A DOMAIN DECOMPOSITION METHOD FOR INCOMPRESSIBLE VISCOUS FLOW

by

John C. Strikwerda
and
Carl D. Scarbnick

Computer Sciences Technical Report #896

December 1989

A DOMAIN DECOMPOSITION METHOD FOR INCOMPRESSIBLE VISCOUS FLOW

John C. Strikwerda† and Carl D. Scarbnick

Department of Computer Sciences

and

Center for Mathematical Sciences

University of Wisconsin-Madison

Abstract. A method for using domain decomposition to solve the equations of incompressible viscous flow is presented. The method is described in detail, and test results are given for two test problems. A notable feature of the method is that the incompressibility constraint is never imposed. The domain decomposition uses finite difference and spectral methods on overlapping domains, with second-order accurate interpolation of the velocity relating the solutions on the different domains. The method is shown to be globally second-order accurate by the test results.

Key words. domain decomposition, incompressible viscous flow, overlapping grids

AMS(MOS) subject classification. 65N20, 65N05

1. Introduction

In applying finite difference methods to a specific problem one of the most important considerations is the design of the finite difference grid. The choice of a finite difference grid is easily made for problems with simple geometry, such as rectangular or circular domains, however, for problems on domains which are more complex the designing of a grid can be a significant problem. The choice of grid also affects many aspects of the finite difference method. The efficiency of the numerical solution algorithm and the accuracy of the computed solution are strongly dependent of the finite difference grid. The type of grid can also restrict the choice of the numerical algorithm.

In this paper we present a method of using overlapping grid systems for solving incompressible fluid dynamics problems. By using several grids which overlap one has more flexibility in the placement of grid points than by using one global grid system. The crucial aspect of using overlapping grids is the interpolation of values between grids. This paper discusses the grids and interpolation methods used on particular test problems and presents insights into the use of overlapping grids for other problems. It is shown

† The first author was partially supported by Contracts N00024-85-M-B720 of the Naval Sea Systems Command and DAAL03-87-K0028 of the U.S. Army Research Office.

that overlapping grid systems can be used to obtain accurate solutions to the equations for incompressible viscous flow. The solutions are obtained efficiently using iterative methods.

Our method is a special case of the general method of domain decomposition, which has been applied to single second-order elliptic equations by many researchers, see the collection of papers in [5], and to incompressible viscous flow by [1, 3, 4] using the finite element method. Much of the work on domain decomposition revolves around methods of accelerating the iterative solution procedure and implementation of domain decomposition on parallel computers. The emphasis of this paper is on the features of the domain decomposition that are independent of the particular solution algorithm used on each domain. The iterative method used in this work is not to be regarded as very fast. In a subsequent publication we will describe a multigrid iterative method that, based on simple test cases, promises to significantly improve the convergence rate [13].

A notable feature of the method presented here is the handling of the conservation of mass. By properly reformulating the equations, the troublesome integrability constraint on the boundary data is removed. The divergence-free nature of the solution is obtained as a result of the consistency of the scheme to the differential equation.

In this paper we regard the overlapping grids as several overlapping domains each with its own grid which is independent of other grids. Thus we refer to our method as the method of overlapping domains. This way of referring to the method calls attention to the central problem of relating the values of the solution on the several domains and deemphasizes the construction of the particular grids. As shown in section 5, some of the primary difficulties are not specific to the choice of grid or the finite difference scheme.

We also restrict our attention to the steady Stokes equations because they are useful to illustrate the basic ideas of the method of overlapping domains as applied to incompressible flow. Overlapping domains can be used with the incompressible Navier-Stokes equations, both steady and time-dependent. The extra features introduced by the more general equations can be incorporated into the method without significant difficulties. The method is applicable to three-dimensional problems.

Also, related to the work related here is the work of Thompson and Ferziger [16] and Vanka [17] in which they applied the multigrid iteration scheme to the incompressible Navier-Stokes equations. They each used staggered grids which limit their methods to rectilinear domains. As we demonstrate by our computational results, our method maintains second-order accuracy even when the grids are not both rectilinear, see also [10]. A use of the multigrid iteration method to accelerate the rate of convergence of the iterative method will appear soon [13].

