THE DIRECT SOLUTION OF THE DISCRETE POISSON EQUATION ON IRREGULAR REGIONS

BY

B. L. BUZBEE

F. W. DORR

J. A. GEORGE

G. H. GOLUB

STAN-CS-71-195

DECEMBER, 1970

COMPUTER SCIENCE DEPARTMENT

School of Humanities and Sciences

STANFORD UNIVERSITY

THE DIRECT SOLUTION OF THE DISCRETE POISSON EQUATION ON IRREGULAR REGIONS

by

B. L. Buzbee*, F. W. Dorr, J. A. George, G. H. Golub

To be issued simultaneously as Los Alamos Scientific Laboratory Report LA 4553.

Computer Science Department, Stanford University, Stanford, California 94305. Research supported by the Office of Naval Research under grant No. N0013-67-A-00112-0029, and by the Atomic Energy Commission under grant No. AT(04-3)326, PA30.

* Los Alamos Scientific Laboratory of the University of California, Los Alamos, New Mexico 87544. Research supported by the U. S. Atomic Energy Commission under Contract No. W-7405-ENG-36.

Abstract

There are several very fast direct methods which can be used to solve the discrete Poisson equation on rectangular domains. We show that these methods can also be used to treat problems on irregular regions.

1. Introduction. Within the past few years, several very fast and accurate direct methods have been developed for solving finite difference approximations to the Poisson equation,

$$\left.\begin{aligned} Au &= f \quad \text{in } R, \\ u &= g \quad \text{on } \partial R . \end{aligned}\right\}$$

These methods can usually be applied only on rectangular regions, although the differential operator and boundary conditions can be more general than those in the Poisson equation. In this paper, we will show how these algorithms for rectangular domains can also be used effectively on irregular regions. The approach used is similar to that employed by Hockney [16, 17], Buneman [7], and George [14]. We also mention the work of Angel [1-4], Angel and Kalaba [5], Collins and Angel [9], Kalaba [20], and Roache [22] on the use of direct methods for problems in irregular regions.

We will not discuss the details of any specific direct method. A survey of these procedures is given in [11], and in particular we cite the recent work of Buneman [6], Buzbee, Golub, and Nielson [8], and Hockney [16].

We will also not consider the derivation of the finite difference equations that approximate the partial differential equation. This subject is treated in detail by Forsythe and Wasow [13], and we assume that the problem has been reduced to finding the solution of a matrix equation $A\underset{\sim}{x} = y$. The matrix A is frequently very large and sparse, but its structure does not permit the application of the most efficient direct methods. For our computational procedure, we alter certain rows of A to obtain a matrix B, and we will show how to define a modified right-hand side z so that the solution $\underset{\sim}{x}$ also satisfies the equation $B\underset{\sim}{x} = \underset{\sim}{z}$. The matrix B is chosen so that these equations can be solved by the direct methods.

This method is computationally advantageous when we are solving a sequence of equations $A\underset{\sim}{x}_i = \underset{\sim}{y}_i$. This situation frequently arises in time-dependent partial differential equations, in nonlinear problems, and in linear problems where the right-hand side is varied but the region and differential operator remain the same. After some initial computation, each solution $\underset{\sim}{x}_i$ can be obtained in approximately twice the time required for the solution of an equation $B\underset{\sim}{x} = z$.

In Sections 2 and 3 we derive this algorithm in a general form. We describe a number of applications of the method in Sections 4 and 5, and in Section 6 we present some computational results.

2. Method of Solution if $\det B \neq 0$. Suppose that we are given an n by n matrix A and an integer p with $1 \leq p \leq n$. We wish to modify p rows of A to obtain another matrix B. Without loss of generality we assume that the first p rows of A are to be changed, since we can achieve this situation by multiplying A by a suitable permutation matrix. However, we emphasize that this multiplication should not be done explicitly in the computational procedure. Rather, the rearrangement of rows should be done implicitly by indexing. The direct methods mentioned later in the paper require that B has a particular structure, which could be altered by the permutation transformation.

Partition A in the form

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix},$$

where Al is a p by n matrix and A_2 is an (n-p) by n matrix. We then write

$$B = \begin{pmatrix} B_1 \\ A_2 \end{pmatrix},$$

where B_1 is a p by n matrix. For the remainder of this section, we assume that $\det B \neq 0$.

Suppose we are given a linear equation $A\underset{\sim}{x} = \underset{\sim}{y}$. We partition $\underset{\sim}{y}$ in the same way as A, and write

$$\underset{\sim}{y} = \begin{pmatrix} \underset{\sim}{y}_1 \\ \underset{\sim}{y}_2 \end{pmatrix}.$$

Let $\bar{\underset{\sim}{y}}$ be any vector of the form

$$\bar{\underset{\sim}{y}} = \begin{pmatrix} \underset{\sim}{y}_1 \\ \underset{\sim}{y}_2 \end{pmatrix} .$$

If W is an arbitrary nonsingular p by p matrix, we define an n by p matrix $\bar{W}$ by

$$\bar{W} = \begin{pmatrix} W \\ 0 \end{pmatrix} .$$

Define the p by p matrix C by

$$C = A_1 B^{-1} \bar{W} .$$

Following Hockney [16], we call C the capacitance matrix.[*] Assume that there exists a p by 1 Vector $\underset{\sim}{\beta}$ that is a solution to the equation

$$C \underset{\sim}{\beta} = \underset{\sim}{y}_1 - A_1 B^{-1} \bar{\underset{\sim}{y}} . \tag{1}$$

Since A and B differ only in the first p rows, it is easy to verify that a solution $\underset{\sim}{x}$ to the equation $A\underset{\sim}{x} = \underset{\sim}{y}$ is given by

$$\underset{\sim}{x} = B^{-1} (\bar{\underset{\sim}{y}} + \bar{W} \underset{\sim}{\beta}) .$$

We first show that this method of obtaining the solution $\underset{\sim}{x}$ will be valid whenever the original system $A\underset{\sim}{x} = \underset{\sim}{y}$ is consistent.

Theorem. If $\det B \neq 0$, then

$$\det C = \frac{(\det A)(\det W)}{\det B} .$$

[*] Hockney actually refers to C^{-1} as the capacitance matrix. Since C may be singular in our development, we have adopted the present notation.

If the system $A\underset{\sim}{x} = \underset{\sim}{y}$ is consistent, then Eq. (1) is also consistent.

