Domain Decomposition Techniques and the Solution of *Poisson's Equation in Infinite Domains**

Christopher R. Anderson†

Abstract. We discuss how domain decomposition ideas can be used to construct efficient methods for solving Poisson's equation in domains of infinite extent. We give the details for the construction of methods to solve Poisson's equation in the region external to a cylinder and for an infinite backstep. Computational results are presented.

1. Introduction. In some recent work on the numerical solution of incompressible fluid motion in two dimensions [1] it was necessary to construct a solver which would compute the solution to Poisson's equation in a domain of infinite extent. The particular problem was that associated with flow around a circular cylinder. The problem domain was the infinite region external to the cylinder and the computational domain, the region in which the flow quantities were being tabulated, was an annulus about the cylinder. The outer radius of this annulus was not sufficiently far so that setting the value of the solution of Poisson's equation or it s normal derivative equal to zero on this boundary was an acceptable boundary condition for a finite difference solution in the annulus. The purpose of these proceedings is to discuss how domain decomposition techniques can be used to construct a solver for this particular infinite domain problem. The basic idea is to treat the annulus and the infinite component exterior to the annulus as two domains in a domain decomposition procedure. We shall discuss the ideas behind domain decomposition in such a way that the method of construction of a solver is straight forward. The resulting algorithm which we have obtained is not really "new", for example, similar results could be obtained employing the work of Bramble and Pasciak [3], Hariharan [5], or Kang and Dehao [6] and presumably many others. However, the implementation we give here is rather easy to carry out and is efficient since it is just a combination of a fast Fourier transform routine (to solve the interface problem) and a fast Poisson solver (for the solution in the interior of the annulus). The approach which is presented here can also be applied to other infinite domains. In order to demonstrate this, we discuss the implementation of a Poisson solver when the domain is an infinite backstep - a domain which is often used in

*Research Supported by ONR Contracts #N00014-86-K-0691, #N00014-86-K-0727 and NSF Grant DM586-57663

†Department of Mathematics, UCLA, Los Angeles, California, 90024

testing of two-dimensional incompressible fluids codes. To our knowledge the method has not been presented before. Whereas, the procedure for the region external to a cylinder is a direct method, the procedure for the backstep uses the conjugate gradient method to solve the equations which arise in the domain decomposition implementation. This iterative technique is certainly well known to those who apply domain decomposition techniques to bounded domains, but has not necessarily been used by those who are in the business of implementing "infinite" boundary conditions - which is what we are essentially doing here. Thus, for those who are familiar with domain decomposition techniques what follows is a description of how those techniques can be extended to problems in which the domain may be of infinite extent. For those who are familiar with solution procedures on infinite domains, the following shows how domain decomposition ideas can be used to construct efficient implementations of infinite domain boundary conditions.

2. Basic Strategy. In this section we discuss the ideas of domain decomposition in such a way that the application to the problem of calculating the solution to Poisson's equation in an infinite domain is clearly revealed.

Consider the problem associated with finding the solution of Poisson's equation for a rectangle. The problem to be solved is

$$\begin{aligned} \Delta u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

The basic strategy of the domain decomposition procedure is to decompose the region into two pieces Ω_1 and Ω_2 and construct a solution to (1) by taking the union of solutions to (1) on sub-domains. If the domain is split up into two rectangular pieces then the sub-domain problems are specified by

$$\begin{aligned} \Delta u_i &= f_i \quad \text{in } \Omega_i \\ u_i &= g \quad \text{on } \partial\Omega_i \backslash \Gamma \\ u_i &= u_\Gamma \quad \text{on } \partial\Gamma \end{aligned} \tag{2}$$

for $i = 1, 2$ and where Γ is the interface between the two regions. (See Figure 1) The difficulty in implementing this technique is the determination of the boundary values u_Γ. What is done is to find equations which the u_Γ satisfy and then solve these equations. With the u_Γ determined, the complete solution over the domain is then obtained by just solving (2) for u_1 and u_2.