A typical example of the use of overlapping grids is in the computation of the flow past several bodies

with non-simple shapes. Boundary-fitted coordinate systems should be used near the bodies to insure good accuracy. However, away from the bodies it is simplest and most convenient to use a standard global cartesian grid or other such simple grid. One may indeed blend the boundary-fitted grids with the grid system used away from the bodies, but it is not done easily nor can it be done by any simple general procedure. The approach advocated here is that of constructing the boundary-fitted and the global grids independently, but large enough to overlap. The values of the solution on the boundary of each grid are determined by interpolation of values from the other grids.

The chief advantages of the use of overlapping grids are the flexibility and generality of the method. It can be used with any geometric configuration and, in principle, the grid can be altered as needed. Local refinements grids and adaptive grid methods can also be used to improve the resolution of the solution. The flexibility of the method can be seen in comparison with global grid generation techniques and grid patching techniques. The need for higher resolution in one area of the domain usually forces both higher resolution in areas that do not need the resolution and lower resolution in other areas.

The use of overlapping grids can be used with the grid generation method to produce grids which enable solutions to be computed efficiently. The elliptic grid generation method of Thompson [15], and methods of Kreiss [7], and Chesshire [2] can be used to construct boundary-fitted grids around bodies. Farther from the bodies a cartesian grid or similar simple grid can be used. Boundary-fitted grids are essential for obtaining accurate solutions near bodies, while the use of simple cartesian grids is important for obtaining efficiency in computations.

2. The Stokes Equations on Overlapping Domains

We first consider questions of uniqueness and existence of solutions to the steady Stokes system of equations on overlapping domains. These questions are not difficult to resolve, but they are interesting and important because of the insight they give to the numerical approximations.

We begin by considering two domains Ω_1 and Ω_2 which overlap, i.e., $\Omega_1 \cap \Omega_2$ is nonempty, see Figure 1. We also assume that neither domain is included in the other. We consider the Stokes system of equations, given by

$$\nabla^2 \vec{u} - \vec{\nabla} p = 0 \tag{2.1}$$

$$\vec{\nabla} \cdot \vec{u} = 0, \tag{2.2}$$

on each separate domain. Boundary data for each domain is specified only on that part of the boundary which is not interior to the other domain. For purposes of discussion, we assume that the velocity $\vec{u}$ is specified by data $\vec{b}$ on the boundary of $\Omega_1 \cup \Omega_2$. For that part of $\partial\Omega_1$ (the boundary of Ω_1) that lies within

Ω_2 we require that the velocity be equal to the velocity obtained from the solution of the Stokes system on Ω_2, and similarly for that portion of Ω_2 within Ω_1.

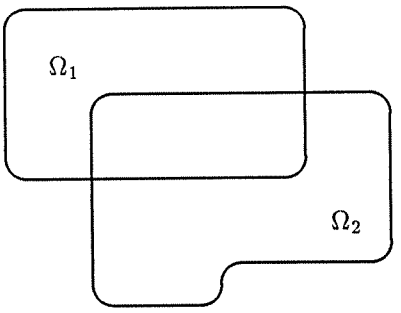

Figure 1.

We also assume that the integrability condition holds on the whole domain, i.e.,

$$\int_{\partial(\Omega_1 \cup \Omega_2)} \vec{b} \cdot \vec{n} = 0. \tag{2.3}$$

In our finite difference method some of the boundary data for a subdomain will be obtained by interpolation from other domains, and it would be difficult to impose the integrability constraint on the data for each subdomain. Thus we will consider a slightly more general case in which the divergence of the velocity field is required to be constant. That is, (2.2) is replaced with

$$\vec{\nabla} \cdot \vec{u}_i = d_i \quad \text{on} \quad \Omega_i. \tag{2.4}$$

The constant d_i is determined by the integrability condition for the domain, i.e.,

$$\int_{\partial\Omega_i} \vec{u}_i \cdot \vec{n} = d_i \, |\Omega_i|. \tag{2.5}$$

As will be seen, this formulation avoids the difficulties inherent in methods that require that the boundary data satisfy the integrability constraint on each subdomain, see [1].