Proof. Partition B^{-1} in the form

$$B^{-1} = \begin{pmatrix} D_1 & D_2 \end{pmatrix} ,$$

where D_1 is n by p and D_2 is n by $(n-p)$. It then follows that

$$BB^{-1} = \begin{pmatrix} B_1 D_1 & B_1 D_2 \\ A_2 D_1 & A_2 D_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} ,$$

and

$$AB^{-1} = \begin{pmatrix} A_1 D_1 & A_1 D_2 \\ A_2 D_1 & A_2 D_2 \end{pmatrix} = \begin{pmatrix} A_1 D_1 & A_1 D_2 \\ 0 & I \end{pmatrix} .$$

Thus we have

$$C = A_1 B^{-1} \overline{W} = A_1 D_1 W ,$$

and so

$$\begin{aligned} \det C &= \det (A_1 D_1) \det W \\ &= \det (A B^{-1}) \det W \\ &= \frac{(\det A)(\det W)}{\det B} . \end{aligned}$$

To prove the consistency statement, suppose $C^T \underset{\sim}{\gamma} = \underset{\sim}{0}$. Write

$$AB^{-1} = \begin{pmatrix} C W^{-1} & E_1 \\ 0 & I \end{pmatrix} ,$$

and define an n by 1 vector $\bar{\underset{\sim}{r}}$ by

$$\bar{\underset{\sim}{r}} = \begin{pmatrix} \underset{\sim}{r} \\ -E_1^T \underset{\sim}{r} \end{pmatrix} .$$

We then have

$$\left(AB^{-1}\right)^T \bar{\underset{\sim}{r}} = \begin{pmatrix} \left(W^{-1}\right)^T C^T & 0 \\ E_1^T & I \end{pmatrix} \begin{pmatrix} \underset{\sim}{r} \\ -E_1^T \underset{\sim}{r} \end{pmatrix} = \underset{\sim}{0} .$$

Since the system $A\underset{\sim}{x} = \underset{\sim}{y}$ is assumed to be consistent, we therefore have $\bar{\underset{\sim}{r}}^T \underset{\sim}{y} = 0$, which is the same as $\underset{\sim}{r}^T (\underset{\sim}{y}_1 - E_1 \underset{\sim}{y}_2) = 0$. But then

$$\begin{aligned} \underset{\sim}{r}^T \left(\underset{\sim}{y}_1 - A_1 B^{-1} \bar{\underset{\sim}{y}}\right) &= \underset{\sim}{r}^T \left(\underset{\sim}{y}_1 - CW^{-1} \bar{\underset{\sim}{y}}_1 - E_1 \underset{\sim}{y}_2\right) \\ &= - \left(C^T \underset{\sim}{r}\right)^T W^{-1} \bar{\underset{\sim}{y}}_1 \\ &= 0 \quad , \end{aligned}$$

which is the' consistency condition for Eq. (1).

The Woodbury formula [18, pp. 123-124] for the inverse of a matrix $(B + FG)$ is

$$\left(B + FG\right)^{-1} = B^{-1} \left(I - F \left(I + GB^{-1}F\right)^{-1} GB^{-1}\right) .$$

This equation has been used in direct methods for solving the Poisson equation by George [14], and for the biharmonic equation by Golub [15]. If A is non-singular we write

$$A = B + FG ,$$

where $F = \bar{W}$, and G is the p by n matrix given by

$$G = W^{-1}(A_1 - B_1) .$$

For the case in which A is nonsingular, the algorithm we have derived is equivalent to using the Woodbury formula for A^{-1}.

Suppose that we have a very efficient method for solving equations of the form $B\underset{\sim}{z} = \underset{\sim}{w}$. The solution of the equation $A\underset{\sim}{x} = \underset{\sim}{y}$ then proceeds in the following steps:

(1) Compute $C = A_1 B^{-1}\bar{W}$,

(2) Compute $\bar{\underset{\sim}{x}} = B^{-1}\underset{\sim}{y}$,

(3) Solve the equation $C\underset{\sim}{\beta} = \underset{\sim}{y}_1 - A_1\bar{\underset{\sim}{x}}$.

The solution $\underset{\sim}{x}$ can then be obtained from the formula

$$\underset{\sim}{x} = B^{-1}(\underset{\sim}{y} + \bar{W}\underset{\sim}{\beta}) . \qquad (2)$$

If it is possible to store the vector $\bar{\underset{\sim}{x}}$ and the matrix $\bar{B} = B^{-1}\bar{W}$, then $\underset{\sim}{x}$ can also be computed from

$$\underset{\sim}{x} = \bar{\underset{\sim}{x}} + \bar{B}\underset{\sim}{\beta} . \qquad (3)$$

The decision whether to use Eq. (2) or Eq. (3) would be made on consideration of storage requirements, and on the-relative speed of solving the system in Eq. (2) versus multiplying by the matrix in Eq. (3). For problems arising from elliptic difference equations, it is frequently better to use Eq. (2) because B has a band structure, but the matrix $\bar{B}$ maybe full.

The type of application we have in mind for this method is one in which we have to solve a number of equations $A\underset{\sim}{x}_i = \underset{\sim}{y}_i$. In this case, we compute the capacitance matrix and factor it as part of a preprocessing stage. The solution of each equation $A\underset{\sim}{x}_i = \underset{\sim}{y}_i$ is then approximately as fast as the time

it takes to solve two equations $B\underset{\sim}{z} = \underset{\sim}{w}$.

To be specific, let $\theta(n)$ denote the number of arithmetic operations necessary to solve a system $B\underset{\sim}{z} = \underset{\sim}{w}$. Then to compute C and form its LU decomposition in a preprocessing stage requires approximately

$$p\theta(n) + k_1 p^2 n + k_2 p^3$$

operations (cf. [19, Sec. 2.11). In many cases the matrix Al is sparse, and this estimate is

$$p\theta(n) + k_3 p^3 \quad (4)$$

operations. To compute the solution to a particular equation $A\underset{\sim}{x} = \underset{\sim}{y}$ using Eq. (2) takes an additional.

$$2\theta(n) + k_4 pn + k_5 p^2$$

operations. If A_1 is sparse and we let $W = I$, this estimate can be replaced by

$$2\theta(n) + k_6 p^2 \quad (5)$$

operations. To compute a particular solution using Eq. (3) requires

$$\theta(n) + k_7 pn + k_8 P^2 \quad (6)$$

operations. In general this estimate cannot be reduced, because the matrix $\bar{B}$ may be full.

3. Method of Solution if rank(B) = n-1 . The method derived in Section 2 gives a procedure for finding a p by 1 vector δ such that a solution $\underset{\sim}{x}$ to Ax = y also satisfies the equation

$$B\underset{\sim}{x} = \underset{\sim}{y} + \begin{pmatrix} \underset{\sim}{\delta} \\ \\ \underset{\sim}{0} \end{pmatrix} \quad .$$

If B is singular, it may not be possible to find such a vector δ . To show this, suppose $B^T\underset{\sim}{v} = \underset{\sim}{0}$ but $\underset{\sim}{v} \neq \underset{\sim}{0}$. In order for δ to exist, we must satisfy the consistency condition

$$\underset{\sim}{v}^T(\underset{\sim}{y} + \sum_{i=1}^{p} \delta_i \underset{\sim}{e}_i) = 0 \quad . \tag{7}$$

If $\underset{\sim}{v}^T\underset{\sim}{e}_i = 0$ for $1 \leq i \leq p$ and $\underset{\sim}{v}^T\underset{\sim}{y} \neq 0$, it is not possible to satisfy Eq. (7) . However, if A is nonsingular this difficulty does not arise, because then the only vector v satisfying $B^T\underset{\sim}{v} = \underset{\sim}{0}$ and $\underset{\sim}{v}^T\underset{\sim}{e}_i = 0$ for $1 \leq i \leq p$ is $\underset{\sim}{v} = \underset{\sim}{0}$.