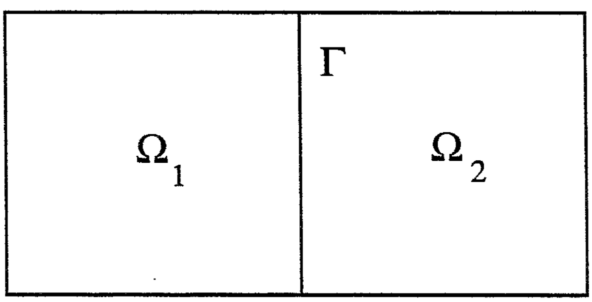

Figure 1

The equations which determine u_Γ are precisely the equations which ensure that if they are solved, and these values are used in the two boundary value problems (2), then the resulting solutions combine to form a solution on the whole domain. Simply; the equations you solve are those which will guarantee that the domain decomposition procedure works.

We now formally derive the equations which the boundary values u_Γ satisfy. In order for the two solutions of (2) , u_1 and u_2, to combine to form a solution of the problem on the whole domain they must be continuous and their normal derivatives must also be continuous across the interface Γ. Under sufficient regularity assumptions this guarantees that the combined solution forms a weak solution of the equations. Under further regularity assumptions this also guarantees that u_1 and u_2 combine to form a strong solution of the equations. If we take the normal derivative along Γ to be outward, then these two conditions can be expressed as

$$u_1 = u_2 \quad \text{on} \quad \Gamma \tag{3}$$

$$\frac{\partial u_1}{\partial n} = -\frac{\partial u_2}{\partial n} \quad \text{on} \quad \Gamma. \tag{4}$$

We are assuming that $u_1 = u_2 = u_\Gamma$ along Γ so relation (3) is satisfied. It is the second relation, the so called "transmission" or "flux" boundary conditions, which determine the equations for u_Γ.

To understand the manner in which the relation (4) determines u_Γ we express it as

$$\frac{\partial u_1}{\partial n}[u_b, f_1, g_1] + \frac{\partial u_2}{\partial n}[u_b, f_2, g_2] = 0. \tag{5}$$

This form reveals the dependence of these normal derivatives on the data in the sub-domains. Here $g_1 = g \quad \text{on} \quad \partial\Omega_1\backslash\Gamma$ and $g_2 = g \quad \text{on} \quad \partial\Omega_2\backslash\Gamma$. Since the problem is linear, we can separate out the contribution to the normal derivatives in (5) from u_Γ and that from the other data. We are lead to the following equation,

$$\frac{\partial \bar{u}_1}{\partial n}[u_\Gamma] + \frac{\partial \bar{u}_2}{\partial n}[u_\Gamma] = -(\frac{\partial \tilde{u}_1}{\partial n}[f_1, g_1] + \frac{\partial \tilde{u}_2}{\partial n}[f_2, g_2]) \tag{6}$$

with $u_1 = \bar{u}_1 + \tilde{u}_1$ and $u_2 = \bar{u}_2 + \tilde{u}_2$. The equations which determine u_Γ are normal derivative (or flux) balance equations – equation (6) expresses the fact that the flux jump of $\bar{u}_1$ and $\bar{u}_2$ induced by the values u_Γ must be equal to the flux jump in $\tilde{u}_1$ and $\tilde{u}_2$ which is induced by the other boundary values and the forcing function.