The mathematical statement of the problem is then described as follows. In Ω_1

$$\begin{aligned} \nabla^2 \vec{u}_1 - \vec{\nabla} p_1 &= 0 \\ \vec{\nabla} \cdot \vec{u}_1 &= d_1 \end{aligned} \tag{2.6}$$

and in Ω_2

$$\begin{aligned} \nabla^2 \vec{u}_2 - \vec{\nabla} p_2 &= 0 \\ \nabla \cdot \vec{u}_2 &= d_2 \end{aligned} \tag{2.7}$$

with boundary conditions

$$\begin{aligned} \vec{u}_1 &= \vec{b} \quad \text{on} \quad \partial\Omega_1 \backslash \Omega_2 \\ \vec{u}_1 &= \vec{u}_2 \quad \text{on} \quad \partial\Omega_1 \cap \Omega_2 \\ \vec{u}_2 &= \vec{b} \quad \text{on} \quad \partial\Omega_2 \backslash \Omega_1 \\ \vec{u}_2 &= \vec{u}_1 \quad \text{on} \quad \partial\Omega_2 \cap \Omega_1, \end{aligned} \tag{2.8}$$

where $\vec{b}$ is the velocity data on the boundary of $\Omega_1 \cup \Omega_2$.

We make the assumption that the Stokes system of equations (2.6) and (2.7) have a unique solution for each domain here when the velocity is specified on the boundary. By a unique solution we mean that the velocity function is determined uniquely and that the pressure function is determined to within an additive constant. The uniqueness of the solution to (2.6) and (2.7) with the velocity specified on the boundary follows from the uniqueness of the Stokes system (2.1) and (2.2). This assumption is certainly justified for all problems of physical significance.

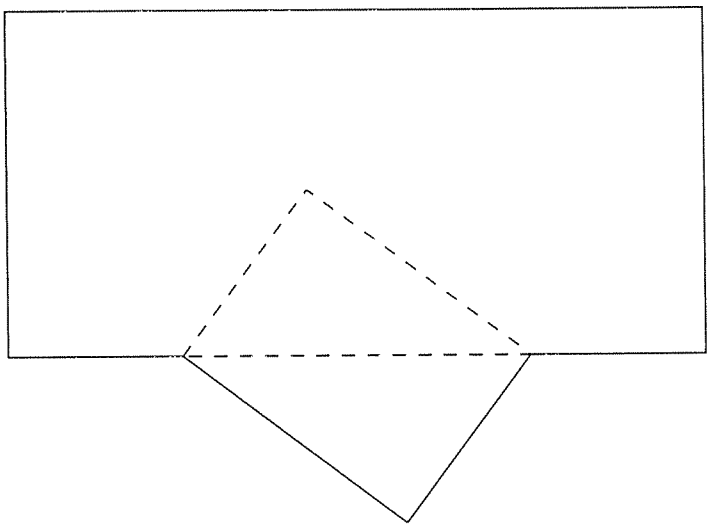

Figure 2

Notice that we do not assume a priori that $\vec{u}_1$ and $\vec{u}_2$ are the same on $\Omega_1 \cap \Omega_2$. Also, notice that the pressure on one domain does not directly interact with the pressure on the other domain. In particular, we can not specify the pressure as a boundary condition along with the velocity as this would result in over-determined boundary value problems on the subdomains.