We will now describe an algorithm we have used when rank(B) = n-1 and A is nonsingular. There are two advantages in treating this particular case. First, the construction is quite simple, and it is easy to see how the method could be extended to a more general matrix B . Second, the case rank(B) = n-1 has a special significance in the solution of partial differential equations, because this condition is satisfied by the matrix corresponding to the Neumann problem. For simplicity, we assume that the matrix W of Section 2 is the identity matrix.

Theorem 2. Assume that A is nonsingular and rank(B) = n-1 , and let $\underset{\sim}{u}$ and $\underset{\sim}{v}_N$ be two non-zero vectors satisfying $B\underset{\sim}{u} = B^T\underset{\sim}{v} = \underset{\sim}{0}$. Then there exists an integer k with $1 \leq k \leq p$ such that $\underset{\sim}{v}^T\underset{\sim}{e}_k \neq 0$. Define a constant

$$\alpha = (v^T e_k)^{-1} ,$$

and let $\bar{x}$ be a solution to

$$B\bar{x} = \bar{y} - (\alpha v^T \bar{y}) e_k .$$

For $1 \le i \le p$ and $i \ne k$ let η_i be a solution to

$$B\eta_i = e_i - (\alpha v^T e_i) e_k ,$$

and let $\eta_k = u$. Let C be the p by p matrix whose i-th column is the vector $A_1 \eta_i$. Then C is nonsingular, and, if β is the solution to

$$C\beta = y_1 - A_1 \bar{x} ,$$

the solution x to $Ax = y$ is given by

$$x = \bar{x} + \sum_{i=1}^{p} \beta_i \eta_i .$$

Proof. If we partition v in the same way as y, we have $A^T v = A_1^T v_1 + A_2^T v_2$ and $B^T v = B_1^T v_1 + A_2^T v_2$. Thus if $B^T v = 0$ and $v_1 = 0$ we would have $A^T v = 0$. Since A is nonsingular and $v \ne 0$ this cannot happen, and hence $v_1 \ne 0$.

To prove that C is nonsingular, we show that $C\beta = 0$ implies $\beta = 0$. Suppose β is an arbitrary vector such that $C\beta = 0$. Then $x = \sum_{i=1}^{p} \beta_i \eta_i$ satisfies $Ax = 0$, and hence $x = 0$. This implies that $Bx = 0$, or

$$\sum_{i=1}^{p} \beta_i e_i = \alpha\left(\sum_{i=1}^{p} \beta_i v^T e_i\right) e_k .$$

Thus $\beta_i = 0$ for $1 \le i \le p$ and $i \neq k$, and the condition $x = \underset{\sim}{0}$ then implies that $\beta_k = 0$. Thus $\underset{\sim}{\beta} = \underset{\sim}{0}$, and so C is nonsingular.

Remark. As we discussed in Section 2, the computation proceeds in the following steps:

(1) Compute (and factor) C,

(2) Compute $\bar{\underset{\sim}{x}}$,

(3) Solve for $\underset{\sim}{\beta}$.

The solution $\underset{\sim}{x}$ can then be obtained from the formula

$$\underset{\sim}{x} = \bar{\underset{\sim}{x}} + \sum_{i=1}^{p} \beta_i \underset{\sim}{\eta}_i \quad .$$

However, if the problem arises from a partial differential equation, it is more efficient computationally to obtain x in the form

$$\underset{\sim}{x} = \hat{\underset{\sim}{x}} + \hat{\beta}\, u \quad ,$$

where $\hat{\underset{\sim}{x}}$ is a solution to

$$B\hat{\underset{\sim}{x}} = \bar{\underset{\sim}{y}} - (\alpha\, \underset{\sim}{v}^T \bar{\underset{\sim}{y}})\, \underset{\sim}{e}_k + \sum_{\substack{i=1 \\ i \neq k}}^{p} \beta_i (\underset{\sim}{e}_i - (\alpha\, \underset{\sim}{v}^T \underset{\sim}{e}_i)\underset{\sim}{e}_k)$$

and

$$\hat{\beta} = [\underset{\sim}{u}^T \bar{\underset{\sim}{x}} + \sum_{i=1}^{p} \beta_i\, \underset{\sim}{u}^T \underset{\sim}{\eta}_i - \underset{\sim}{u}^T \hat{\underset{\sim}{x}}][\underset{\sim}{u}^T \underset{\sim}{u}]^{-1} \quad .$$

4. Applications to Partial Differential Equations by Imbedding. Suppose we axe given a two-dimensional bounded region R in the x-y plane, and we wish to find a solution u to the Poisson equation,

$$\left.\begin{aligned} \Delta u &= f \quad \text{in } R \quad , \\ u &= g \quad \text{on } \partial R \quad . \end{aligned}\right\}$$

We ass-me that this differential equation is approximated by a finite difference equation (cf. Forsythe and Wasow [13]). Thus we have a finite set of unknowns $\{U_i \mid 1 \le i \le n_0\}$ which approximate the solution u at the grid points. If we denote by Ah a finite difference approximation to the Laplacian operator A, by R_h the discrete interior of the grid, and by ∂R_h the discrete boundary of the grid, then the discrete Poisson equation can be written in the form

$$\left.\begin{aligned} \Delta_h U &= f \quad \text{in } R_h \quad , \\ U &= g \quad \text{on } \partial R_h \quad . \end{aligned}\right\} \tag{8}$$

Let R_h' be a discrete rectangular region such that $R_h \subset R_h'$ and $\partial R_h \subset R_h' \cup \partial R_h'$, and let $S_h = \partial R_h \cap R_h'$. Extend the functions f and g to the regions R_h' and $\partial R_h \cup \partial R_h'$ respectively, and consider the equation

$$\left.\begin{aligned} \Delta_h U &= f \quad \text{in } R_h' - S_h \quad , \\ U &= g \quad \text{on } S_h \cup \partial R_h' \quad . \end{aligned}\right\} \tag{9}$$

We will solve Eq. (9), and the solution U will then also satisfy Eq. (8).