The terms in (6) which determine u_Γ can be evaluated operationally. Specifically, given u_Γ to evaluate $\frac{\partial \bar{u}_1}{\partial n}[u_\Gamma]$, one first solves

$$\begin{aligned} \Delta\bar{u}_1 &= 0 \quad \text{in} \quad \Omega_1 \\ \bar{u}_1 &= 0 \quad \text{on} \quad \partial\Omega_1\backslash\Gamma \\ \bar{u}_1 &= u_\Gamma \quad \text{on} \quad \partial\Gamma \end{aligned}$$

and then takes the normal derivative of this function. Similarly, to evaluate the first term on the right hand side of the equation (6), we first solve a Poisson problem of the form

$$\begin{aligned} \Delta\tilde{u}_1 &= f_1 \quad \text{in} \quad \Omega_1 \\ \tilde{u}_1 &= g_1 \quad \text{on} \quad \partial\Omega_1\backslash\Gamma \\ \tilde{u}_1 &= 0 \quad \text{on} \quad \partial\Gamma \end{aligned}$$

and evaluates the normal derivative of this solution. It is worth noting that in this process of evaluating these quantities, we have not depended on any specific discretization of the equations. It is this fact which allows one to have one or both of the domains of infinite extent - all that is needed is some technique for evaluating the terms of (6).

The basic strategy, as will be exemplified in the next two sections, will be to use (6) in the context of different discretizations and domains to define a system of equations which determine the interfacial values of a solution of Poisson's equation. This system will be solved, and the resulting values will be used as boundary data for Poisson solves on subdomains to construct the remaining parts of the solution.

3. A Solver For The Region External to a Circular Cylinder. The domain under consideration is composed of two pieces, Ω_1 and Ω_2. Ω_1 is the annulus between the circle of radius r_a and a circle of radius r_b. Ω_2 is the region external to the circle of radius r_b. Our goal is to find the restriction of the solution to

$$\Delta u = f \qquad \text{for} \quad r_a < r \leq \infty \tag{7}$$

$$u = g \qquad \text{on } r = r_a$$

on the set of grid points of a polar grid in the annular region Ω_1. (See Figure 2.)

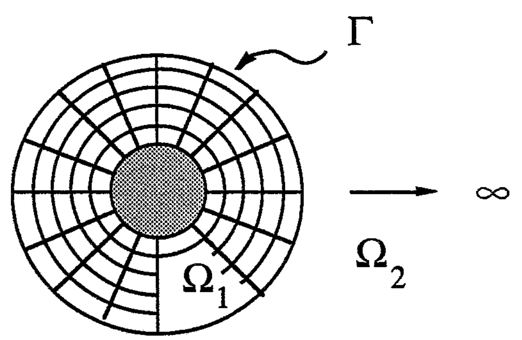

Figure 2

In order for our procedure to work as it is described here we assume that the support of the forcing function f is contained in Ω_1 and the logarithmic behavior of the solution to (7) at infinity is known. (In fluids problems in which the solution is a stream function, this is equivalent to specifying the circulation of the velocity field.)

Following the basic ideas of domain decomposition, the technique here will be to first find the values of the solution to (7) on the boundary Γ located at $r = r_b$. We shall refer to these values as u_Γ. We then employ a fast Poisson solver to calculate the remainder of the solution in Ω_1 by solving a problem with forcing function f and boundary values g on $r = r_a$ and u_Γ on $r = r_b$.

The determination of the solution values u_Γ is done by solving a finite dimensional analog of the equations described by (6) in the previous section. Taking note of our assumptions on the data of the problem, the equations which determine u_Γ. are

$$\frac{\partial \bar{u}_1}{\partial n}[u_\Gamma] + \frac{\partial \bar{u}_2}{\partial n}[u_\Gamma] = -\frac{\partial \bar{u}_1}{\partial n}[f_1, g] \tag{8}$$

In these equations $\bar{u}_1$ represents an approximate solution of Poisson's equation in Ω_1 with homogeneous forcing function and boundary data non-zero only along $r = r_b$. $\bar{u}_2$ is defined similarly. $\tilde{u}$ is the solution of Poisson's equation in Ω_1 with forcing function f and boundary data $\tilde{u}_1 = g$ for $r = r_a$ and $\tilde{u}_1 = 0$ for $r = r_b$. In order to reduce the problem to a finite dimensional one,we require that a relation of the form (8) hold at N equally spaced points on the boundary $r = r_b$. We take N to be the number of panels in the θ direction associated with the underlying finite difference approximation.