It is useful to consider the following iterative procedure, essentially the Schwarz alternating procedure (see [8]), to solve the overlapping domain problem. We begin with any velocity field $\vec{u}_1^0$ defined on Ω_1. This

need not be divergence-free. Using $\vec{u}_1^0$ as the boundary condition on $(\partial\Omega_2) \cap \Omega_1$ solve the system (2.7) for $(\vec{u}_2^0, p_2^0)$. Then using $\vec{u}_2^0$ as boundary data on $(\partial\Omega_1) \cap \Omega_2$ solve (2.6) for $(\vec{u}_1^1, p_1^1)$. Continue in this way, so that $(\vec{u}_1^\nu, p_1^\nu)$ is determined from $\vec{u}_2^{\nu-1}$ and $(\vec{u}_2^\nu, p_2^\nu)$ is determined from $\vec{u}_1^\nu$. The values of d_1^ν and d_2^ν are determined at each step by the integrability conditions. Supposing that the iterative method converges, we may indeed ask if the final values of constants d_1 and d_2 are the same, and if $(\vec{u}_1, p_1)$ and $(\vec{u}_2, p_2)$ together may be regarded as a solution on the total domain $\Omega_1 \cup \Omega_2$. The convergence of this procedure can be shown using the methods of [8].

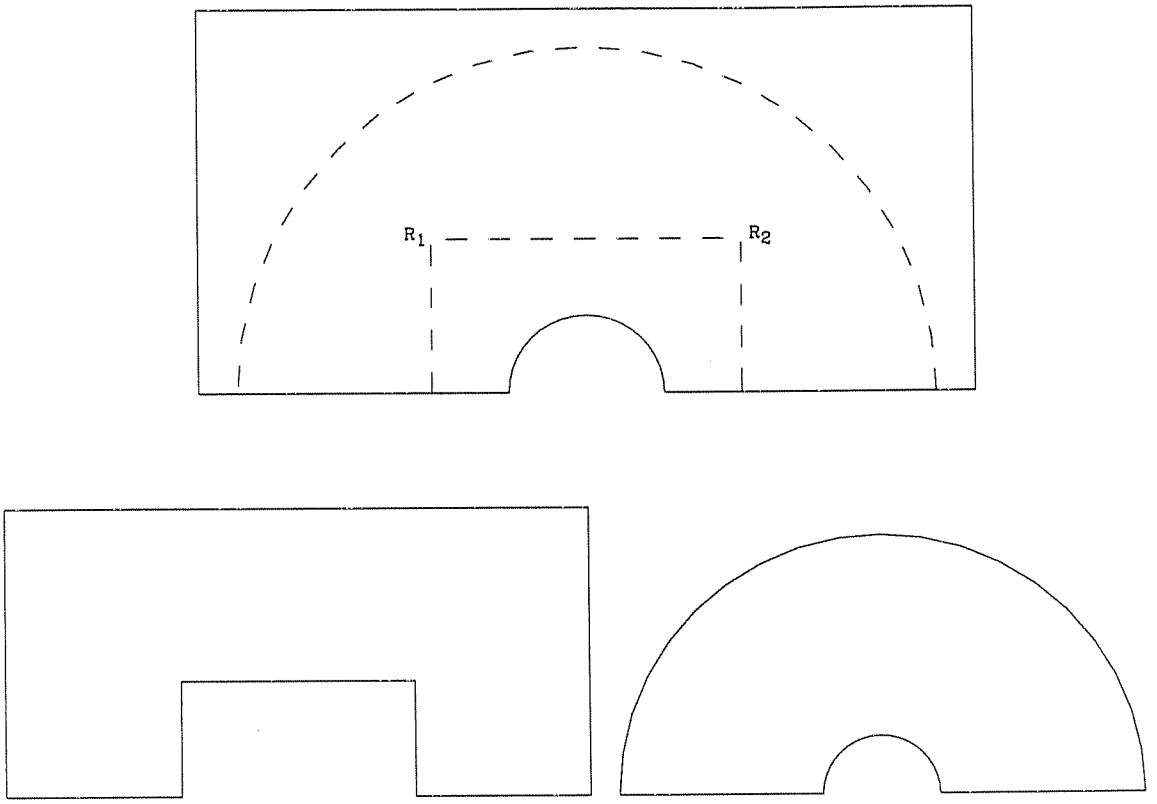

Figure 3

First, a solution to this problem exists. This is seen by considering the one problem defined on $\Omega_1 \cup \Omega_2$, the union of the two domains. The data vector function $\vec{b}$ is defined on boundary of $\Omega_1 \cup \Omega_2$, and thus by our assumption, a solution exists.

