_j)-g(x_3,y_j) \\
\vdots \\
f(x_{n-2},y_j)-f(x_{n-1},y_j)=g(x_{n-2},y_j)-g(x_{n-1},y_j) \\
f(x_{n-1},y_j)-f(x_n,y_j)=g(x_{n-1},y_j)-g(x_n,y_j) \\
\\
hkG(\xi_1,\psi_j)f(x_0,y_j)+hk[G(\xi_1,\psi_j)+G(\xi_2,\psi_j)]f(x_1,y_j)+ \\
hk[G(\xi_2,\psi_j)+G(\xi_3,\psi_j)]f(x_2,y_j)+\ldots+hk[G(\xi_{n-2},\psi_j)+G(\xi_{n-1},\psi_j)]f(x_{n-2},y_j)+ \\
hk[G(\xi_{n-1},\psi_j)+G(\xi_n,\psi_j)]f(x_{n-1},y_j)+4[hkG(\xi_n,\psi_j)-1]f(x_n,y_j) \\
= \\
-\{4[g(x_n,y_j)+f(x_1,y_{j-1})-g(x_1,y_{j-1})]+hk\{ \\
[f(x_0,y_{j-1})+f(x_1,y_{j-1})]G(\xi_1,\psi_j)+[f(x_1,y_{j-1})+f(x_2,y_{j-1})]G(\xi_2,\psi_j)+ \\
[f(x_2,y_{j-1})+f(x_3,y_{j-1})]G(\xi_3,\psi_2)+\ldots+[f(x_{n-3},y_{j-1})+f(x_{n-2},y_{j-1})]G(\xi_{n-2},\psi_j)+ \\
[f(x_{n-2},y_{j-1})+f(x_{n-1},y_{j-1})]G(\xi_{n-1},\psi_j)+[f(x_{n-1},y_{j-1})+f(x_n,y_{j-1})]G(\xi_n,\psi_j)\}\}.
\end{array}
\right. \quad (1.45)
$$

To express (1.45) in matrix form, we let

$$
\vec{x}_j=\begin{bmatrix} f(x_0,y_j) \\ f(x_1,y_j) \\ f(x_2,y_j) \\ \vdots \\ f(x_{n-2},y_j) \\ f(x_{n-1},y_j) \\ f(x_n,y_j) \end{bmatrix}.
$$

Also, we let

$$A_j = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \\ a_j^1 & a_j^2 & a_j^3 & \cdots & a_j^{n-1} & a_j^n & a_j^{n+1} \end{bmatrix}_{(n+1)\times(n+1)}$$

where a's superscript corresponds to its column, $a_j^1 = hkG(\xi_1, \psi_j)$,

$$a_j^2 = hk[G(\xi_1,\psi_j) + G(\xi_2,\psi_j)],\ a_j^3 = hk[G(\xi_2,\psi_j) + G(\xi_3,\psi_j)],\ \ldots,$$

$$a_j^{n-1} = hk[G(\xi_{n-2},\psi_j) + G(\xi_{n-1},\psi_j)],\ a_j^n = hk[G(\xi_{n-1},\psi_j) + G(\xi_n,\psi_j)],$$

and

$$a_j^{n+1} = 4[hkG(\xi_n,\psi_j) - 1].$$

Finally, we set

$$\vec{b}_j = \begin{bmatrix} g(x_0,y_j) - g(x_1,y_j) \\ g(x_1,y_j) - g(x_2,y_j) \\ g(x_2,y_j) - g(x_3,y_j) \\ \vdots \\ g(x_{n-2},y_j) - g(x_{n-1},y_j) \\ g(x_{n-1},y_j) - g(x_n,y_j) \\ b_j \end{bmatrix}$$

where

$$\begin{aligned} b_j = -\{4[g(x_n,y_j) + f(x_0,y_{j-1}) - g(x_0,y_{j-1})] + hk\{[f(x_0,y_{j-1}) + f(x_1,y_{j-1})]G(\xi_1,\psi_j) + \\ [f(x_1,y_{j-1}) + f(x_2,y_{j-1})]G(\xi_2,\psi_j) + [f(x_2,y_{j-1}) + f(x_3,y_{j-1})]G(\xi_3,\psi_j) + \ldots + \\ [f(x_{n-3},y_{j-1}) + f(x_{n-2},y_{j-1})]G(\xi_{n-2},\psi_j) + [f(x_{n-2},y_{j-1}) + f(x_{n-1},y_{j-1})]G(\xi_{n-1},\psi_j) + \\ [f(x_{n-1},y_{j-1}) + f(x_n,y_{j-1})]G(\xi_n,\psi_j)\}\}. \end{aligned}$$

