

MANIPULATING FAST SOLVERS - CHANGING THEIR BOUNDARY CONDITIONS AND PUTTING THEM ON MULTIPLE PROCESSOR COMPUTERS

CHRISTOPHER R. ANDERSON*

Abstract. In this paper we discuss some issues related to the use of fast solvers for Laplace's equation on rectangular domains. We present methods to solve Laplace's equation with non-standard boundary conditions which are built from Laplace solvers with standard boundary conditions. We also present a direct method by which solutions to Laplace's equation on a rectangular domain can be constructed from solutions of Laplace's equation on rectangular subdomains. The majority of the steps in this latter method can be carried out simultaneously and hence the method is suitable for multiple processor machines.

1. Introduction. One often used numerical tool is the subroutine for computing solutions of Laplace's equation on rectangular domains. In this paper we will address two problems related to the use of these fast solvers. The first problem has to do with solving Laplace's equation when the boundary conditions do not correspond to a Dirichlet or Neumann problem. For such problems, standard fast solvers are not directly applicable, but fast solvers can still be used to advantage to solve them. One of the purposes of this paper is to show precisely how, by incorporating existing fast solvers, efficient methods can be created to solve Laplace's equation when the boundary conditions are not the usual type. The second problem has to do with implementing a fast Laplace solver on multiple processor machines. Specifically, if we have a rectangular region which is broken up into rectangular subregions, we consider the problem "Given that a single processor (using an existing fast Laplace solver) can generate a solution of Laplace's equation on a subregion, how do we go about joining solutions on these subregions to form a solution on the whole region?". This problem is one focus of the work in the area of domain decomposition. The design of iterative methods for solving Laplace's equation using multiple processors has been discussed by many researchers. (See the proceedings [6] and [7] for references.) Direct methods - those which require no iteration to join the subdomain solutions together - have not been discussed as much. To our knowledge Chan and Resasco [2], [3] are the first to describe a direct method. The problem which they consider is that of a rectangular domain which has been divided into strips (just vertical or horizontal dividing lines). We will present a similar, but slightly different, method for a rectangular domain decomposed into strips. We will also give a direct method for a rectangular domain which has been decomposed into boxes. One may wonder why apparently different problems are discussed in the same paper. The reason for this is that the two problems can be solved using similar techniques. In particular, the procedures for both problems are based on the use of the same special set of solutions to the discrete Laplace equation.

Our interest in solving Laplace's equation with non-standard boundary conditions comes from the problem of determining the pressure of an incompressible fluid when a non-staggered grid is used [1]. In this problem it is necessary to solve four distinct sets of equations. These equations are described by the five point discrete Laplace operator for interior points with differing equations along the boundaries. Unfortunately, the types of equations which occur at the boundaries cannot be seen as the discretization of any usual boundary conditions, and one is confronted with the problem of having to design solutions procedures for each system. The equations are such that one could design effective multigrid techniques for each system, or go about designing a direct method based on block cyclic

* This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Dept. of Energy, under contract DE-AC03-76SF00098 at the Lawrence Berkeley Laboratory, NSF Grant DMS86-03631, and A.P Sloan Foundation Grant BR-2683.

reduction, however, these methods require quite a bit of time to implement if one is not familiar with the techniques. In an effort to obtain a reasonably efficient procedure which utilizes existing codes as much as possible, we derived a method which is implemented using sine transforms and a routine for computing the solution to a standard Dirichlet problem in the domain.

Similarly, the two essential ingredients of our approach in implementing a fast Laplace solver on a multiple processor computer is a routine for solving Laplace's equation and a routine for computing discrete sine transforms. Thus, the implementation can be carried out using standard routines which are usually available. The particular variety of multiple processor machine we have in mind is one with a moderate number of processors – from two or three to a few hundred processors – not a “massively parallel” machine. While we have not undertaken a detailed analysis of the performance of the method, it is clear that it is best used when there are significant numbers points associated with each processor.

In the first section we discuss a result concerning some special solutions of the discrete Laplace equation which will be used to derive our methods. In the second section we discuss direct and iterative methods which can be used to solve Laplace's equation when non-standard boundary conditions are present. In the third section we discuss the implementation of a fast Laplace solver on a multiple processor machine. Listings of the codes which implement the methods described in this paper are available from the author upon request.

2. Preliminaries. In this section we derive a result about an explicit solution of the discrete Laplace equation which will be needed to solve our problems. Consider a rectangular region Ω described by $0 \leq x \leq a$ and $0 \leq y \leq b$. We assume that the domain is covered by a uniform rectangular grid with m panels in the x-direction and n panels in the y-direction. Let $\delta x = \frac{a}{m}$, $\delta y = \frac{b}{n}$ and denote by Δ^h the discrete five point Laplacian.

PROPOSITION 2.1. *The function*

$$(2.1) \quad S(i, j) = z_k(i) \sin\left(\frac{k\pi j \delta y}{b}\right)$$

where $z_k(i)$ is given by

$$z_k(i) = \lambda_k^i + (\lambda_k^i - \lambda_k^{-i}) \left(\frac{1}{(1 - \lambda_k^{2m})} - 1 \right)$$

and

$$(2.2) \quad \lambda_k = \frac{2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin^2\left(\frac{k\pi \delta y}{2b}\right) - \sqrt{\left(2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin^2\left(\frac{k\pi \delta y}{2b}\right)\right)^2 - 4}}{2}$$

satisfies the equation

$$\Delta^h S = 0 \quad i = 1, \dots, m-1 \quad j = 1, \dots, n-1$$

with the boundary conditions

$$S = \begin{cases} \sin\left(\frac{k\pi j \delta y}{b}\right) & i = 0 & j = 1, \dots, n \\ 0 & i = m & j = 1, \dots, n \\ 0 & j = n & i = 1, \dots, m \\ 0 & j = 0 & i = 1, \dots, m \end{cases}$$

Proof. We first assume that the width of the rectangle is infinite and seek a solution of

$$(2.3) \quad \Delta U^h = 0 \quad i = 1, \dots, \infty \quad j = 1, \dots, n-1$$

with boundary conditions

$$U = \begin{cases} \sin\left(\frac{k\pi j \delta x}{b}\right) & i = 0 & j = 1, \dots, n-1 \\ 0 & i = \infty \\ 0 & j = n & i = 1, \dots, \infty \\ 0 & j = 0 & i = 1, \dots, \infty \end{cases}$$

The method of separation of variables suggests we consider a solution of the form

$$(2.4) \quad U(i, j) = \lambda_k^i \sin\left(\frac{k\pi j \delta y}{b}\right).$$

We substitute this solution into (2.3) to determine λ_k . At the point $i\delta x$ and $j\delta y$ we find

$$\sin\left(\frac{k\pi j \delta y}{b}\right) \lambda_k^i \left(\frac{\cos\left(\frac{k\pi \delta x}{b}\right) - 2 + \cos\left(\frac{k\pi \delta y}{b}\right)}{\delta y^2} \right) + \sin\left(\frac{k\pi j \delta x}{b}\right) \lambda_k^i \left(\frac{\lambda_k^{-1} - 2 + \lambda_k}{\delta x^2} \right) = 0$$

If we cancel the sine term and λ_k^i and simplify using the identity $2 \cos\left(\frac{k\pi \delta y}{b}\right) - 2 = 4 \sin\left(\frac{k\pi \delta y}{2b}\right)^2$, we arrive at a quadratic equation for λ_k ,

$$\lambda_k^2 - \left(2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin\left(\frac{k\pi \delta y}{2b}\right)^2\right) \lambda_k + 1 = 0$$

We use the root of this equation which is less than one in magnitude to get a bounded solution at infinity. Thus,

$$\lambda_k = \frac{2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin\left(\frac{k\pi \delta y}{2b}\right)^2 - \sqrt{\left(2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin\left(\frac{k\pi \delta y}{2b}\right)^2\right)^2 - 4}}{2}.$$