In this problem, Fourier analysis can be used effectively. The technique for constructing and solving the system of equations on the left hand side of (8) is done using the Fourier series solutions of the boundary value problems which implicitly define these operators. Given values of u_Γ at $N = 2m+1$ equally spaced points on the circle $r = r_b$ we first form the trigonometric interpolant -

$$u_\Gamma(\theta) = a_0 + \sum_{k=1}^{m} a_k \cos(k\theta) + \sum_{k=1}^{m} b_k \sin(k\theta)$$

The value of $\frac{\partial \bar{u}_1}{\partial n}[u_\Gamma]$, the normal derivative of the function obtained by solving a Laplace equation with this interpolated boundary data, is given by

(9) $$\frac{\partial \bar{u}_1}{\partial n} = \alpha_0 a_0 + \sum_{k=1}^{m} \alpha_k a_k \cos(k\theta) + \sum_{k=1}^{m} \alpha_k b_k \sin(k\theta).$$

Here,

(10) $$\begin{aligned} \alpha_0 &= \frac{1}{r_b}\frac{1}{log(r_b) - log(r_a)} \\ \alpha_k &= \frac{1}{r_b}\frac{(1+(\frac{r_b}{r_a})^{2k})}{(1-(\frac{r_b}{r_a})^{2k})} \end{aligned}$$

The same approach allows us to evaluate $\frac{\partial \bar{u}_2}{\partial n}$, and we find

(11) $$\frac{\partial \bar{u}_2}{\partial n} = \frac{\gamma}{r_b} + \sum_{k=1}^{m} \beta_k a_k \cos(k\theta) + \sum_{k=1}^{m} \beta_k b_k \sin(k\theta).$$

With

(12) $$\beta_k = \frac{k}{r_b}.$$

γ is the coefficient of the log term in the solution. (If the solution is a stream function, then $2\pi\gamma =$ the circulation of the velocity field.)

The right hand side of equation (8) can be calculated by using a second order difference approximation to $\frac{\partial \tilde{u}_1}{\partial n}$, where $\tilde{u}_1$ is the solution of the discrete Poisson problem associated with the equation

(13) $$\begin{aligned} \Delta\tilde{u}_1 &= f \quad \text{in } \Omega_1 \\ \tilde{u}_1 &= g \quad \text{on } r = r_a \\ \tilde{u}_1 &= u_\Gamma \quad \text{on } r = r_b. \end{aligned}$$

We denote this normal derivative by f_Γ.

What we have just described is the procedure for evaluating both sides of equation (8) at the points $r = r_b$ and $\theta_j = j(\frac{2\pi}{N})$ $j = 1 \ldots N$. Usually in domain decomposition procedures an iterative approach is taken to solve (8), and we have sufficient information to begin the process of creating an iterative scheme for solving the equations. However, the problem at hand is such that the matrix which represents our approximation to the operator on the right hand side of (8) diagonalized by the discrete fourier transform. Specifically, if we designate $\hat{f}_\Gamma$ as the discrete transform of f_Γ, then we have the following matrix equation for the coefficients of the transform of u_Γ,

$$\text{(14)} \qquad \begin{bmatrix} & & \\ & \text{A} & \\ & & \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} \hat{f}_{\Gamma 0} \\ \hat{f}_{\Gamma 1} \\ \\ \vdots \\ \\ \hat{f}_{\Gamma N} \end{bmatrix} + \begin{bmatrix} \frac{\gamma}{r_b} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Where A is the matrix

$$\begin{bmatrix} \alpha_0 & & & & & \\ & \alpha_1 + \beta_1 & & & & \\ & & \ddots & & & \\ & & & \alpha_m + \beta_m & & & \\ & & & & \alpha_1 + \beta_1 & & \\ & & & & & \ddots & \\ & & & & & & \alpha_m + \beta_m \end{bmatrix}$$

If we put all the pieces together, the algorithm is as follows:

(i) Solve (13) for $\tilde{u}_1$ and evaluate f_Γ by differencing the result.