Then, (1.45) could be expressed by

$$A_j \vec{x}_j = \vec{b}_j . \tag{1.46}$$

Note that since G is continuous over R, it is bounded [3], hence as $m, n \to \infty$, or $h, k \to 0$, each A_j will acquire the form of an upper-triangular matrix with all the diagonal entries except the last equal to 1, and the last equal to -4. Thus, for large m and n, and for any j = 1, 2, …, m, (1.46) has unique solution.

The above discussion provides an algorithm for solving (1.2): "One, initially, needs to find $f(x_0, y_0), f(x_1, y_0), \ldots, f(x_n, y_0)$ via (1.3). Then, for j = 1, 2, …, m, one has to solve (1.45), or (1.48), to find $f(x_0, y_j), f(x_1, y_j), \ldots, f(x_n, y_j)$."

CHAPTER 2

Numerical Results and Errors

The algorithm presented in Chapter 1 has been implemented in a Matlab program which appears in Appendix. To compare the output of the program with exact solutions, we need to distinguish between $f(x_i, y_j)$ and the value which is computed through the algorithm at (x_i, y_j); we denote the latter by $u_{i,j}$. In particular, $u_{0,1}$, $u_{1,1}$, ..., $u_{n,1}$ satisfy (1.24), and in general

$$\vec{u}_j = \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{n,j} \end{bmatrix}$$

satisfies (1.46). Thus, defining $\varepsilon_j = u_{n,j} - f(x_n, y_j)$, $|\varepsilon_j|$ will be the error of the algorithm at (x_n, y_j). It is notable that since

$$u_{n-1,j} - u_{n,j} = g(x_{n-1}, y_j) - g(x_n, y_j) =$$

$$f(x_{n-1}, y_j) - \int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt - [f(x_n, y_j) - \int_{y_0}^{y_j}\int_{x_0}^{x_n} f(s,t)G(s,t)dsdt] =$$

$$f(x_{n-1}, y_j) - f(x_n, y_j),$$

we find $u_{n-1,j} - f(x_{n-1}, y_j) = u_{n,j} - f(x_n, y_j) = \varepsilon_j$. Employing the same reasoning repeatedly, we could see

$$u_{0,j} - f(x_0, y_j) = u_{1,j} - f(x_{n-1}, y_j) = \ldots = u_{n,j} - f(x_n, y_j) = \varepsilon_j. \tag{2.1}$$

This allows us to define the approximation error at a fixed j by $e_j = |\varepsilon_j|$.

Example 1: We first show the results produced by the program when $R=[0,1]\times[0,1]$, $G(x,y)=-xy$, and $g(x,y)=x^2+\frac{y^2}{8}(y^2+9)$, in which case $f(x,y)=x^2+y^2$ analytically.

When $m=n=5$, the program provides the following matrix

$$\begin{bmatrix} 0 & 0.0372 & 0.1481 & 0.3312 & 0.5842 & 0.9047 \\ 0.0400 & 0.0772 & 0.1881 & 0.3712 & 0.6242 & 0.9447 \\ 0.1600 & 0.1972 & 0.3081 & 0.4912 & 0.7442 & 1.0647 \\ 0.3600 & 0.3972 & 0.5081 & 0.6912 & 0.9442 & 1.2647 \\ 0.6400 & 0.6772 & 0.7881 & 0.9712 & 1.2242 & 1.5447 \\ 1.0000 & 1.0372 & 1.1481 & 1.3312 & 1.5842 & 1.9047 \end{bmatrix}.$$