The solution we seek for the finite rectangle is now an infinite sum of solutions of the form (2.4) with λ defined above,

$$\begin{aligned} S(i, j) &= \sin\left(\frac{k\pi j \delta y}{b}\right) \left[\lambda_k^i + \sum_{p=2,3,6,\dots} (\lambda_k^{mp+i} - \lambda_k^{mp-i}) \right] \\ &= \sin\left(\frac{k\pi j \delta y}{b}\right) \left[\lambda_k^i + (\lambda_k^i - \lambda_k^{-i}) \left(\frac{1}{(1 - \lambda_k^{2m})} - 1 \right) \right] \end{aligned}$$

This last expression is derived using the method of images. The simplification is obtained by using the summation formula for a geometric series. \square

PROPOSITION 2.2. *Consider the domain Ω decomposed into two rectangular pieces with a vertical boundary which corresponds to one of the vertical grid lines. We assume that the region on the left of the dividing line has width m_1 grid panels and the region on the right has width m_2 grid panels. The function defined by*

$$(2.5) \quad S(i, j) = z_k(i - m_1) \sin\left(\frac{k\pi j \delta y}{b}\right)$$

where $z_k(i)$ is given by

$$(2.6) \quad z_k(i) = \begin{cases} \lambda_k^{-i} + (\lambda_k^{-i} - \lambda_k^i) \left(\frac{1}{(1 - \lambda_k^{2m_1})} - 1 \right) & i < 0 \\ \lambda_k^i + (\lambda_k^i - \lambda_k^{-i}) \left(\frac{1}{(1 - \lambda_k^{2m_2})} - 1 \right) & i \geq 0 \end{cases}$$

and

$$\lambda_k = \frac{2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin\left(\frac{k\pi\delta y}{2b}\right)^2 - \sqrt{\left(2 + 4\left(\frac{\delta x}{\delta y}\right)^2 \sin\left(\frac{k\pi\delta y}{2b}\right)^2 - 4\right)}}{2}$$

satisfies the equation

$$\Delta^h S = \begin{cases} 0 & i = 1, \dots, m_1 - 1 & j = 1, \dots, n - 1 \\ \left[\frac{z_k(1) - 2 + z_k(-1)}{\delta x^2} + \frac{2 \cos\left(\frac{k\pi\delta y}{b}\right) - 2}{\delta y^2} \right] \sin\left(\frac{k\pi j \delta y}{b}\right) & i = m_1 & j = 1, \dots, n - 1 \\ 0 & i = m_1 + 1, \dots, m & j = 1, \dots, n - 1 \end{cases} \quad (2.7)$$

with the boundary conditions

$$S = \begin{cases} 0 & i = 0 & j = 1, \dots, n \\ 0 & i = m & j = 1, \dots, n \\ 0 & j = n & i = 1, \dots, m \\ 0 & j = 0 & i = 1, \dots, m \end{cases}$$

Proof. This function is obtained by piecing together two solutions of the form (2.1) – one for the left rectangle and one for the right rectangle. The satisfaction of (2.7) follows from the properties of the solutions of the form (2.1). \square

3. Changing Boundary Conditions. In this section we consider the problem of solving a set of finite difference equations associated with a rectangular domain Ω . Ω is described by $0 \leq x \leq a$ and $0 \leq y \leq b$. We assume that it is covered by a uniform rectangular grid with m panels in the x -direction and n panels in the y -direction. Let $\delta x = \frac{a}{m}$, $\delta y = \frac{b}{n}$. At interior points, the equations are those corresponding to a five point Laplace operator. Along the boundaries and in the corners the equations are described by a three point stencil with constant coefficients. We will consider constructing a fast solver for a specific set of equations of this type, and we indicate where the method should be changed to accommodate other sets of equations (ones with different coefficients at the boundary).

The equations which will serve as our model problem consist of

$$(3.1) \quad \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\delta x^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{\delta y^2} = f_{i,j}$$

at interior points - i.e. for $i = 1, \dots, m - 1$ and $j = 1, \dots, n - 1$. For the sides the equations are determined by a three point stencil and have the form,

$$(3.2) \quad \frac{p_{i+1,n} - 2p_{i,n} + p_{i-1,n}}{\delta x^2} + \frac{-p_{i,n} + p_{i,n-1}}{\delta y^2} = f_{i,n}$$

$$(3.3) \quad \frac{p_{i+1,0} - 2p_{i,0} + p_{i-1,0}}{\delta x^2} + \frac{-p_{i,0} + p_{i,1}}{\delta y^2} = f_{i,0}$$

for $i = 1, \dots, m - 1$, and

$$(3.4) \quad \frac{p_{0,j+1} - 2p_{0,j} + p_{0,j-1}}{\delta y^2} + \frac{-p_{0,j} + p_{1,j}}{\delta x^2} = f_{0,j}$$

$$(3.5) \quad \frac{p_{m,j+1} - 2p_{m,j} + p_{m,j-1}}{\delta y^2} + \frac{-p_{m,j} + p_{m-1,j}}{\delta x^2} = f_{m,j}$$

for $j = 1, \dots, n-1$. In the corners we have another three point stencil giving the equations

$$(3.6) \quad \frac{p_{1,0} - p_{0,0}}{\delta x^2} + \frac{p_{0,1} - p_{0,0}}{\delta y^2} = f_{0,0}$$

$$(3.7) \quad \frac{p_{1,n} - p_{0,n}}{\delta x^2} + \frac{p_{0,n-1} - p_{0,n}}{\delta y^2} = f_{0,n}$$

$$(3.8) \quad \frac{p_{m-1,0} - p_{m,0}}{\delta x^2} + \frac{p_{m,1} - p_{m,0}}{\delta y^2} = f_{m,0}$$

$$(3.9) \quad \frac{p_{m-1,n} - p_{m,n}}{\delta x^2} + \frac{p_{m,n-1} - p_{m,n}}{\delta y^2} = f_{m,n}$$

These equations are those corresponding to a discrete Neumann problem in the domain. However, the equations are slightly different than the standard discrete Neumann problem because the boundary equations are obtained by using a first order approximation to the normal derivative at the boundary rather than the usual second order approximation. There is a non-trivial null space corresponding to these equations corresponding to the constant vector - the vector of which each entry is identical and non-zero. We will assume that the right hand side of equations (3.1)-(3.9), $f_{i,j}$, is orthogonal to this vector.

There are numerous fast solvers available for solving the standard Dirichlet problem in a rectangular region, and it is desirable to have a method which takes advantage of these solvers. For this reason, a natural candidate for a solution procedure is the method of matrix partitioning [5], or as it is more recently called, the method of domain decomposition. To apply the method to this problem, the unknowns are split into two groups. The first group, denoted by U_i , is the set of unknowns corresponding to the interior points, and the second group, denoted by U_b , is the unknowns corresponding to the boundary points. With this grouping, the equations (3.1)-(3.9) have the block structure

$$(3.10) \quad \left(\begin{array}{c|c} \text{A} & \text{V} \\ \hline \text{V}^t & \text{B} \end{array} \right) \begin{pmatrix} U_i \\ U_b \end{pmatrix} = \begin{pmatrix} F_i \\ F_b \end{pmatrix}.$$

The reason for this decomposition is that the matrix A corresponds to that of a Dirichlet problem on the interior points and its inverse is easily computed with a fast Dirichlet solver. When the system is written in this form, the solution procedure is carried out in two steps. One first solves

$$(3.11) \quad (\text{B} - \text{V}^t \text{A}^{-1} \text{V}) U_b = F_b - \text{V}^t \text{A}^{-1} F_i$$

to obtain the values of the solution on the boundary, U_b , and then solves the equations

$$(3.12) \quad \text{A} U_i = F_i - \text{V} U_b$$

to obtain the values of the solution at interior points. The equation (3.11) is the Schur complement of B, and is often called the capacitance matrix. Since the solution of (3.12) can be carried out using a fast solver, the primary difficulty in the implementation of this approach is that of solving (3.11). We shall concentrate on both direct and iterative methods for the solution of (3.11).

However, before we discuss the solution procedures for (3.11) we must first address the problems which are introduced by the fact that the equations we are solving are singular. To cope with this difficulty we rely on the following propositions,

PROPOSITION 3.1. *If*

$$\begin{pmatrix} c_i \\ c_b \end{pmatrix}$$

is a null vector for the system (3.10) then c_b is a null vector of (3.11), i.e. if the complete system is singular then the Schur complement equations are also singular.