(ii) Solve (14) for the coefficients of the transform of u_Γ. Evaluate u_Γ using the inverse Fourier transform.

(iii) Find the restriction to Ω_1 of the solution to (7) by solving

$$\Delta u = f \qquad \text{in } \Omega_1$$

$$u = g \quad \text{on} \quad r = r_a \qquad u = u_\Gamma \quad \text{on} \quad r = r_b$$

We have implemented the above algorithm, and performed some tests on the method. We were primarily interested in demonstrating that the effect of truncating the domain gives errors which are always on the order of the truncation error of the finite difference method used in Ω_1 (i.e. that the solution on the finite difference mesh is second order in δr and $\delta\theta$ independent of the mesh size). We expect that this should be the case since there is no error in the boundary conditions -we are using the exact analytic boundary conditions for the first m modes of the solution. In Table 1 we present the results of a problem for which $r_a = 1$ and $r_b = 2$. The solution computed was that induced by a unit charge located inside the cylinder and offset from the origin (so all modes of the boundary conditions would be tested). These results show that second order accuracy is preserved.

We were also interested in observing the change in the solution when the outer boundary of Ω_1 at $r = r_b$ was made larger, but the mesh size was kept fixed. Ideally there should be no change in the solution, but in light of our implementation, it can be expected to be on the order of the truncation error of the finite difference scheme. This behavior is reflected by the results presented in Table 2. Here the grid size was 16×64 for an outer radius of $r_b = 2$, and was changed to 32×64, 64×64 and 128×64 for the radii used to construct the table. The L^2 error of the 16×64 grid results was 1.580×10^{-3} while the L^∞ error was 8.437×10^{-4}

Mesh Size (r, θ)	L^2 Err.	L^∞ Err.
32×64	1.580×10^{-4}	8.437×10^{-4}
64×128	3.646×10^{-5}	1.972×10^{-4}
128×256	8.446×10^{-6}	4.928×10^{-5}

Table 1
Error With Change in Mesh Size

Outer Radius	L^2 Err.	L^∞ Err.
$r_b = 3$	1.883×10^{-3}	1.686×10^{-3}
$r_b = 5$	2.419×10^{-3}	2.183×10^{-3}
$r_b = 9$	2.646×10^{-3}	2.389×10^{-3}

Table 2
Error With Change in External Radius

4. Solver For An Infinite Step. In this section we use the ideas of domain decomposition to construct a solution to Poisson's equation in a domain which is an infinite step. Figure 3 shows the region which we are considering. The problem to be solved is

$$\Delta u = f \quad \text{in} \quad \Omega \tag{15}$$

$$u = 0 \quad \text{on } \partial\Omega$$

where $\Omega = \cup_{i=1}^{4} \Omega_i$. Ω_1 and Ω_4 are the unbounded components on the left and right ends of the step while Ω_2 is the region in which $x_a \leq x \leq x_b$, while Ω_3 is the adjoining region in which $x_b \leq x \leq x_c$. We designate the boundaries at $x = x_a$, $x = x_b$ and $x = x_c$ by Γ_a, Γ_b and Γ_c and the values of the solution to (15) on these boundaries as u_a, u_b and u_c. The idea is to compute u_a, u_b and u_c first and then obtain the restriction of the solution in the domains Ω_2 and Ω_3 by solving the appropriate Poisson problem in these domains. As before we assume that the support of f is compact, and contained in a region $x_a \leq x \leq x_c$.

The key differences between this problem and that for the cylinder is that the outer computational boundary (the one connecting the computational region to the infinite components of the domain) is not connected and the interior computational domain is not regular. This fact precludes the use of analytic procedures to evaluate the analog of equations (6) directly. However, as will be seen below, the construction of a solution procedure for this domain is only a minor modification of that for a region consisting of bounded rectangles. Moreover, the solution we find will be the restriction of the *exact* solution (up to roundoff) of the solution of the finite difference equations over the whole domain Ω.