This is $[\vec{u}_0\ \vec{u}_1\ \vec{u}_2\ \vec{u}_3\ \vec{u}_4\ \vec{u}_5]$. Computing the exact solution (via $f(x,y)=x^2+y^2$) at the corresponding points, we obtain

$$\begin{bmatrix} 0 & 0.0400 & 0.1600 & 0.3600 & 0.6400 & 1.0000 \\ 0.0400 & 0.0800 & 0.2000 & 0.4000 & 0.6800 & 1.0400 \\ 0.1600 & 0.2000 & 0.3200 & 0.5200 & 0.8000 & 1.1600 \\ 0.3600 & 0.4000 & 0.5200 & 0.7200 & 1.0000 & 1.3600 \\ 0.6400 & 0.6800 & 0.8000 & 1.0000 & 1.2800 & 1.6400 \\ 1.0000 & 1.0400 & 1.1600 & 1.3600 & 1.6400 & 2.0000 \end{bmatrix}.$$

Subtracting this from the preceding matrix gives

$$\begin{bmatrix} 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \\ 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \\ 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \\ 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \\ 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \\ 0 & -0.0028 & -0.0119 & -0.0288 & -0.0558 & -0.0953 \end{bmatrix}$$

which was expected by (2.1). Table 1 demonstrates the convergence of the solutions computed by the program to the exact solution.

$m = n =$	5	10	20	50	100	200
e_1	0.0028	3.5924e-4	4.5801e-5	2.9710e-6	3.7316e-7	4.6759e-8
Max. e_j	0.0953	0.0492	0.0249	0.0100	0.0050	0.0025

Table 1. Error Comparison for Different m and n when $f(x,y) = x^2 + y^2$

Example 2: Table 2, as well, exhibits the convergence of the approximated solutions. It shows the results produced by the program when $R = [0,1]\times[0,1]$, $G(x,y) = xy$, and $g(x,y) = y^2(x - \frac{y^2}{12})$, in which case $f(x,y) = xy^2$ analytically.

$m = n =$	5	10	20	50	100	200
e_1	1.0749e-4	3.5474e-6	1.1397e-7	1.1867e-9	3.7292e-11	1.1686e-12
Max. e_j	0.0476	0.0222	0.0106	0.0042	0.0021	0.0010

Table 2. Error Comparison for Different m and n when $f(x,y) = xy^2$

CHAPTER 3

Conclusion and Future Extensions

As mentioned earlier, the method we have introduced may be utilized to solve other double integral equations such as

$$f(x,y) = g(x,y) + \int_c^y \int_a^b f(s,t)G(x-s,y-t)dsdt,$$

and

$$f(x,y) = g(x,y) + \int_c^y \int_a^x f(s,t)G(s,t)dsdt \tag{3.1}$$

under the assumptions that hold on (1.1). (The reader notes that in the first equation the variables of G are $x-s$ and $y-t$, and in the second equation the upper limit of the inner integral is x.) We shall provide a sketch of the solution for (3.1). We assume the same domain and conditions that hold on (1.1), and use the same partitions and indices as on (1.2). Then,

$$f(x_i,y_j) = g(x_i,y_j) + \int_{y_0}^{y_j} \int_{x_0}^{x_n} f(s,t)G(s,t)dsdt \tag{3.2}$$

is a discrete form of (3.1) where f is required at each point (x_i, y_j).

Initially, note that since $\int_c^y \int_a^a f(s,t)G(s,t)dsdt = \int_c^c \int_a^x f(s,t)G(s,t)dsdt = 0$, we have $f(a,y) = g(a,y)$, and $f(x,c) = g(x,c)$; hence, $f(x_0,y_j) = g(x_0,y_j)$ for every j (= 1, 2, ..., m), and $f(x_i,y_0) = g(x_i,y_0)$ for every i (= 0, 1, ..., n). We have

$$f(x_1,y_1)=g(x_1,y_1)+\int_{y_0}^{y_1}\int_{x_0}^{x_1}f(s,t)G(s,t)dsdt\,. \tag{3.3}$$

Assuming that the value of f at the center of the rectangle with vertices (x_0,y_0), (x_1,y_0), (x_1,y_1), and (x_1,y_0) equals the average of the values of f at the vertices, we set

$$\int_{y_0}^{y_1}\int_{x_0}^{x_1}f(s,t)G(s,t)dsdt=\frac{f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)+f(x_1,y_1)}{4}G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk\,.$$