Proof. If

$$\begin{pmatrix} c_i \\ c_b \end{pmatrix}$$

is a null vector of (3.10) then this implies $A c_i = -V c_b$ or $c_i = -A^{-1}V c_b$. We also have $V^t c_i + B c_b = 0$ so by using the previous expression for c_i we have $(-V^t A^{-1}V + B) c_b = 0$, i.e. c_b is a null vector for (3.11). \square

PROPOSITION 3.2. *If c_b is a null vector for (3.11) then the vector*

$$(3.13) \quad \begin{pmatrix} -A^{-1}V c_b \\ c_b \end{pmatrix}$$

is a null vector for the complete system (3.10).

Proof. This result follows immediately upon substitution of (3.13) into the system of equations (3.10). \square

PROPOSITION 3.3. *Let*

$$\begin{pmatrix} c_i \\ c_b \end{pmatrix}$$

be a null vector of (3.10). If the right hand side of (3.10) is orthogonal to this vector then the right hand side of (3.11) is orthogonal to the vector c_b .

Proof. We are given

$$(c_i, c_b)^t \begin{pmatrix} F_i \\ F_b \end{pmatrix} = 0$$

as well as the fact $A c_i + V c_b = 0$. If we use the transpose of this relation, $c_b^t V^t = -c_i^t A$, then

$$\begin{aligned} c_b^t (F_b - V^t A^{-1} F_i) &= c_b^t F_b + c_i^t A A^{-1} F_i \\ &= c_b^t F_b + c_i^t F_i \\ &= 0 \end{aligned}$$

\square

PROPOSITION 3.4. *Let D be a symmetric negative semi-definite matrix with a one dimensional null space spanned by the vector \vec{n} of unit length. Assume that the vector \vec{f} is orthogonal to \vec{n} . The system of equations $D - \vec{n}\vec{n}^t$ is negative definite, and the solution of*

$$(D - \vec{n}\vec{n}^t)\vec{x} = \vec{f}$$

has the property that

$$D\vec{x} = \vec{f} \quad \text{and} \quad \vec{n}^t \vec{f} = 0$$

Proof. We first show that the system $D - \vec{n}\vec{n}^t$ is negative definite. Consider any vector \vec{y} , we have $\vec{y} = (\vec{y} - (\vec{y}, \vec{n})\vec{n}) + (\vec{y}, \vec{n})\vec{n} = \vec{w} + \vec{z}$, i.e. so \vec{y} is expressed as a component in the null space as well as a component orthogonal to the null space. Now,

$$\begin{aligned} (D\vec{y}, \vec{y}) &= (D\vec{w} + \vec{z}, \vec{w} + \vec{z}) \\ &= (D\vec{w}, \vec{w}) + (D\vec{w}, \vec{z}) + (D\vec{z}, \vec{w}) + (D\vec{z}, \vec{z}) \\ &= (D\vec{w}, \vec{w}) + (D\vec{w}, \vec{z}) \\ &= (D\vec{w}, \vec{w}), \end{aligned}$$

so that

$$\begin{aligned} ((D - \vec{n}\vec{n}^t)\vec{y}, \vec{y}) &= (D\vec{y}, \vec{y}) - ((\vec{n}\vec{n}^t)\vec{y}, \vec{y}) \\ &= (D\vec{w}, \vec{w}) - (\vec{z}, \vec{z}) < 0 \end{aligned}$$

This last inequality follows from the fact that if $\vec{w} \neq 0$ then the right hand side is strictly less than zero. If $\vec{w} = 0$ then we must have $\vec{z} \neq 0$ and hence $(\vec{z}, \vec{z}) < 0$ so the right hand side is again strictly less than zero.

If \vec{x} is a solution of $(D - \vec{n}\vec{n}^t)\vec{x} = \vec{f}$ then by taking the inner product of this equation with \vec{n} and using the fact $(\vec{f}, \vec{n}) = 0$ implies $\vec{n}^t\vec{x} = 0$. Hence $D\vec{x} = \vec{f}$ and $(\vec{x}, \vec{n}) = 0$.

□

These propositions guarantee that the dimension of the null space of (3.11) is less than that of the complete system. Moreover, if the complete system of equation is singular then the system (3.11) will be singular as well. Equations (3.1)-(3.9) have a one dimensional null space and so the null space of (3.11) is also one dimensional. In light of Proposition 3.1 the null vector for (3.11) is a constant vector (of size the number of boundary points). We are assuming the the right hand side of the complete system is orthogonal to the constant vector and so by Proposition 3.3 we have that the right hand side of (3.11) is also orthogonal to the constant vector. By Fredholms theorem the system of equations (3.11) therefore possesses a unique solution.

With the question of the solvability of (3.11) settled we focus on methods to construct the solution. We first consider direct methods for obtaining a solution of the equations. There are two aspects of this approach, the formation of the equations and their solution. The biggest difficulty confronting us is the first aspect, that of forming the equations. Once the equations have been formed, their solution can be obtained using Gaussian elimination (i.e. constructing and using the LU decomposition of the matrix). Of course, one must be careful because the system is singular, but the problems which occur because of the singularity can be avoided by using the result of Proposition 3.4. One works with the matrix corresponding to (3.11) with the term $\vec{n}\vec{n}^t$ subtracted from it. (Here \vec{n} is the null vector of (3.11) which we know a priori.)

The matrix corresponding to the left hand side of equations (3.11), $(B - V^tA^{-1}V)$, is a linear operator from the boundary points to the boundary points. To construct this matrix it is sufficient to consider the action of this operator on a set of basis vectors for the set of boundary points. Our technique will be to use a special set of basis vectors for the boundary points so that the action of the operator in (3.11) can be computed easily (in particular without having to compute A^{-1}). Our choice of basis vectors is motivated by a particular interpretation of the action of this operator, namely, the forward action of the equations represented by $(B - V^tA^{-1}V)$ is the evaluation of the boundary equations on a solution of an interior Dirichlet problem. If we are given some vector U_b of dimension the size of the

boundary points, then the vector $(B - V^t A^{-1} V) U_b$ is that obtained by solving the discrete Dirichlet problem

$$(3.14) \quad \begin{aligned} \Delta^h U &= 0 & \text{in} & \Omega \\ U &= U_b & \text{on} & \partial\Omega \end{aligned}$$

to obtain U and then applying the operator defined by equations (3.2)-(3.9) to this solution. Our basis vectors for the boundary points will therefore be sets of boundary values for which the solution of the discrete Dirichlet problem (3.14) can be computed analytically.

Let us group the boundary points into five sets of unknowns, the first four sets are those points associated with one of the sides of the rectangular region (excluding the corners) and the fifth set corresponds to the corner points. A basis for the complete set of unknowns consists of the union of basis vectors for each subset, when these subset vectors have been extended to be zero at the other boundary points. For the first four subsets we use as our basis the set of vectors consisting of the functions $\sin(\frac{k\pi x}{a})$ for $k = 1, \dots, m-1$ (or $\sin(\frac{k\pi y}{b})$ for $k = 1, \dots, n-1$) evaluated at the grid points along the boundary. For the corner points we use the standard basis. For example, the k th basis vector associated with top side is of size $2m + 2n$ and of the form

$$(3.15) \quad \begin{pmatrix} \vec{s}_k \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where the vector \vec{s}_k are the k th sine function evaluated at the appropriate data points - i.e. for the top basis vector this has the form

$$\vec{s}_k = \begin{pmatrix} \sin(\frac{k\pi h}{a}) \\ \vdots \\ \sin(\frac{k\pi i h}{a}) \\ \vdots \\ \sin(\frac{k\pi(m-1)h}{a}) \end{pmatrix}$$

The basis vectors associated with the corner points are the same size as (3.15), but have a one as one of the last four elements. We shall refer to this basis for the boundary values as a sine basis. The representation of a vector of boundary values in the standard basis can be expressed in the sine basis by performing four discrete sine transforms - one corresponding to each side.