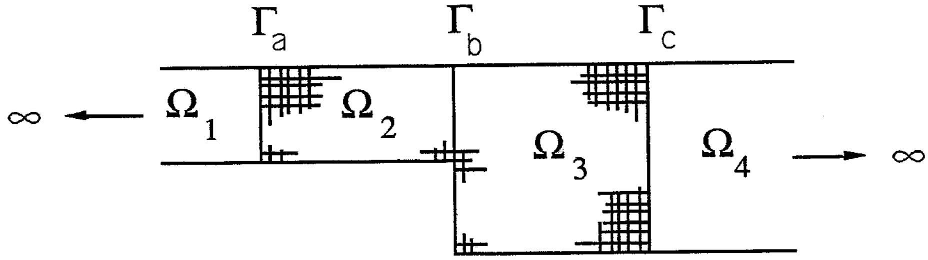

Figure 3

In the notation of the previous two sections, the conditions which determine the values of the solution along the interfaces are the following

(16) $$\frac{\partial \bar{u}_1}{\partial n}[u_a] + \frac{\partial \bar{u}_2}{\partial n}[u_a, u_b] = -\frac{\partial \tilde{u}_2}{\partial n}[f_2] \qquad \text{along} \quad \Gamma_a$$

(17) $$\frac{\partial \bar{u}_2}{\partial n}[u_a, u_b] + \frac{\partial \bar{u}_3}{\partial n}[u_b, u_c] = -(\frac{\partial \tilde{u}_2}{\partial n}[f_2] + \frac{\partial \tilde{u}_3}{\partial n}[f_3]) \qquad \text{along} \quad \Gamma_b$$

(18) $$\frac{\partial \bar{u}_3}{\partial n}[u_b, u_c] + \frac{\partial \bar{u}_4}{\partial n}[u_c] = -\frac{\partial \tilde{u}_3}{\partial n}[f_3] \qquad \text{along} \quad \Gamma_c.$$

As before, we interpret the terms in these equations operationally, that is

$$\frac{\partial \bar{u}_2}{\partial n}[u_a, u_b]$$

is the normal derivative of $\bar{u}_2$ the solution of

$$\Delta \bar{u}_2 = 0 \quad \text{in} \quad \Omega_2$$

$$\bar{u}_2 = u_a \quad \text{at} \quad \Gamma_a \qquad \text{and} \qquad \bar{u}_2 = u_b \quad \text{at} \quad \Gamma_b.$$

What is in brackets indicates what data is used for the Poisson problem and the subscript indicates in which domain the solution is computed.

We construct a finite dimensional analog of (16) - (18) by using finite difference approximations. The finite difference approximations are those which are inherited by the underlying discretization which we are attempting to solve rather than just difference approximations to the operators in (16) - (18). We assume that the underlaying finite difference mesh is a uniform rectangular one, with mesh widths δx and δy in the x and y directions respectively. If we consider the (i,j)th node of the finite difference grid along Γ_a, then the equations used to approximate (16) have the form

$$(19)\quad \frac{\bar{u}_1(i,j)-\bar{u}_1(i-1,j)}{\delta x}+(\frac{\delta x}{2\delta y})\frac{\bar{u}_1(i,j+1)-2\bar{u}_1(i,j)+\bar{u}_1(i,j-1)}{\delta y}+ \frac{\bar{u}_2(i+1,j)-\bar{u}_2(i,j)}{\delta x}+(\frac{\delta x}{2\delta y})\frac{\bar{u}_2(i,j+1)-2\bar{u}_2(i,j)+\bar{u}_2(i,j-1)}{\delta y} = \frac{\tilde{u}_2(i+1,j)-\tilde{u}_2(i,j)}{\delta x}+(\frac{\delta x}{\delta y})\frac{\tilde{u}_2(i,j+1)-2\tilde{u}_2(i,j)+\tilde{u}_2(i,j-1)}{\delta y}$$