Plugging this into (3.3) we find

$$f(x_1,y_1)=g(x_1,y_1)+\frac{f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)+f(x_1,y_1)}{4}G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk\,.$$

Noting that the only unknown in this equation is $f(x_1,y_1)$, we write

$$[4-G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk]f(x_1,y_1)=$$

$$4g(x_1,y_1)+[f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)]G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk\,.$$

Finally, we obtain

$$f(x_1,y_1)=\frac{4g(x_1,y_1)+[f(x_0,y_0)+f(x_0,y_1)+f(x_1,y_0)]G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk}{4-G(\frac{x_0+x_1}{2},\frac{y_0+y_1}{2})hk}\,.$$

Proceeding to the next x, we have

$$f(x_2,y_1)=g(x_2,y_1)+\int_{y_0}^{y_1}\int_{x_0}^{x_2}f(s,t)G(s,t)dsdt\,. \tag{3.4}$$

But

$$\int_{y_0}^{y_1}\int_{x_0}^{x_2}f(s,t)G(s,t)dsdt=\int_{y_0}^{y_1}\int_{x_0}^{x_1}f(s,t)G(s,t)dsdt+\int_{y_0}^{y_1}\int_{x_1}^{x_2}f(s,t)G(s,t)dsdt\,, \tag{3.5}$$

and

$$\int_{y_0}^{y_1}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt = f(x_1,y_1) - g(x_1,y_1)\,. \tag{3.6}$$

Now setting

$$\int_{y_0}^{y_1}\int_{x_1}^{x_2} f(s,t)G(s,t)dsdt =$$

$$\frac{f(x_1,y_0)+f(x_1,y_1)+f(x_2,y_0)+f(x_2,y_1)}{4}G(\frac{x_1+x_2}{2},\frac{y_0+y_1}{2})hk \tag{3.7}$$

and utilizing (3.5) – (3.7) in (3.4), we obtain the equation

$$f(x_2,y_1) = g(x_2,y_1) + f(x_1,y_1) - g(x_1,y_1) +$$

$$\frac{f(x_1,y_0)+f(x_1,y_1)+f(x_2,y_0)+f(x_2,y_1)}{4}G(\frac{x_1+x_2}{2},\frac{y_0+y_1}{2})hk\,.$$

The only unknown here is $f(x_2,y_1)$, and the equation could be solved for it. This process may be repeated until every $f(x_i,y_1)$ is found. We can then advance to the next j. We have

$$f(x_1,y_2) = g(x_1,y_2) + \int_{y_0}^{y_2}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt\,. \tag{3.8}$$

But,

$$\int_{y_0}^{y_2}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt = \int_{y_0}^{y_1}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt + \int_{y_1}^{y_2}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt \tag{3.9}$$

and

$$\int_{y_0}^{y_1}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt = f(x_1,y_1) - g(x_1,y_1)\,. \tag{3.10}$$

Setting

$$\int_{y_1}^{y_2}\int_{x_0}^{x_1} f(s,t)G(s,t)dsdt =$$

$$\frac{f(x_0,y_1)+f(x_0,y_2)+f(x_1,y_1)+f(x_1,y_2)}{4}G\left(\frac{x_1+x_2}{2},\frac{y_0+y_1}{2}\right)hk, \qquad (3.11)$$

and combining (3.8) – (3.11) we reach the equation.

$$f(x_1,y_2) = g(x_1,y_2)+f(x_1,y_1)-g(x_1,y_1)+$$

$$\frac{f(x_0,y_1)+f(x_0,y_2)+f(x_1,y_1)+f(x_1,y_2)}{4}G\left(\frac{x_1+x_2}{2},\frac{y_0+y_1}{2}\right)hk \qquad (3.12)$$

Again, there is only one unknown in this equation, namely $f(x_1,y_2)$, and the equation could be solved for it. Obviously, this process may be repeated until every $f(x_i,y_2)$ is found; then we can advance to the next j, and so on.

We believe that this method could be employed in solving integral equations in higher dimensions. In triple integral equations, for instance, approximating the value of the required function at the center of each infinitesimal cuboid (or rectangular parallelepiped) by the average of the values of the function at the vertices of the cuboid will provide first degree equations that are indispensable for solving the integral equation numerically.