In Proposition 2.1 we gave an explicit solution to a Dirichlet problem with sine data along an edge. These solutions are precisely the solution of a Dirichlet problem with boundary data given by one of the sine basis vectors we have described above. We therefore compute the action of the operator $(B - V^t A^{-1} V)$ on the basis vectors associated with the sides by applying the difference equations (3.2)-(3.9) to these particular solutions. The vector which results is expressed in the standard basis and must be followed by a transformation to express it in the sine basis we are using. The action of the operator on the corner points is easily computed since the corner points are only coupled to other boundary points and no interior points. (The evaluation of $(B - V^t A^{-1} V)$ on the corner points does not involve computing A^{-1} .) We now go into more detail about the explicit form of $(B - V^t A^{-1} V)$ in the sine basis we have chosen.

Let U_1 to U_4 represent vectors associated with the top, bottom, left and right sides respectively, and U_c the vector associated with the corner values, then (3.11) has the form

$$\begin{pmatrix} D_{1,1} & E_{1,2} & E_{1,3} & E_{1,4} & O_{1,5} \\ E_{2,1} & D_{2,2} & E_{2,3} & E_{2,4} & O_{2,5} \\ E_{3,1} & E_{3,2} & D_{3,3} & E_{3,4} & O_{3,5} \\ E_{4,1} & E_{4,2} & E_{4,3} & D_{4,4} & O_{4,5} \\ O_{5,1} & O_{5,2} & O_{5,3} & O_{5,4} & D_{5,5} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_c \end{pmatrix} = \begin{pmatrix} & & & & \\ & & & & \\ & & \text{Tr} & & \\ & & & & \\ & & & & \end{pmatrix} \begin{pmatrix} F_b - V^t A^{-1} F_i \end{pmatrix}. \quad (3.16)$$

Here the matrix Tr is the matrix corresponding to a discrete sine transform of the boundary points, i.e.

$$\text{Tr} = \begin{pmatrix} T_{1,1} & & & & \\ & T_{2,2} & & & \\ & & T_{3,3} & & \\ & & & T_{4,4} & \\ & & & & I \end{pmatrix} \quad (3.17)$$

where each of the diagonal blocks $T_{i,i}$ is a discrete sine transform matrix of $m-1$ or $n-1$ points.

The diagonal blocks of the system $D_{i,i}$, $i = 1, \dots, 4$ are diagonal matrices. For the blocks corresponding to the top and bottom sides, $D_{1,1}$ and $D_{2,2}$, they are identical and the k th element of their diagonal is given by

$$d_k = \left[\frac{\lambda_k + (\lambda_k - \lambda_k^{-1}) \left(\frac{1}{(1-\lambda_k^{2n})} - 1 \right) - 1}{\delta y^2} \right] + \left[\frac{2 \cos\left(\frac{k\pi\delta x}{a}\right) - 2}{\delta x^2} \right]$$

where λ_k is that given by (2.2). The diagonal elements of $D_{3,3}$ and $D_{4,4}$ are similar. The block $D_{5,5}$ is also diagonal with each entry being $\left(-\frac{1}{\delta x^2} - \frac{1}{\delta y^2}\right)$.

The other blocks in the system are not diagonal. The reason for this is that when equations (3.2)-(3.9) are applied to a solution of the form (2.1) associated with a given side, the result on an adjacent side is no longer a described by a single sine function. As an example of these blocks, consider the block $E_{3,1}$. Each column of this matrix is the vector which results from the application of (3.2)-(3.9) along the right side to a special solution of the form (2.1) associated with the top side. In the standard basis this matrix is given by

$$\begin{pmatrix} \sin\left(\frac{\pi(m-1)h}{a}\right)z_1(n-1) & \sin\left(\frac{2\pi(m-1)h}{a}\right)z_2(n-1) & \dots & \sin\left(\frac{(m-1)\pi(m-1)h}{a}\right)z_{m-1}(n-1) \\ \sin\left(\frac{\pi(m-1)h}{a}\right)z_1(n-2) & \sin\left(\frac{2\pi(m-1)h}{a}\right)z_2(n-2) & \dots & \sin\left(\frac{(m-1)\pi(m-1)h}{a}\right)z_{m-1}(n-2) \\ & \vdots & & \vdots \\ \sin\left(\frac{\pi(m-1)h}{a}\right)z_1(1) & \sin\left(\frac{2\pi(m-1)h}{a}\right)z_2(1) & \dots & \sin\left(\frac{(m-1)\pi(m-1)h}{a}\right)z_{m-1}(1) \end{pmatrix}$$

and when multiplied by $T_{3,3}$, the matrix corresponding to a discrete sine transform of the right side values, gives the entries in $E_{3,1}$. The other off diagonal blocks are obtained in an analogous manner.

The blocks denoted by $O_{i,j}$ represent the coupling of the points on the sides with the corner values and vice-versa. The blocks $O_{i,j}$ with $i = 1, \dots, 4$ and $j = 5$ are $4 \times (m-1)$ or $4 \times (n-1)$ blocks and represent the side equations (3.2)-(3.5) applied to the corner basis

vectors. If the corner points are ordered so that the elements of U_c correspond to the top left, bottom right, bottom left, and top right, then for $O_{1,5}$ we get the block

$$\frac{1}{\delta x^2} (T_{1,1}) \begin{pmatrix} 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & 0 \\ 0 & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

where $T_{1,1}$ is the discrete sin transform matrix for the top side values. The other blocks have the same structure. The blocks along the bottom are of size $(m-1) \times 4$ or $(n-1) \times 4$ and represent the equations of the corners when applied to a side basis vector. These blocks are the transpose of the side blocks just described.

If one changes the equations at the boundaries then the entries in the matrix (3.16) will change. However, it is important to realize that whatever the boundary equations are, the entries in the matrix can be computed analytically with the use of the special solutions given by Proposition 2.1. It may be that the diagonal blocks are no longer diagonal matrices, which would be the case if the coefficients in the boundary stencil are not constant coefficients, but the entries are still computable.

This completes our discussion of the construction of the matrix corresponding to $(B - V^t A^{-1} V)$ in the sine basis. As we know from Propositions 3.1 to 3.3 this matrix will be singular with a null vector given by $\vec{w} = \text{Tr} \vec{n}$ where Tr is given by (3.17) and \vec{n} is the null vector of the matrix $(B - V^t A^{-1} V)$ in the standard basis. (\vec{n} is the constant vector with length the number of boundary points.) With the results of Proposition 3.4 in mind, before we compute the LU factorization of this matrix (which we shall use to solve the resulting system of linear equations) we subtract off the term $\vec{w} \vec{w}^t$.

There is one last item to be discussed, and that is the construction of the right hand side of (3.11). This is accomplished by noting that in the standard basis $V^t A^{-1} F_i$ is the vector obtained by solving the Poisson problem

$$\begin{aligned} \Delta^h U &= F_i & \text{in } \Omega \\ U &= 0 & \text{on } \partial\Omega \end{aligned}$$

and then applying the operator defined by (3.2)-(3.9) to the resulting solution U . The vector that results from this process is subtracted from F_b and then transformed into a sine basis to yield the right hand side for (3.11).

In summary the direct solution procedure for equations (3.1)-(3.9) consists of the following steps.

- (i) Construct right hand side of (3.11). This involves solving a discrete Laplace equation in the domain Ω with homogeneous boundary data.
- (ii) Form the matrix $(B - V^t A^{-1} V)$ in the discrete sine basis and then subtract off the term $\vec{w} \vec{w}^t$, where $\vec{w} = \text{Tr} \vec{n}$ and \vec{n} the null vector for the system in the standard basis.
- (iii) Solve for the transform of the boundary values U_b using the LU decomposition of the matrix in (ii). Obtain the actual values by multiplying the resulting vector by the matrix Tr - i.e. perform an inverse sine transform on each of the sides.
- (iv) Solve (3.12) to obtain the values of the solution at the interior points.