Equation (9) is a linear equation in the unknowns $\{U_i \mid 1 \le i \le n\}$. Observe that we may have increased the number of unknowns by the imbedding process, so that $n_0 \le n$. We write Eq. (9) as a matrix equation $A\underset{\sim}{U} = \underset{\sim}{V}$,

and the matrix A can frequently be chosen to be block tridiagonal with tridiagonal matrices as the non-zero blocks (cf. [13]).

Let p be the number of grid points in S_h. We modify the p rows of A and $\underset{\sim}{V}$ corresponding to the equations

$$U = g \qquad \text{on } S_h \ ,$$

and replace them with the equations

$$\Delta_h U = f \qquad \text{on } S_h \ .$$

This defines a new matrix B and a new right-hand side $\underset{\sim}{\bar{V}}$. An equation $B\underset{\sim}{U} = \underset{\sim}{\bar{V}}$ corresponds to the difference equation

$$\left.\begin{aligned} \Delta_h U &= f \qquad \text{in } R_h' \\ U &= g \qquad \text{on } \partial R_h' \end{aligned}\right\} \tag{10}$$

Since R_h' is a rectangular region, we have very fast methods for solving Eq. (10). We can now apply the method of Section 2 to solve the equation $A\underset{\sim}{U} = \underset{\sim}{V}$ by using the modified matrix B.

To illustrate this construction, let R be a rectangular region with an interior rectangle removed, such as that shown in Figure 1. For simplicity, we assume that the discrete boundary ∂R_h is a subset of ∂R. The imbedding rectangle is $R_h' = R_h \cup S_h \cup T_h$. The only function extension required for this example is that f be defined (arbitrarily) in $S_h \cup T_h$. To define this extension, we can set $f \equiv 0$ in $S_h \cup T_h$, or we can define f so that it is continuous in all of R_h'. The advantage of using a continuous f is that the solution to Eq. (10) is then smooth. However, the direct methods used to solve Eq. (10) are so accurate that the smoothness of the solution does not appear to influence the computational results. Therefore,

in the examples we have considered, we extend f by setting $f \equiv 0$ in $S_h \cup T_h$.

If we let $W = I$ in the method of Section 2, this algorithm is closely connected with the discrete Green's function for the region R_h' (cf. [13, pp. 314-318]). In fact, the method is then equivalent to adding suitable multiples of the discrete Green's function for the points on S_h so that the boundary conditions on S_h will be satisfied. Since we have Dirichlet boundary conditions on S_h, by a proper ordering of the unknowns we can write

$$A_1 = (I \quad 0) \ .$$

Since B is positive definite and

$$C = (I \quad 0)\, B^{-1} \begin{pmatrix} I \\ 0 \end{pmatrix} ,$$

we see that C is also positive definite in this case. This is advantageous because Cholesky decomposition can then be used to compute an LL^T decomposition of C (cf. [12, Chap. 23]).

If the grid on R_h' has N points on a side, we have $n = N^2$. In that case, we can solve the system $\underset{\sim}{BU} = \underset{\sim}{\bar{V}}$ in approximately

$$e(N) = 5N^2 \log_2 N$$

operations (cf. [11, p. 260]). The preprocessing then takes

$$5pN^2 \log_2 N + k_3 p^3$$

operations (cf. Eq. (4)). To solve Eq. (8) for a particular choice of f and g by using Eq. (2) with $W = I$ takes an additional

$$10N^2 \log_2 N + k_6 p^2$$

operations (cf. Eq. (5)). If we use Eq. (3) to compute the solution, it takes an additional

$$5N^2 \log_2 N + k_7 p N^2 + k_8 p^2$$

operations (cf. Eq.(6)). Thus if $p \gg \log_2 N$ it is faster to use Eq. (2) to compute the solution. We also observe that for this problem the matrix B^{-1} is full [23, p. 85], so to store $\bar{B}$ in using Eq. (3) would require pN^2 locations. Thus for large values of p and N it is both faster and more economical in terms of storage to use Eq. (2) to compute the solution to a particular equation.

It should be clear that the imbedding procedure can be applied to other elliptic difference operators with other types of boundary conditions. To be a practical procedure, we simply require that we have a fast method for solving the imbedded problem in the rectangular region.

As another example, consider the region shown in Figure 2. This problem arises in the time-dependent study of a rotating fluid [10], and the fluid surface is moving slowly. We are given Dirichlet boundary data on S_h, and Neumann boundary data on $\partial R_h - S_h$. The imbedding rectangle is $R_h' = R_h \cup S_h \cup T_h$, and we use Neumann boundary conditions on $\partial R_h'$. Thus B corresponds to the Neumann problem on R_h', and the rank of B is n-1. The method of Section 3 can then be applied, and direct methods for solving the rectangular Neumann problem are given in [8].

For an example with the Poisson operator in another geometry, consider the region in the z - r plane shown in Figure 3. This problem arises in the

time-dependent study of a plasma [21], and a Poisson equation must be solved at each time step. The boundary conditions are Dirichlet on S_h and Neumann on $\partial R_h^{(1)}$. We use Neumann boundary conditions on $\partial R_h^{(1)}$ and $\partial R_h'^{(3)}$ and Dirichlet boundary conditions on $\partial R_h'^{(2)}$ and $\partial R_h'^{(4)}$ for the imbedding region $R_h' = R_h \cup S_h \cup T_h$. The elliptic difference equation in R_h' is solved by the method of matrix decomposition [8].

5. Applications to Partial Differential Equations by Splitting. There are many problems for which the imbedding approach is not an economical algorithm. For example, imbedding the region in a rectangle may introduce an excessively large number of additional unknowns that are not necessary to the solution of the original. problem. Another instance is one in which the differential operator or the mesh size changes in different parts of the region. In this section, we give two such examples. In each case, the method of Section 2 can be used to split the problem into two rectangular problems, which can be solved by the usual direct methods.

Consider the elongated L-shaped region in Figure 4, and the equation

$$\Delta_h U = f \quad \text{in } R_h \ ,$$
$$U = g \quad \text{on } \partial R_h \ .$$

We assume that points on the line marked T_h are all grid points. To define the matrix B, we replace the equations

$$\Delta_h U = f \quad \text{on } T_h$$

by the equations

$$U = g \quad \text{on } T_h \ ,$$

where g has been (arbitrarily) extended to T_h. The solution of an equation $B\underset{\sim}{U} = \underset{\sim}{\bar{V}}$ now consists of solving the two rectangular problems

$$\Delta_h U_i = f \quad \text{in } R_h^{(i)} \ ,$$
$$U_i = g \quad \text{on } \partial R_h^{(i)} \ ,$$

for i = 1, 2. We can then apply the method of Section 2 to solve the original

problem. This algorithm is similar to one developed in [8, Sec. 9] for non-rectangular regions.