The equations for the other interfaces are of similar form. In these difference equations there are differences in the direction tangent to the interface, whereas there are only normal derivatives in the flux balance equations (16) - (18). These terms arise in the finite difference implementation because tangential differences contribute to the discrete flux balance equations which are the analogs of (16) - (18). The evaluation of the terms in (19) and of the equations for the other interfaces is accomplished by solving the appropriate Poisson problem and then differencing the result. This procedure works well for the problems which are associated with the bounded components, but for the infinite components, this poses some difficulty. In particular, we must evaluate the difference approximations to $\frac{\partial \bar{u}_1}{\partial n}$ and $\frac{\partial \bar{u}_4}{\partial n}$ in which the functions $\bar{u}_1$ and $\bar{u}_4$ are solutions of a Poisson problem in half infinite tubes.

Fortunately, one can solve such problems exactly using discrete Fourier analysis. Specifically, if we have discrete data along the interface Γ_a of the form

$$(20)\quad u_a(y_j)=\sum_{k=1}^{m} b_k \sin(\frac{2\pi y_j}{2L})$$

where $y_j = y_a + (j-1)\delta y$ and L is the length of Γ_a, then a solution of a discrete Laplace equation in the half-tube Ω_1 is given by

$$(21)\quad \bar{u}_1(x_i,y_j)=\sum_{k=1}^{m} \lambda_k^{i-1} b_k \sin(\frac{2\pi y_j}{2L})$$

where the values λ_k are given by

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

The explicit form of (21) allows us to compute the differences needed in the evaluation of (19). The evaluation of the term involving $\bar{u}_4$ is accomplished in a similar fashion.

In light of the specific form of the solutions which define $\bar{u}_1$ and $\bar{u}_4$, we see that the evaluation of the appropriate differences can be accomplished with fast sin transforms. It is also the case that the other operators in the difference approximations to (16) - (18) can be evaluated using discrete sine transforms. This follows because all of the boundary value problems which define these operators can be solved explicitly using discrete Fourier analysis. (See [2] for more details.) Thus, the forward application of the operators which determine the boundary values u_a, u_b, and u_c can be carried out using fast sine transforms.

There is a natural block structure to the equations when one groups the unknowns together according to which interface they occur on. With this blocking the discrete equations for the values u_a, u_b and u_c take the form,

$$\begin{bmatrix} A^1_{11}+A^2_{11} & B_{12} & 0 \\ B_{21} & A^1_{22}+A^2_{22} & B_{23} \\ 0 & B_{32} & A^1_{33}+A^2_{33} \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} \tilde{f}_a \\ \tilde{f}_b \\ \tilde{f}_c \end{bmatrix} \tag{22}$$

Here $\dfrac{\partial \bar{u}_1}{\partial n}$ is approximated by A^1_{11}, $\frac{\partial \bar{u}_2}{\partial n}$ is approximated by A^2_{11} etc. The right hand side of the equations, designated by $(\tilde{f}_a, \tilde{f}_b, \tilde{f}_c)^t$ are determined by differencing solutions of the appropriate finite difference Poisson problem.

This equation is not completely diagonalized by the discrete sin transform, for the number of points which define the transform used to evaluate the equations for Γ_b are different for the terms involving u_2 and u_3. However, the matrices in the first and third rows are simultaneously diagonalized by the fast sine transform.

Unlike the cylinder case, the equations are not solved directly but iteratively. As is typical with domain decomposition techniques we used pre-conditioned conjugate gradients. The preconditioner used was the inverse of the matrix

$$\begin{bmatrix} A^1_{11}+A^2_{11} & 0 & 0 \\ 0 & A^1_{22} & 0 \\ 0 & 0 & A^1_{33}+A^2_{33} \end{bmatrix}.$$

Each of these matrices is diagonalized by the discrete sin transform, so the application of the preconditioner can be accomplished using fast sin transforms. This preconditioner worked well and converged rapidly. (This is to be expected by analogy with results for the bounded domain case [4])

In sum the algorithm is as follows:

(I) Solve appropriate Dirichlet problems in Ω_2 and Ω_3 and evaluate $(\tilde{f}_a, \tilde{f}_b, \tilde{f}_c)^t$ by differencing the result.