In two dimensions the operation cost of the method is $O(N^{\frac{3}{2}})$, where N is the total number of points ($N = nm$) in the domain. However, if one wishes to solve the equations repeatedly, then the LU decomposition of $(B - V^t A^{-1} V)$ can be saved and the work in

computing this decomposition can be avoided. The cost of obtaining a solution then becomes $O(N \log N)$ or $O(N)$, depending on the efficiency of the interior Dirichlet solver used. In three dimensions an analogous direct procedure also has an operation count of $O(N^{\frac{3}{2}})$, but if one requires repeated solutions, the cost per solve becomes only $O(N^{\frac{4}{3}})$. (The LU backsolve is the dominant cost). This operation count is not so favorable, and iterative methods in which a small number of Dirichlet solves of order $O(N \log N)$ or $O(N)$ are needed may be more efficient. We therefore discuss an iterative method for the solution of the equations (3.11).

The iterative method of choice here is preconditioned conjugate-gradients [4]. As our preconditioner we will use the inverse of the of the matrix obtained from (3.16) with the off diagonal blocks $E_{i,j}$ deleted. We review briefly the steps of preconditioned conjugate gradients. If a real symmetric definite system to be solved is of the form

$$Cx = b$$

and M is a symmetric definite approximation to C whose inverse is readily computed, then preconditioned conjugate gradients is derived by applying the standard conjugate gradient method to the prepared system

$$M^{-1}Cx = M^{-1}b$$

while using the inner product defined by $\langle \cdot, \cdot \rangle = (\cdot, M \cdot)$. Here (\cdot, \cdot) is the standard inner product. The system we are solving is real symmetric but only semi-definite, and this can lead to difficulties. However, if the preconditioner is definite (which it is in our problem) and the initial starting vector is chosen to be orthogonal to the null space, then no problems arise. The iterates converge to a solution of the system which is orthogonal to the null space. The steps of the iteration, written using the standard inner product (\cdot, \cdot) are then as follows -

$$r^0 = b - Cx^0 \quad x^0 \text{ initial guess}$$

For $k = 1, 2, 3, \dots$

$$\begin{aligned} z^k &= M^{-1}r^k \\ \beta^k &= \frac{(z^k, r^k)}{(z^{k-1}, r^{k-1})} \quad k \geq 1 \quad \beta^0 = 0 \\ p^k &= z^k + \beta^k p^{k-1} \\ \alpha^k &= \frac{(z^k, z^k)}{(p^k, Cp^k)} \\ x^{k+1} &= x^k + \alpha^k p^k \\ r^{k+1} &= r^k - \alpha^k Ap^k. \end{aligned}$$

In order to implement this iteration scheme for (3.11) we need to be able to evaluate $C = (B - V^t A^{-1} V)$ applied to a vector as well as the preconditioner to this system of equations. If one forms the matrix corresponding to $(B - V^t A^{-1} V)$ and then directly evaluates the action of this matrix on a vector, the resulting operation count will be $O(N^{\frac{3}{2}})$. However, an alternative procedure is to evaluate the action of $(B - V^t A^{-1} V)$ upon a vector by solving a Dirichlet problem in the domain Ω - the problem described above by (3.14). Thus, the action of the matrix on a vector can be computed in $O(N \log N)$ or $O(N)$ operations. With this observation, we find that it is more natural to carry out the iteration of (3.11)

in the standard basis. This makes the evaluation of the preconditioner (which is defined by a matrix in the sine basis) slightly more complicated. In order to apply the preconditioner we take the current iterate (a collection of boundary values) and express them in the sine basis. (This is the same as multiplying the current vector by the matrix T describe above.) We then multiply the transformed vector by the inverse of (3.16) with the the blocks $E_{i,j}$ removed and then transform back to get the resultant vector in the standard basis. The inverse of the preconditioner is computed using the method of matrix partitioning once again. Here we group the unknowns into two sets, corner points and edge points. With this partitioning, the inverse can readily be computed at a cost of inverting a diagonal matrix and solving a dense four by four system of equations. (The formulas used are identical to those in (3.10) - (3.12).) If we ignore the cost of inner products, the cost of each iterate is eight discrete sine transforms (two for each side) and the cost of computing a Dirichlet solution in the interior of the domain.

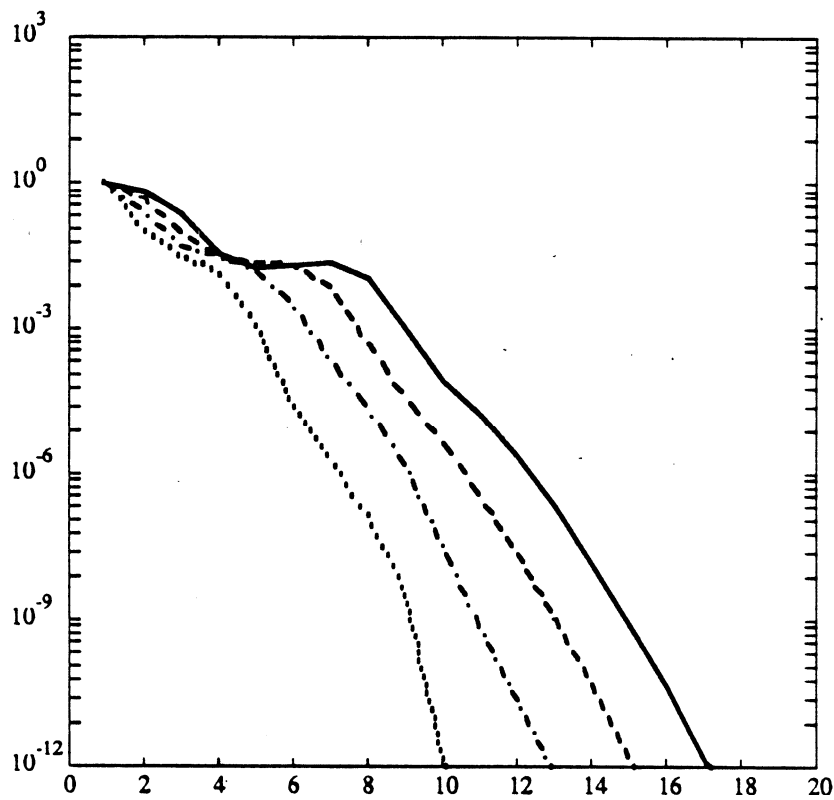


Figure 1

Relative L^2 Error in Solution vs. Number of Iterations.

..... : 9 X 9 grid .-.-.- : 17 X 17 gri
 - - - - : 33 X 33 grid ——— : 65 X 65 gri

To create a test problem we applied the equations (3.1)-(3.9) to the function $f(x, y) = \exp(x) \sin(2y)$ in a unit square and used the result as the right hand side. We then solved the equations using the iterative procedure described. To measure the error we computed an approximate solution to (3.1)-(3.9) in the whole domain by solving (3.12) with boundary values given by the current iterate of the above described procedure. In Figure 1 we plot the relative L^2 norm of the error of this solution verses the number of iterations. The four

different curves represent the results for different mesh sizes - from a 9×9 grid to a 65×65 grid. (We successively doubled the number of panels in each test.) For a given mesh size, the curves indicate that the procedure is rapidly convergent. As the mesh size decreases, the number of iterations required to achieve a given level of accuracy increases somewhat, but only by a very negligible amount (about 2 iterations each time the mesh is refined by a factor of two). This fact demonstrates the efficacy of the preconditioner.

4. Implementation on Parallel Processors. In this section we describe a direct method for solving the discrete Laplace equation on a rectangular domain which consists of combining solutions of Laplace's equation on rectangular subdomains. Each of the subdomain solutions can be carried out simultaneously, and hence the method is suitable for computers with multiple processors.

Our domain is the rectangular region Ω which is described by $0 \leq x \leq a$ and $0 \leq y \leq b$. We assume that a uniform grid covers the domain. There is a division of the top and bottom sides into m panels and the left and right sides into n panels. The mesh widths are then $\delta x = \frac{a}{m}$ and $\delta y = \frac{b}{n}$. The equation to be solved is

$$(4.1) \quad \Delta^h U = F \quad \text{in } \Omega$$

$$(4.2) \quad U = 0 \quad \text{on } \partial\Omega$$

where Δ^h is the discrete five-point Laplace operator. The case of non-homogeneous boundary conditions can be taken care of by a suitable change of the right hand side of (4.1).