As another example, consider the multiple-material problem shown in Figure 5. The differential equation is

$$\frac{\partial}{\partial x}\left(\sigma(x)\ \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\tau(y)\ \frac{\partial u}{\partial y}\right) = f(x, y) \quad ,$$

and

$$a(x) = \begin{cases} \sigma_1(x) & \text{in } R^{(1)} \\ \sigma_2(x) & \text{in } R^{(2)} \end{cases} ,$$

The functions $\sigma_1(x)$, $\sigma_2(x)$, and $\tau(y)$ are assumed to be smooth. Dirichlet data is given on ∂R, and we require that $\sigma\frac{\partial u}{\partial x}$ be continuous across the boundary between $R^{(1)}$ and $R^{(2)}$. The computational procedure is essentially the same as that for the L-shaped region. The only difference is that in forming the matrix B we replace the equations for the continuity of $\sigma\frac{\partial u}{\partial x}$ across the line T_h by the equations

$$U = g \qquad \text{on } T_h \ .$$

As before, the equation $\underset{\sim}{B}\underset{\sim}{U} = \underset{\sim}{\bar{V}}$ corresponds to the two rectangular problems

$$\frac{a}{\partial x}\left(\sigma_i(x)\ \frac{\partial u_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(\tau(y)\ \frac{\partial u_i}{\partial y}\right) = f(x, y) \qquad \text{in } R^{(i)} ,$$

$$u_i(x, y) = g(x, y) \qquad \text{on } \partial R^{(i)} ,$$

for $i = 1, 2$. These problems can be solved directly by the method of matrix decomposition [8, Sec. 8]. A similar method can be used for the case in which $\tau(y)$ is only piecewise smooth.

It is clear that this splitting method can be applied to the Poisson equation in regions such as that in Figure 5 when different mesh sizes are used in $R^{(1)}$ and $R^{(2)}$. The method developed in [8, Sec. 8] can also be adapted to include rectangular problems with irregular meshes.

6. Computational Results. In Table 1 we have tabulated some computational results for two regions of the form of Figure 1. In each case, a square with sides of length 1 has a symmetrically located square removed from its center. For region 1 the inner square has sides of length $\frac{1}{8}$, and for region 2 the inner sides are of length $\frac{1}{4}$. We solve the Poisson equation with Dirichlet boundary conditions for the function $u(x, y) = x^4 + y^4$. This function was selected because there is no truncation error, and all of the measured error is due to inaccuracies in the solution of the difference equations. All of the computations were performed on a CDC 6600 computer.

The iterative methods used are:

SOR : point successive over-relaxation [23, p. 58],

SLOR: successive line overrelaxation [23, p. 80],

ADI : Peaceman-Rachford alternating direction implicit iteration [24, Chap. 6].

The iteration parameters used are those for the imbedding rectangle R_h', and for AD1 the parameters for cycles of length four are calculated by the Wachspress algorithm [24, Chap.6]. The initial guess is identically zero, and the iterations are terminated when the maximum difference between iterates is less than 10^{-5}.

The direct method used is variant one of the Buneman algorithm [8, Sec. 11]. Preprocessing times are given in Table 2. Computational results for a similar problem are given in [14].

The problem described in Section 4 for the region in Figure 2 has been treated by Daly and Nichols [10]. The mesh used has $23 \times 40 = 920$ points. Using the direct method of matrix decomposition, a particular solution requires about 30 - 50% of the time required for a point Gauss-Seidel iterative procedure.

The problem discussed in Section 4 for the region in Figure 3 has been treated by Morse and Rudsinski [21]. The mesh used has 52 x 98 = 5096 points, and the preprocessing time is approximately 150 seconds. The region and differential operator are very seldom changed, so the factored capacitance matrix is stored on magnetic tape. Thus there is essentially no preprocessing time for the execution of the program. To solve for a particular solution requires about 2 seconds, which is approximately 40% of the time required for a successive line overrelaxation iterative procedure.

Table 1. Computational results for solving the discrete Poisson equation.

Region	h	P	Method	Maximum Error	Computation Time (Sec.)	Scaled Computation Time
1	$\frac{1}{32}$	16	SOR	5.02 (-6)	3.586	21.866
			SLOR	7.63 (-6)	2.654	16.183
			AD1	2.36 (-6)	1.128	6.878
			Direct	4.44 (-13)	0.164	1.000
	$\frac{1}{64}$	32	SOR	8.12 (-6)	29.388	43.994
			SLOR	7.95 (-6)	21.424	32.072
			AD1	3.41 (-6)	5.642	8.446
			Direct	1.90 (-12)	0.668	1.000
2	$\frac{1}{32}$	32	SOR	2.35 (-6)	3.570	21.250
			SLOR	6.48 (-6)	2.558	15.226
			AD1	2.11 (-6)	0.870	5.179
			Direct	3.77 (-13)	0.168	1.000
	$\frac{1}{64}$	64	SOR	2.02 (-6)	29.624	43.565
			SLOR	9.96 (-6)	20.510	30.162
			AD1	3.57 (-6)	5.332	7.841
			Direct	1.54 (-12)	0.680	1.000

Table 2. Preprocessing time for the direct method results in Table i.

Region	h	Preprocessing Time (Sec.)
1	$\frac{1}{32}$	1.062
	$\frac{1}{64}$	8.670
2	$\frac{1}{32}$	2.188
	$\frac{1}{64}$	17.698