Next we show that the solution is unique. Assuming that solutions $(\vec{u}_1, p_1)$ and $(\vec{u}_2, p_2)$ exist on Ω_1 and Ω_2, respectively, the first question to be addressed is whether they agree on the intersection $\Omega_1 \cap \Omega_2$. Since $\vec{u}_1$ and $\vec{u}_2$ are equal on the boundary of $\Omega_1 \cap \Omega_2$, it follows by our uniqueness assumption that there is a unique solution and that d_1 and d_2 must be equal by the integrability condition on $\Omega_1 \cap \Omega_2$. Moreover the pressure functions p_1 and p_2 differ at most by an additive constant. It then follows that the two solutions

$(\vec{u}_1, p_1)$ and $(\vec{u}_2, p_2)$ can be regarded as a solution to the global boundary value problem on $\Omega_1 \cup \Omega_2$. Hence d_1 and d_2 are zero because of the integrability condition (2.3) on $\Omega_1 \cup \Omega_2$.

An observation that is important in the numerical approximation is that the two pressure fields, p_1 and p_2, differ by a constant whose value is not determined by the solution. Thus there are two undetermined parameters in the solution; the average value of p_1 on Ω_1 and the average value of p_2 on Ω_2. In general, there is an undetermined additive constant for the pressure on each subdomain in the decomposition.

Another important observation is to notice that it is by comparing the two different problems on the intersection of the domains that we see that they define a global solution on the union of the domains. This suggests that the *conditioning* of the problem is dependent on the size of the intersection. This conditioning is evident in the numerical investigations. A smaller overlap domain requires more iterations for convergence and gives less accuracy in the answers.

Finally, we observe that essentially the same argument holds in case one domain is a subset of the other. This situation arises when a finer grid is used to better resolve some feature in the interior of a flow domain.

3. Interpolation Between Domains

An important feature of the approach taken here is that the overall method is composed of relatively independent components. For example, the finite difference methods used on the different subdomains need not be the same. Also, the exact form of the iterative methods on the subdomains do not depend on each other.

The topic of this section, namely, the interpolation between domains, is also relatively independent of the other aspects of the overall problem. In particular, the interpolation is not dependent on the finite difference methods nor on the iterative method. As shown in this section, the choice of interpolation method is primarily based on convenience and the need for accuracy in the solution.

For each of the test cases for which the numerical experiments were made there were two grids. In the first set of cases the two grids are both cartesian grids, but rotated with respect to each other, see Figure 2, while for the second set of test cases, one grid is a standard cartesian grid with uniform spacing and the other grid is a polar coordinate grid, also with uniform spacing in each coordinate direction, see Figures 3. Rather than discuss the interpolation in general we will confine ourselves to these two situations.

The basic interpolation problem is a local problem, that is, given a discrete function defined on a cartesian grid, how should a value be assigned to a point which is not on the grid? To mathematically formulate the problem in two space dimensions we consider a uniform cartesian grid with points (x_i, y_j) given by $x_i = ih$ and $y_j = jh$, where the grid spacing, h, is a positive number. Given a general point $(\bar{x}, \bar{y})$ there is a grid point (x_i^*, y_j^*) which is closest to $(\bar{x}, \bar{y})$. A second parameter of significance is the quadrant

relative to (x_i^*, y_j^*) in which $(\bar{x}, \bar{y})$ lies. Without loss of generality we can assume that (x_i^*, y_j^*) is $(0, 0)$. The general case can be reduced to this special case by a simple translation. We also can assume, to simplify the discussion, that $(\bar{x}, \bar{y})$ is in the first quadrant, that is both $\bar{x}$ and $\bar{y}$ are positive.