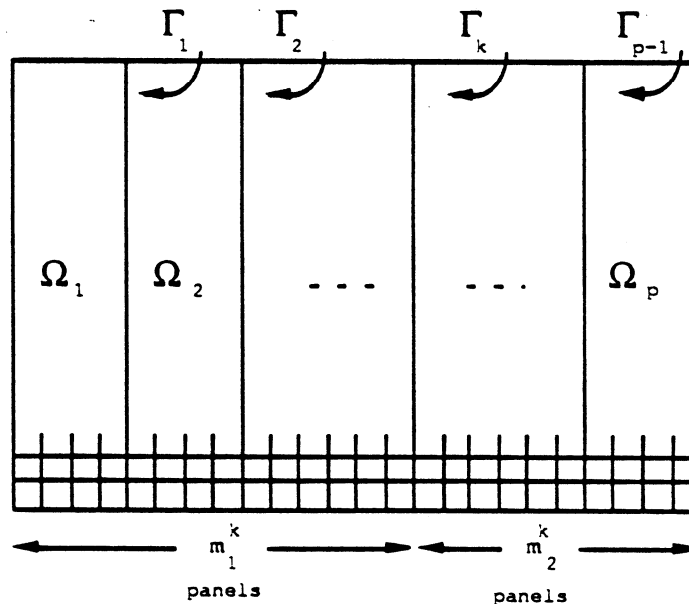


Figure 2

Rectangular domain decomposed into p strips

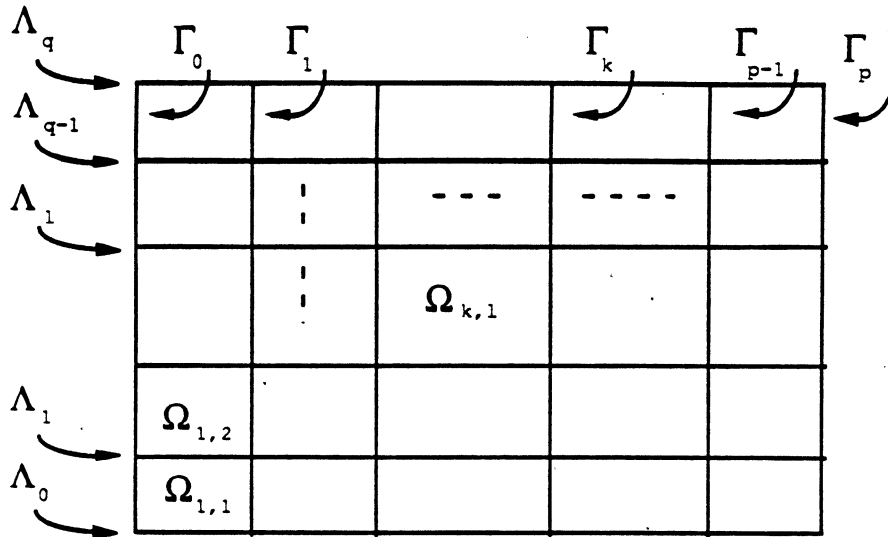


Figure 3

Rectangular domain decomposed into $p \times q$ boxes.

We will describe two different techniques for solving this problem, one corresponds to decomposing the domain into strips (see Figure 1)² and the other corresponds to decomposing the domain into boxes (see Figure 2)³. When solving Laplace's equation on a multiple processor machine, it is natural to think of associating a processor with each subdomain. With this objective in mind, we shall describe a method which combines solutions of the discrete Laplacian on each subdomain (which can be computed by the individual processors) so that the result is a solution on the whole domain. The method we present for doing this is direct in the sense that it requires just two or three solutions of Laplace's equation on each subdomain and a few discrete sine transforms along the interfaces between the domains. There is no iteration performed.

In the method for the strips we assume that the domain is broken up into p pieces as in Figure 1². We denote the interfaces between adjacent regions by Γ_k . There is no need for the widths of the regions to be equal - although from a parallel processing viewpoint, this is probably the most efficient one. The first step of the method consists of solving a discrete Laplace equation on each subdomain assuming homogeneous boundary data along the interfaces. For $k = 1, \dots, p$ we compute \tilde{U}_k , the solution to

$$(4.3) \quad \begin{aligned} \Delta^h \tilde{U}_k &= F_k & \text{in} & \Omega_k \\ \tilde{U}_k &= 0 & \text{on} & \partial\Omega_k \end{aligned}$$

where F_k is the restriction of the right hand side to Ω_k . We denote the solution which is the union of these solutions by \tilde{U} , i.e.

$$(4.4) \quad \tilde{U} = \begin{cases} \tilde{U}_k & \text{in } \Omega_k \\ 0 & \text{on } \Gamma_k \end{cases}$$

The function \tilde{U} satisfies (4.1)-(4.2) except at points along the interior interfaces Γ_k . We now add to \tilde{U} a collection of correction functions (one associated with each interface) in order to construct the complete solution -

$$(4.5) \quad U(i, j) = \tilde{U}(i, j) + \sum_{k=1}^{p-1} C_k(i, j)$$

For our C_k we will use a sum of the special functions (2.5) which are described by Proposition 2.2 in the preliminaries. Specifically, if Γ_k is located m_1^k panels from the left edge and m_2^k panels from the right edge (see Figure 2) then we choose

$$(4.6) \quad C_k(i, j) = \sum_{\kappa=1}^{n-1} c_{\kappa}^k z_{\kappa}(i - m_1^k) \sin\left(\frac{\kappa\pi j \delta y}{2b}\right)$$

where z_{κ} is a function of the form given by (2.6). Now what remains to be determined are the coefficients c_{κ}^k in each of these correction functions. Fortunately, and this is why these functions were selected, the functions C_k are discrete harmonic at all points in the domain except along the interface Γ_k . Thus, when we consider the result of applying the discrete Laplace operator to (4.5),

$$\Delta^h U = F = \Delta^h \tilde{U} + \Delta^h \left(\sum_{k=1}^{p-1} C_k(i, j) \right),$$

we find that this expression is zero at all points of the domain except along the interfaces. Along each interface we get the equation

$$(4.7) \quad \Delta^h C_k = F - \Delta^h \tilde{U} = \tilde{F}^k \quad (i, j) \in \Gamma_k.$$

Thus we get $p - 1$ sets of equations for the $p - 1$ sets of coefficients c_{κ}^k . In view of the structure of C_k we observe that the equations for the coefficients c_{κ}^k are diagonalized by the discrete sine transform. If we denote by S the matrix corresponding to the discrete sine transform of $n - 1$ points, then we have

$$(4.8) \quad H \tilde{c}^k = S(\tilde{F}^k)$$

where H is the diagonal matrix with entries

$$h(\kappa, \kappa) = \frac{z_{\kappa}(1) - 2 + z_{\kappa}(-1)}{\delta x^2} + \frac{2 \cos\left(\frac{\kappa\pi \delta y}{b}\right) - 2}{\delta y^2}.$$

We can therefore determine all of the sets of coefficients of the correction functions by performing a discrete sine transform of the appropriate data along Γ_k and then multiplying the result by a diagonal matrix (the inverse of H).

This process allows one to compute the representation of the complete solution in the form (4.5). It now remains to evaluate this solution at all of the grid points of the domain. If the formula (4.5) is used directly, then the cost of evaluation is $O(mn^2)$ per subdomain. A more efficient way to evaluate the solution is to just evaluate (4.5) at the boundaries and then fill in the solution by solving Laplace's equation with this boundary data. On each subdomain we solve for

$$(4.9) \quad \begin{aligned} \Delta^h U_k &= F_k \quad \text{in } \Omega_k \\ U_k &= \sum_{k=1}^{p-1} C_k(i, j) \quad \text{on } \partial\Omega_k \end{aligned}$$

In the formation of the boundary values it is expedient to gather together all of the sine coefficients first and then do one sine transform to obtain the boundary values - i.e. we have

$$\begin{aligned} \sum_{k=1}^{p-1} C_k(i, j) &= \sum_{k=1}^{p-1} \sum_{\kappa=1}^{n-1} c_{\kappa}^k z_{\kappa}(i - m_1^k) \sin\left(\frac{\kappa\pi j \delta y}{2b}\right) \\ &= \sum_{\kappa=1}^{n-1} \left(\sum_{k=1}^{p-1} c_{\kappa}^k z_{\kappa}(i - m_1^k) \right) \sin\left(\frac{\kappa\pi j \delta y}{2b}\right) \end{aligned}$$

so it is useful to form the inner sum of this last expression first.