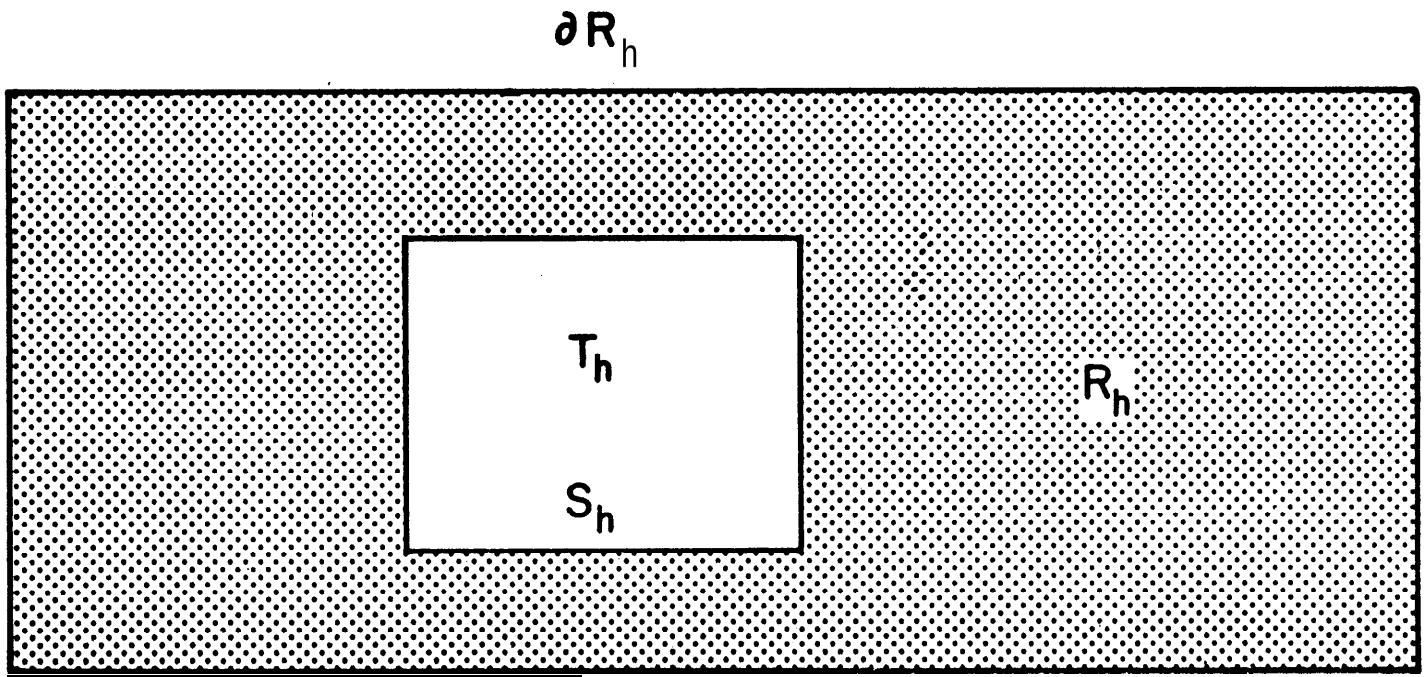

Figure 1. Region in the x-y plane.

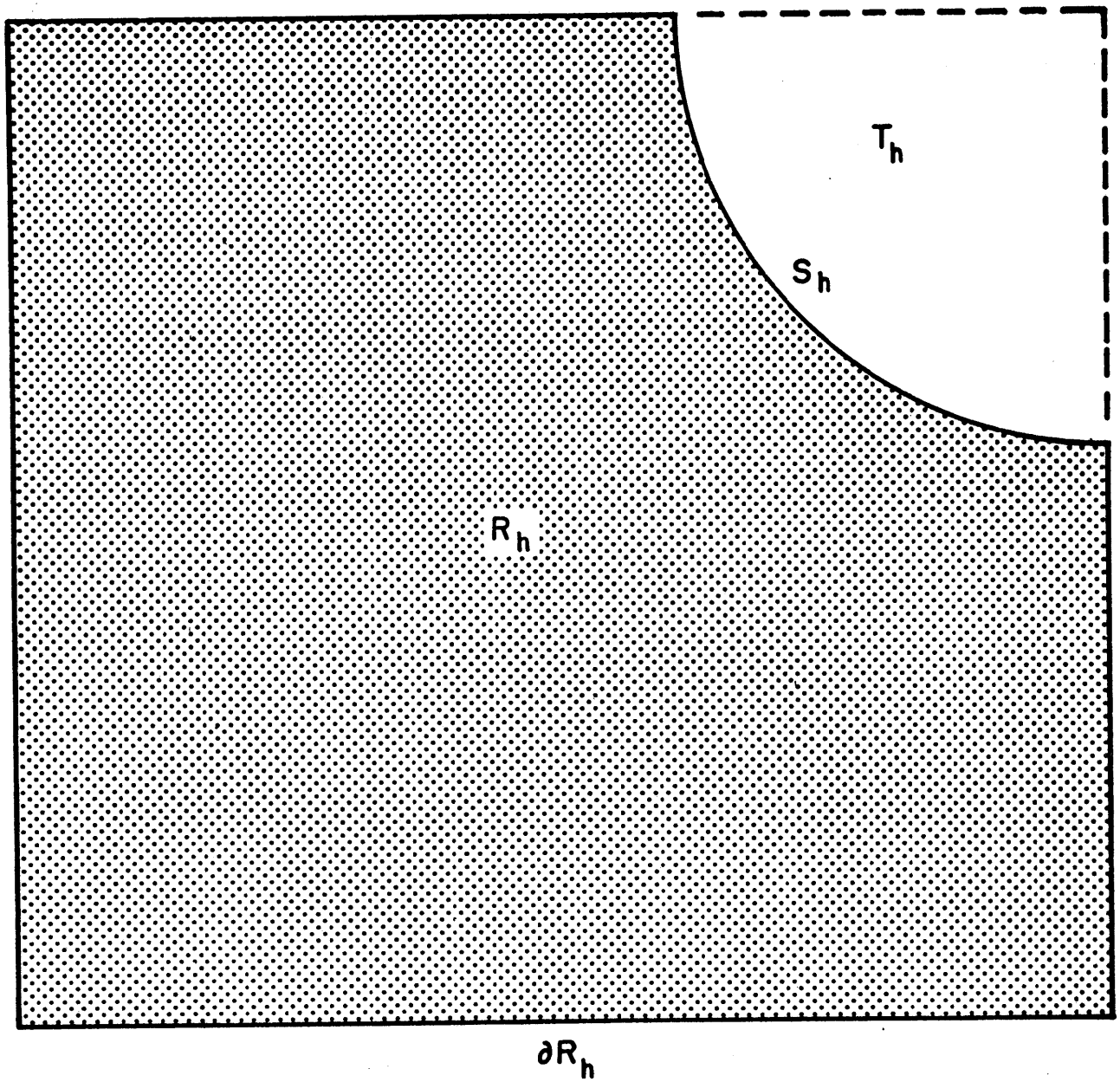

Figure 2. Region in the x-y plane.