Finally, we suggest performing von Neumann's method to verify the stability of numerical schemes employed in solving (1.1) or (3.1) which may be accomplishable under some conditions. (Von Neumann's method for PDEs is discussed extensively in [4].)

REFERENCES

[1] Delves, L. M. & Mohamed, J. L. *Computational methods for integral equations*, Cambridge University Press, Cambridge, 1985.

[2] Kanwal, R. P. *Linear Integral Equations* , Birkhauser, New York , 1996.

[3] Kreyszig, E. *Introductory Functional Analysis* John Wile & Sons, New York, 1985.

[4] Linz, P. *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.

[5] Maron, M. J. & Lopez, R. J. *Numerical Analysis*, Wadsworth Publishing Company, Belmont, 1991.

[6] McKee, S. Tang, Tao. & Diogo, T. *An Euler-type Method for Two-dimensional Volterra Integral Equations of the First Kind*, IMA Journal of Numerical Analysis, 20:423-440, 2000.

[7] Smith, G. D. *Numerical Solutions of Partial Differential Equations*, Clarendon Press, Oxford, 2003.

[8] Matlab Version 6 – Software Package by MATHWORKS, 2003.

APPENDIX

The Program

```
% This program solves the discrete form of the equation
% f(x,y) = g(x,y) + Int Int f(s,t) G(s,t) ds dt
% where Int denotes integral, and f is required; the inner
% integral's limits being a and b, and the outer one's c and y.
clear all;
% Functions to be used to test the program when f(x,y) = x*x + y*y.
f = inline('x*x + y*y', 'x', 'y');
G = inline('-x*y', 'x', 'y');
g = inline('x*x + (y*y/8)*(y*y+9)', 'x', 'y');
% Functions to be used to test the program when f(x,y) = x*y*y.
%f = inline('x*y*y', 'x', 'y');
%G = inline('x*y', 'x', 'y');
%g = inline('y*y*(x - y*y/12)', 'x', 'y');
a = 0;
b = 1;
c = 0;
d = 1;
n = 5;
m = 5;
N = n+1; % The number of points in the x-axis partition.
M = m+1; % The number of points in the y-axis partition.
% Create appropriate partitions on the axes.
x = linspace(a,b,N);
y = linspace(c,d,M);
h = x(2) - x(1);
k = y(2) - y(1);
```

```
hk = h*k;
half_h = h/2;
half_k = k/2;
U = zeros(N,M); % The solution matrix.
for i  = 1 : N
   U(i,1) = g(x(i), y(1));
end
b_vector = zeros(N,1);
% Create A of the size N*N.
A = eye(N);
for i  = 1 : n
   A(i,i+1) = -1;
end
error = 0;
for j = 2 : M
 jMinus1 = j - 1;
 psi = y(j) - half_k;
 ksi =  x(1) + half_h;
 A(N,1) = hk * G(ksi, psi);
 b_vector(1) = g(x(1), y(j)) - g(x(2), y(j));
 b_vector(N) = 4 * ( g(x(N), y(j)) + U(1, jMinus1) - g(x(1), y(jMinus1)) );
 sum = 0;
 for i  = 2 : n
    iMinus1 = i - 1;
    A(N,i) = hk * ( G(ksi, psi) + G(ksi + h, psi) );
    b_vector(i) = g(x(i), y(j)) - g(x(i+1), y(j));
    sum = sum + ( U(iMinus1, jMinus1) + U(i, jMinus1) ) * G(ksi, psi);
    ksi = ksi + h;
 end
```

```
    A(N,N) = 4 * ( hk * G(ksi, psi) - 1 );
    sum = sum + ( U(n, jMinus1) + U(N, jMinus1) ) * G(ksi, psi);
    b_vector(N) = - ( b_vector(N) + hk*sum );
    U(1:N, j) = A\b_vector;
    tempError = abs ( U(1,j) - f(x(1),y(j)) );
    if error < tempError
        error = tempError;
    end
end
% Print the first error of approximation.
abs( U(N,2) - f(x(N),y(2)) )
% Print the maximum error of approximation.
error
```