We now summarize the steps in the method:

- (i) Solve a discrete Laplace equation with homogeneous data on each subdomain Ω_k to obtain \tilde{U} .
- (ii) Evaluate the sine transform of the right hand side of (4.7) and multiply the result by the inverse of H in (4.8) to obtain the coefficients of the correction functions c_κ^k .
- (iii) Evaluate the correction functions for each interface at every other interface to determine the boundary values of the solution along the interfaces.
- (iv) Solve a discrete Laplace equation (4.9) in each subdomain to construct the solution in the subdomain.

Each of the steps described above can be carried out simultaneously, and so the method is suitable for multiple processor computer.

To obtain a method for a domain decomposed into rectangles (see Figure 3) one can apply the method for strips recursively. In this form, one uses the method described above to combine solutions on vertical (or horizontal) strips. When it is necessary to form the solution on the individual strips, the strips are further subdivided into boxes and solutions on these boxes are combined to form a solution on the strip. This application leads to a direct solution procedure which requires four solutions of Laplace's equation on the boxes which make up the domain. A slightly different approach provides a method by which one can obtain the complete solution to (4.1)-(4.2) at a cost of only three subdomain solves.

We assume that the region is divided up by $p - 1$ internal vertical boundaries (denoted by Γ_k) and $q - 1$ internal horizontal boundaries (denoted by Λ_l). In this decomposition there are pq rectangular subdomains which we label by $\Omega_{k,l}$. We label the left and right edges by Γ_0 and Γ_p respectively and the top and bottom edges by Λ_0 and Λ_q . With such a labelling, then $\Omega_{k,l}$ is that domain which is bordered by Γ_{k-1} on the left and Γ_k on the right and Λ_{l-1} on the bottom and Λ_l on the top.

The first step of our method is to construct the solution of a discrete Laplace equation with homogeneous boundary data on each subdomain. We compute $\tilde{U}_{k,l}$, the solution to

$$\begin{aligned} \Delta^h \tilde{U}_{k,l} &= F_{k,l} & \text{in } \Omega_{k,l} \\ \tilde{U}_{k,l} &= 0 & \text{on } \partial\Omega_{k,l} \end{aligned}$$

where $F_{k,l}$ is the restriction of the right hand side to $\Omega_{k,l}$. We then form the union of these solutions,

$$(4.10) \quad \tilde{U} = \begin{cases} \tilde{U}_{k,l} & \text{in } \Omega_{k,l} \\ 0 & \text{on } (\cup_{k=1}^p \Gamma_k) \cup (\cup_{l=1}^q \Lambda_l) \end{cases}$$

To this solution we add a collection of corrections functions which are associated with both vertical and horizontal dividing lines. We seek a solution on the whole domain of the form

$$(4.11) \quad U(i, j) = \tilde{U}(i, j) + \sum_{k=1}^{p-1} C_k(i, j) + \sum_{l=1}^{q-1} D_l(i, j)$$

where the functions C_k are as described by (4.6) and D_l are similarly defined functions with respect to the interfaces Λ_l ,

$$D_l(i, j) = \sum_{\kappa=1}^{m-1} d_\kappa^l z_\kappa(j - n_1^l) \sin\left(\frac{\kappa\pi i \delta x}{2a}\right)$$

if n_1^l and n_2^l are the number of panels that the vertical interface Λ_l is from the bottom and top respectively.

When we consider the application of the discrete Laplacian to the solution of the form (4.11) we obtain equations along the interfaces Γ_k and Λ_l for the coefficients in the correction functions C_k and D_l . Specifically, we get the following equation along each Γ_k ,

$$(4.12) \quad \Delta^h C_k = \begin{cases} F - \Delta^h \tilde{U} & \text{for } (i, j) \in \Gamma_k / (\Gamma_k \cap_{l=1}^{q-1} \Lambda_l) \\ \frac{(F - \Delta^h \tilde{U})}{2} & \text{for } (i, j) \in \Gamma_k \cap_{l=1}^{q-1} \Lambda_l \end{cases}$$

and along Λ_l we get the equation

$$(4.13) \quad \Delta^h D_l = \begin{cases} F - \Delta^h \tilde{U} & \text{for } (i, j) \in \Lambda_l / (\Lambda_l \cap_{k=1}^{p-1} \Gamma_k) \\ \frac{(F - \Delta^h \tilde{U})}{2} & \text{for } (i, j) \in \Lambda_l \cap_{k=1}^{p-1} \Gamma_k. \end{cases}$$

These equations are the same as those for the strip case except that at the points where the interfaces intersect (the interior corner points) the right hand side of the equations is weighted by $\frac{1}{2}$. This weighting accounts for the fact that the Laplacian of both types of correction functions C_k and D_l contribute at those points.

The equations (4.12)-(4.13), being similar to the strip case, are diagonalized by the discrete sine transform and thus their solution is efficiently accomplished. This allows one to determine the coefficients in the expansion of the solution (4.11).

At this point the complete solution is obtained in the form (4.11). The cost to obtain this representation is one solution of a Laplace equation on each subdomain and two sine transforms along each horizontal and vertical interface. It is the evaluation of the solution given by (4.11) at the interior points which introduces difficulties. One does not want to evaluate the sums in (4.11) at all interior points in the domain directly - this is computationally very expensive. Our method will be to use the values of the solution (4.11) along the edges in combination with two solutions of Laplace's equation in each subdomain to form the interior values of the solution.

We assume that \tilde{U} in (4.10) has been saved and what remains to be determined is the contribution to the solution from the correction functions. We first evaluate the function

$$(4.14) \quad \sum_{k=1}^{p-1} C_k(i, j)$$

along all of the vertical interfaces Γ_k . The function (4.14) is then the harmonic function in the vertical strips Ω_k with these boundary values. Similarly the function defined by

$$(4.15) \quad \sum_{l=1}^{q-1} D_l(i, j)$$

is harmonic in the horizontal strips divided by Λ_l with boundary values given by (4.15) along these boundaries. We can now evaluate these correction functions at interior points by a solving a homogeneous Laplace equation in the horizontal strips (to obtain (4.14)) and the vertical strips (to obtain (4.15)). To accomplish this we employ the strip technique described above to construct the solution from solutions on each of the domains $\Omega_{k,l}$. If these vertical and horizontal solutions are obtained separately then the cost becomes that of four subdomain solutions for each $\Omega_{k,l}$. (Two solutions are necessary for the horizontal strips and two solutions necessary for the vertical strips.) However, we can combine these problems and obtain a solution on both the horizontal and vertical strips at a cost of computing two solutions on the subdomains.

When the strip method is applied to finding the solution in either the horizontal or vertical strips, one begins by constructing a solution to Laplace's equation in each of the subdomains $\Omega_{k,l}$ with homogeneous boundary data. In our particular instance, the right hand side for the subdomain problem consists of the incorporation of the boundary values of (4.14) along Γ_k and the boundary values of (4.15) along Λ_l . When employing the strip method to determine the values of (4.14) in the vertical strips, we use functions of the form $\tilde{W}_{k,l}$ and they are determined as the solution of

$$\begin{aligned} \Delta^h \tilde{W}_{k,l} &= 0 && \text{in } \Omega_{k,l} \\ \tilde{W}_{k,l} &= \begin{cases} \sum_{k=1}^{p-1} C_k(i, j) & \text{on } \partial\Omega_{k,l} \cap (\Gamma_{k-1} \cup \Gamma_k) \\ 0 & \text{on } \Omega_{k,l} \cap (\Lambda_{l-1} \cup \Lambda_l) \end{cases} \end{aligned}$$