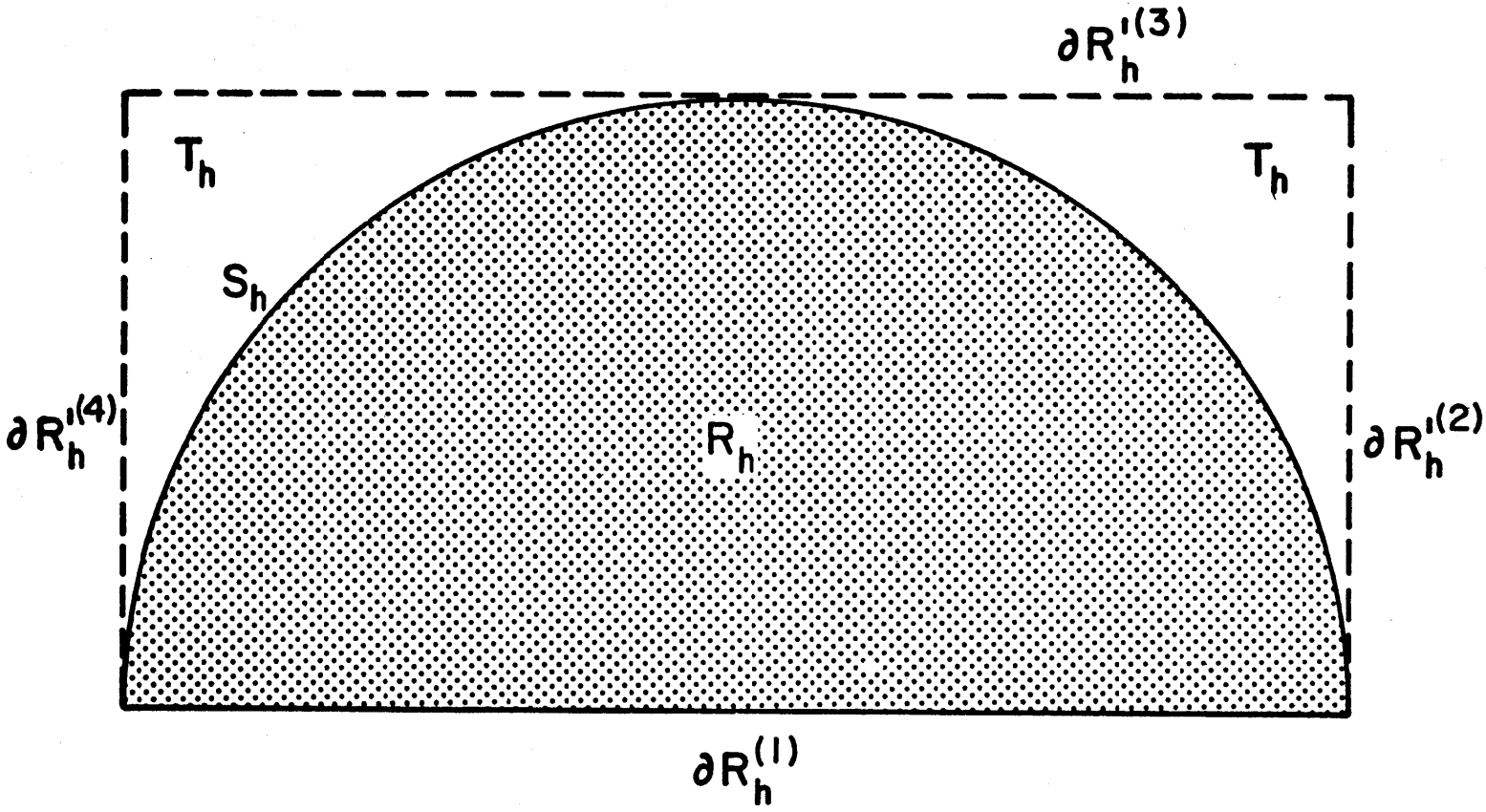

Figure 3. Region in the z-r plane.

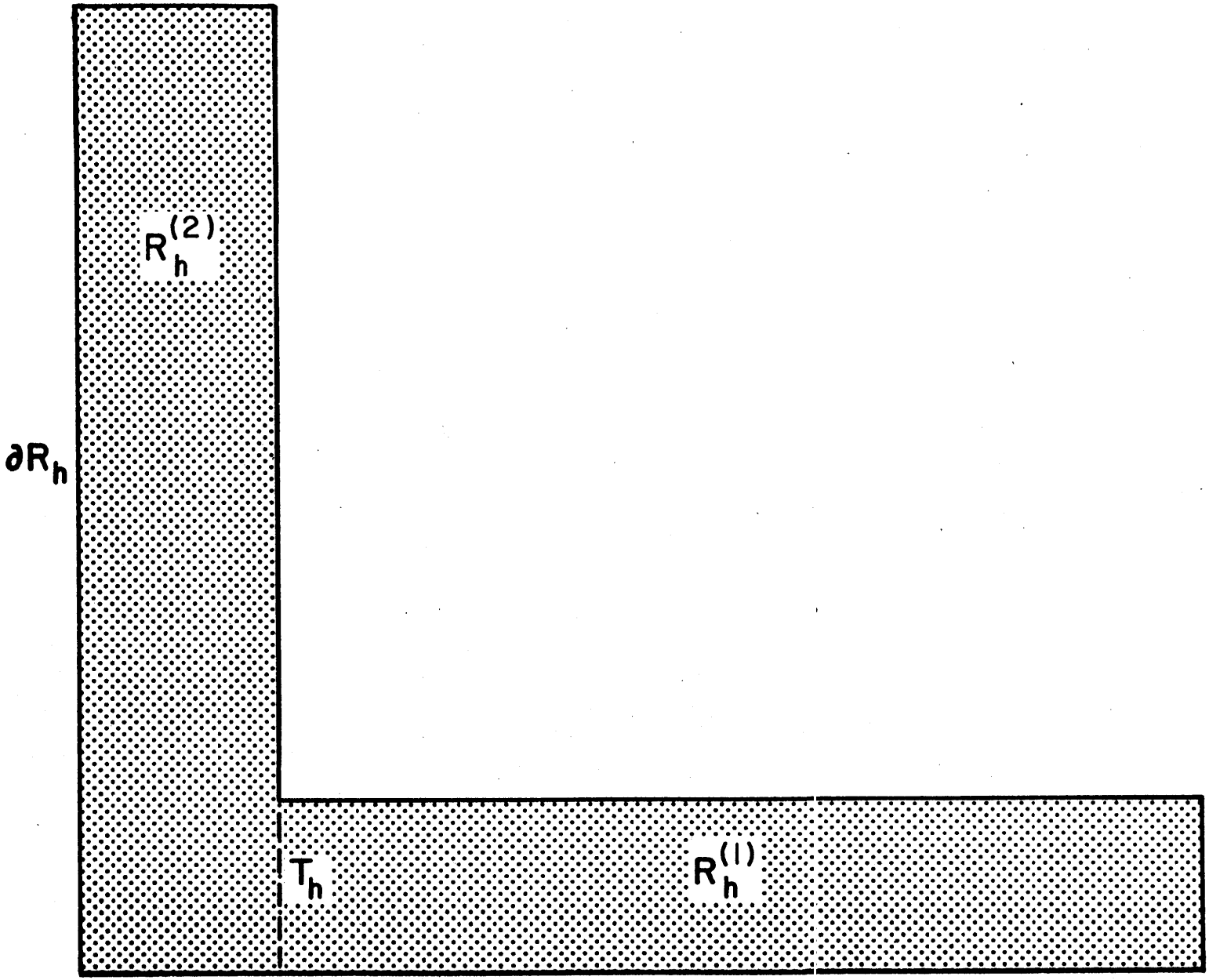

Figure 4. Region in the x-y plane.