(ii) Solve (22) using the method of pre-conditioned conjugate gradients. (Use the fact that the the foward action of the operator in (22) can be computed using fast sin transforms.)

(iii) Find the restriction to Ω_2 and Ω_3 of the solution to (15) by solving Dirichlet problems in each of these domains.

In the first test of this method, a fixed uniform mesh (with $\delta x = \delta y = 0.1$) was used and the upstream and downstream boundaries were changed. The domain was taken to be an infinite tube with width 0.8 on the left and 1.0 on the right. A unit source was placed at a fixed location in Ω_2 and the values in Ω_2 were monitored when the computational domains Ω_2 and Ω_3 were extended from a length of 1.0 to 2.0 and from 2.0 to 4.0. The maximum change which occurred was 9.357×10^{-5} and 9.359×10^{-5} - a quantity on the order of the accuracy with which the fast Poisson solver can calculate the solution to the discrete Laplacian. This degree of accuracy is to be expected since we are constructing an exact solution to the finite difference equations in the whole infinite region. In the second test we measured the number of conjugate-gradient iterations which were needed to reach a given residual size. In each case the domain was fixed to be of width 0.8 on the left and 1.0 on the right. The length of Ω_2 and Ω_3 was fixed at 1.0. In Table 3 we present the L^2 norm of the residual for different size grids. As can be seen from the table the solution converges rapidly demonstrating the efficiency of the conjugate-gradient method with our choice of pre-conditioner.

Iterations	$\delta x = \delta y = 0.1$	$\delta x = \delta y = 0.05$	$\delta x = \delta y = 0.025$
1	3.813×10^{-2}	8.693×10^{-2}	1.923×10^{-1}
2	6.494×10^{-4}	1.747×10^{-3}	6.317×10^{-3}
3	1.332×10^{-5}	1.343×10^{-5}	8.397×10^{-5}

Table 3
Residual Error

5. Conclusion. We have discussed the manner in which domain decomposition techniques can be used to construct solvers for infinite domains. The central idea is to observe that the conditions which determine the interface values (values which are necessary to implement the domain decomposition technique) are derived from a condition of an equality of normal derivative's or a flux balance condition. With this observation, such conditions can be used to derive equations for the interfacial values even if one or more of the domains is of infinite extent. The two applications, one for a domain external to a cylinder and one for an infinite step exhibit the potential of the approach.

REFERENCES

[1] C. R. Anderson, *Vorticity Boundary Conditions and Boundary Generation of Vorticity*, J. Comp. Phys. *to appear.*

[2] C. R. Anderson, *The Application of Domain Decomposition to the Solution of Laplace's Equation in Infinite Domains*, CAM Report 87-19, Dept. of Mathematics, UCLA, 1987.

[3] J. H. Bramble and J. E. Pasciak, *A Boundary Parametric Approximation To The Linearized Scaler Potential Magnetostatic Field Problem*, App. Num. Math. 1, 1985, pp. 493-514.

[4] T.F. Chan and D.C. Resasco, *A Survey of Preconditioners for Domain Decomposition*, Research Report YALEU/DCS/RR-414, September 1985.

[5] S. I. Hariharan, *Absorbing Boundary Conditions for Exterior Problems*, ICASE Report #95-33, NASA Langley Research Center, Virginia, 1985.

[6] F. Kang and Y. De-hao, *Canonical Integral Equations of Elliptic Boundary-value Problems and Their Numerical Solutions*, Proceedings of the China-France Symposium on Finite Element Methods, Ed. F. Kang and J.L. Lions, 1983, pp. 211-252.