To determine the values of (4.15) in the horizontal strips, we use functions of the form $\tilde{Z}_{k,l}$ and they are determined as the solution of

$$\begin{aligned} \Delta^h \tilde{Z}_{k,l} &= 0 && \text{in } \Omega_{k,l} \\ \tilde{Z}_{k,l} &= \begin{cases} 0 & \text{on } \Omega_{k,l} \cap (\Gamma_{k-1} \cup \Gamma_k) \\ \sum_{l=1}^{q-1} D_l(i, j) & \text{on } \partial\Omega_{k,l} \cap (\Lambda_{l-1} \cup \Lambda_l) \end{cases} \end{aligned}$$

The subdomain solutions are then combined to form $\tilde{W} = \cup \tilde{W}_{k,l}$ and $\tilde{Z} = \cup \tilde{Z}_{k,l}$. These solutions in turn are used in the computation of the right hand side for the correction equations of the form (4.7) along the subinterfaces. (By subinterfaces we mean the vertical dividing lines of the horizontal strips and the horizontal dividing lines of the vertical strips.) This computation consists of evaluating the discrete Laplacian of these functions i.e. computing $\Delta^h \tilde{W}$ along the horizontal subinterfaces and $\Delta^h \tilde{Z}$ along the vertical subinterfaces. Consider the solution to

$$(4.16) \quad \begin{aligned} \Delta^h \tilde{V}_{k,l} &= 0 && \text{in } \Omega_{k,l} \\ \tilde{V}_{k,l} &= \begin{cases} \sum_{k=1}^{p-1} C_k(i, j) & \text{on } \partial\Omega_{k,l} \cap (\Gamma_{k-1} \cup \Gamma_k) \\ \sum_{l=1}^{q-1} D_l(i, j) & \text{on } \partial\Omega_{k,l} \cap (\Lambda_{l-1} \cup \Lambda_l) \end{cases} \end{aligned}$$

and designate the union of these solutions over the whole domain by \tilde{V} . Solutions of Laplace's equation are linear in the boundary data, so $\tilde{V} = \tilde{W} + \tilde{Z}$. Furthermore, along the l th horizontal subinterface between Γ_{k-1} and Γ_k we have that

$$(4.17) \quad \Delta^h \tilde{W}_k = \Delta^h \tilde{V} - \Delta^h \tilde{Z}.$$

It is important to observe that the term $\Delta^h \tilde{Z}$ can be evaluated with the aid of a discrete sine transform. Specifically, this term is the discrete Laplacian applied to the union of the two solutions

$$(4.18) \quad \begin{aligned} \Delta^h \tilde{Z}_{k,l} &= 0 && \text{in } \Omega_{k,l} \\ \tilde{Z}_{k,l} &= \begin{cases} \sum_{l=1}^{q-1} D_l(i, j) & \text{on } \partial\Omega_{k,l} \cap (\Lambda_{l-1} \cup \Lambda_l) \\ 0 & \text{on } \Omega_{k,l} \cap (\Gamma_{k-1} \cup \Gamma_k) \end{cases} \end{aligned} \quad (4.19)$$

and

$$(4.20) \quad \Delta^h \tilde{Z}_{k,l+1} = 0 \quad \text{in} \quad \Omega_{k,l+1}$$

$$(4.21) \quad \tilde{Z}_{k,l+1} = \begin{cases} \sum_{l=1}^{q-1} D_l(i, j) & \text{on} \quad \partial\Omega_{k,l+1} \cap (\Lambda_l \cup \Lambda_l) \\ 0 & \text{on} \quad \Omega_{k,l+1} \cap (\Gamma_{k-1} \cup \Gamma_k) \end{cases}.$$

If we designate the nonzero data along the $l-1$, l , and $l+1$ subinterfaces in (4.19) and (4.21) by d_{l-1} , d_l , and d_{l+1} , and their sine transform by \hat{d}_{l-1} , \hat{d}_l , and \hat{d}_{l+1} , then by using the result of Proposition 3.1 we find that the transform of $\Delta^h \tilde{Z}$ along the l th subinterface between Γ_{k-1} and Γ_k is given by

$$\widehat{\Delta^h \tilde{Z}}(\kappa) = d_{l-1}(\kappa) z_\kappa^l(n' - 1) + d_l(\kappa) \left(\frac{z_\kappa(1) - 2 + z_\kappa(-1)}{\delta y^2} + \frac{2 \cos(\frac{\kappa \pi \delta y}{b}) - 2}{\delta x^2} \right) + d_{l+1} z_\kappa^{l+1}(n' - 1)$$

(4.22)

where κ is the wave number and z_κ^l and z_κ^{l+1} are the functions corresponding to the solutions given by Proposition 3.1 when the domain is $\Omega_{k,l}$ and $\Omega_{k,l+1}$ respectively. (We are assuming the subdomains have a height of n' panels. We find the values of $\Delta^h \tilde{Z}$ along this subinterface by taking the inverse transform of (4.22). In this way we can form each of the terms in (4.17) and hence the right hand sides for the equations on the horizontal subinterfaces which determine the correction functions used in evaluating (4.14) in a vertical strip. In a similar fashion one determines the right hand sides on the vertical subinterfaces used in the process of determining (4.15) on the horizontal strips.

With the right hand sides of the equations on the subinterfaces formed, the correction functions associated with these subinterfaces are determined by solving the analog of (4.8). Using these correction functions the values of (4.14) along horizontal subinterfaces can be computed and as well as the values of (4.15) along vertical subinterfaces.

The values of (4.14) and (4.15) are now known at all points of both Γ_k and Λ_l . Since both of these functions are harmonic at interior points of $\Omega_{k,l}$, we can evaluate these functions in the interior by performing one more subdomain solve on $\Omega_{k,l}$ using the sum of their boundary values as boundary data. This solution is then added to \tilde{U} and the complete solution of the problem is obtained.

The complete technique is summarized as follows,

- (i) Solve a discrete Laplace equation with homogeneous data on each subdomain $\Omega_{k,l}$.
- (ii) Evaluate the sine transform of the right hand side of (4.12)-(4.13) along the vertical and horizontal dividing lines Γ_k and Λ_l . Determine the coefficients of the correction functions associated with these interfaces (using the analog of (4.7)-(4.8)).
- (iii) Evaluate the correction functions associated with each interface at every other interface.
- (iv) Solve (4.16), a discrete Laplace equation in each subdomain using these values as boundary data.
- (v) Compute the right hand sides of the subinterface equations by combining differences of the solution defined by step (iv) and a correction which is obtained using formulas similar to (4.22).

- (vi) Solve the subinterface equations using discrete sine transforms and thus determine the values of the horizontal corrections along the vertical interfaces and the vertical corrections along the horizontal interfaces.
- (vii) Solve a homogeneous Laplace equation with boundary values given by the sum of the correction functions (4.12)-(4.13) to obtain the values of the correction at all interior points. Add this result to the function defined by (4.10) to obtain the complete solution.

As is the case with the strip method, most of the work in each step can be carried out simultaneously.

REFERENCES

- [1] Anderson, C.R., "Derivation and Solution of the Discrete Pressure Equations for the Incompressible Navier-Stokes Equations", Report CAM 88-36, UCLA Department of Mathematics, 1988.
- [2] Chan, T. F., and Resasco, D.C., Hypercube Multiprocessors 1987, Proceedings of the 2nd Conference on Hypercube Multiprocessors, Knoxville, August 1986, M. Heath ed., SIAM Publications, Philadelphia, pp. 738-746, 1987.
- [3] Chan, T. F., and Resasco, D.C., Proceedings of the 1st International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, January 1987, R. Glowinski, et. al. eds., Siam Publications, Philadelphia., pp. 217-230., 1988.
- [4] Concus, P., Golub, G.H., and O'Leary, D.P., Sparse Matrix Computations, J.R. Bunch and D.J. Rose, eds., Academic Press, pp. 309-332, 1976.
- [5] Lanczos, C., Applied Analysis, Dover Publications, pg. 141-143, 1988.
- [6] Proceedings of the 1st International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, January 1987, R. Glowinski, et. al. eds., Siam Publications, Philadelphia, 1988.
- [7] Proceedings of the 2nd International Symposium on Domain Decomposition Methods, Los Angeles, January 1988, T. Chan, et. al. eds., Siam Publications, Philadelphia, 1988.