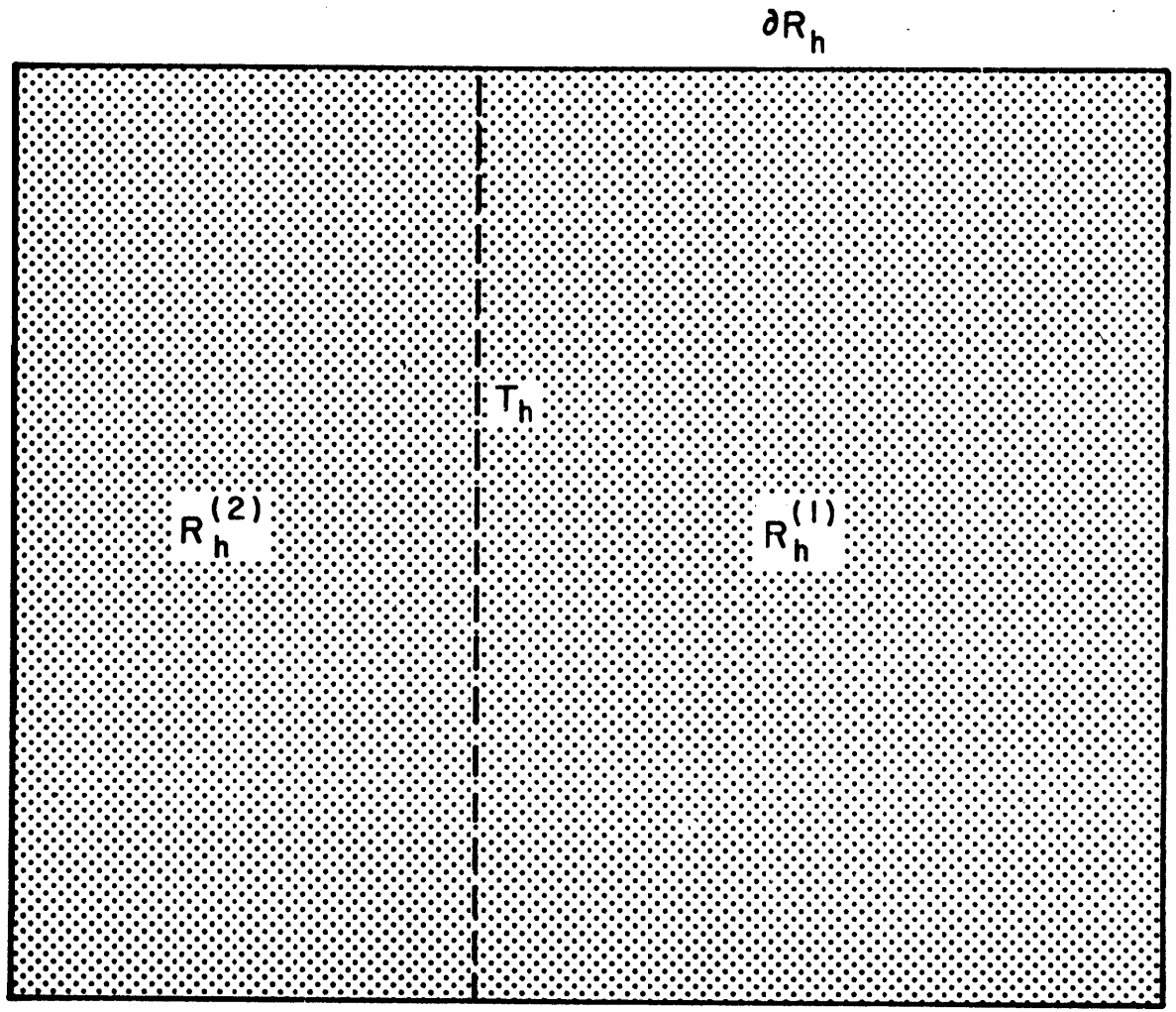

Figure 5. Region in the x-y plane.

References

[1] E. Angel, "Discrete invariant imbedding and elliptic boundary-value problems over irregular regions," J. Math. Anal. Appl., 23 (1968), pp. 471-484.

[2] E. Angel, 'Dynamic programming and linear partial differential equations," J. Math. Anal. Appl., 23 (1968), pp. 628-638.

[3] E. Angel, "A building block technique for elliptic boundary-value problems over irregular regions," J. Math. Anal. Appl., 26 (1969), pp. 75-81.

[4] E. Angel, "Inverse boundary-value problems: elliptic equations," J. Math. Anal. Appl., 30 (1970), pp. 86-98.

[5] E. Angel and R. Kalaba, "A one sweep numerical method for vector-matrix difference equations with two-point boundary conditions," Report 70-16, Department of Electrical Engineering, University of Southern California, Los Angeles, 1970.

[6] O. Buneman, "A compact non-iterative Poisson solver," Report SUIPR-294, Institute for Plasma Research, Stanford University, Stanford, California, 1969.

[7] O. Buneman, "Computer simulation of the satellite photosheath," Report ESRIN-92, European Space Research Institute, Rome, 1970.

[8] B. L. Buzbee, G. H. Golub, and C. W. Nielson, "On direct methods for solving Poisson's equations," SIAM J. Numer. Anal., to appear.

[9] D. C. Collins and E. Angel, "The diagonal decomposition technique applied to the dynamic programming solution of elliptic partial differential equations," J. Math. Anal. Appl., to appear.

[10] B. Daly and B. Nichols, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, personal communication, November 1970.

[11] F. W. Dorr, "The direct solution of the discrete Poisson equation on a rectangle," SIAM Rev., 12 (1970), pp. 248-263.

[12] G. E. Forsythe and C. B. Moler, Computer Solution of Linear Algebraic Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

[13] G. E. Forsythe and W. R. Wasow, Finite-Difference Methods for Partial Differential Equations, Wiley, New York, 1960.

[14] J. A. George, "The use of direct methods for the solution of the discrete Poisson equation on non-rectangular regions," Report STAN-CS-70-159, Computer Science Department, Stanford University, Stanford, California, 1970.

[15] G. H. Golub, "An algorithm for the discrete biharmonic equation," unpublished, 1970.

[16] R. W. Hockney, "The potential calculation and some applications," Methods in Computational Physics, 9 (1970), pp. 135-211.

[17] R. W. Hockney, "POT4 - a fast direct Poisson-solver for the rectangle allowing some mixed boundary conditions-and internal electrodes," to appear.

[18] A. S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell, New York, 1964.

[19] E. Isaacson and H. B. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

[20] R. Kalaba, "A one sweep method for linear difference equations with two point boundary conditions," Report 69-23, Department of Electrical Engineering, University of Southern California, Los Angeles, 1969.

[21] R. L. Morse and L. Rudsinski, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, personal communication, August 1970.

[22] P. J. Roache, "A direct method for the discretized Poisson equation," Report SC-RR-70-579, Sandia Laboratories, Albuquerque, New Mexico, 1970.

[23] R. S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, New-Jersey, 1962.

[24] E. L. Wachspress, Iterative Solution of Elliptic Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.