The first interpolation method which we consider is bilinear interpolation. In this case the value of a function at $(\bar{x}, \bar{y})$ is given by

$$\begin{aligned} f = f_{0,0} + \bar{x}\,(f_{1,0} - f_{0,0})\,/h + \bar{y}\,(f_{0,1} - f_{0,0})\,/h \\ + \bar{x}\bar{y}\,(f_{1,1} - f_{1,0} - f_{0,1} + f_{0,0})\,/h^2 \end{aligned} \tag{3.1}$$

where $f_{0,0} = f(x^*, y^*)$, $f_{1,0} = f(x^* + h, y^*)$, $f_{0,1} = f(x^*, y^* + h)$, and $f_{1,1} = f(x^* + h, y^* + h)$. Bilinear interpolation is only accurate of order one. The order of accuracy is equal to the highest degree of polynomial for which the interpolation must be exact. Thus (3.1) is exact for any polynomial of first degree, but is not exact for all second degree polynomials. The use of bilinear interpolation gave only first-order accuracy in the solution.

The second interpolation method which we consider is quadratic interpolation, see Figure 4. The particular formula we used is

$$\begin{aligned} f = f_{0,0} + \bar{x}\,(f_{1,0} - f_{-1,0})\,/2h + \bar{x}^2\,(f_{1,0} - 2f_{0,0} + f_{-1,0})\,/2h^2 \\ + \bar{y}\,(f_{0,1} - f_{0,-1})\,/2h + y^2\,(f_{0,1} - 2f_{0,0} + f_{0,-1})\,/2h^2 \\ + \bar{x}\bar{y}\,(f_{1,1} - f_{1,0} - f_{0,1} + f_{0,0})\,/h^2. \end{aligned} \tag{3.2}$$

In the test cases it was found that quadratic interpolation gave overall second-order accuracy, which is the same order as the finite difference schemes, so no methods of higher order were investigated.

Henshaw and Chesshire [6] reduced the amount of overlap as the grid spacing was reduced, and reported that a third-order interpolation was necessary to maintain the over-all second-order accuracy of the solution. Since we consider the domain fixed, independent of the grid spacing, we are able to use interpolation of the same order as the scheme.

For the interpolation from the polar grid to the cartesian grid in the second test problem, a method based on Fourier interpolation was used. This method was very natural to use because the numerical method on the polar grid used finite Fourier series to approximate the derivatives with respect to the angular variable. The finite Fourier series are discussed in section 6.

The polar grid used in the numerical experiments consisted of grid points (r_i, θ_j) where $r_i = r_{\min} + i \cdot \Delta r$ for $0 \le i \le N$ and $\theta_j = j \cdot \Delta\theta$ for $0 \le j \le J$ with $\Delta\theta = \pi/J$ and $\Delta r = (r_{\max} - r_{\min})\,/N$. The even spacing of the grid is not crucial to the method; it was used for simplicity. Indeed, because the grid spacing in the

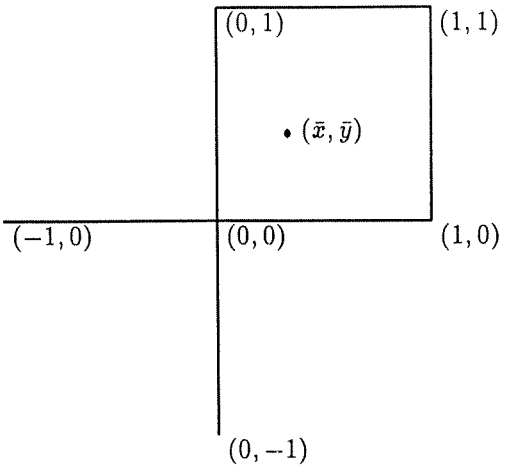

Figure 4.

two grids was independent of each other, good resolution with the polar grid could easily be obtained using a uniform grid without significantly increasing the overall computational effort.

The actual interpolation procedure was as follows. Given a point $(\bar{x}, \bar{y})$ on the boundary of the cartesian grid for which data had to be supplied from the polar grid for a function f, its polar coordinates $(\bar{r}, \bar{\theta})$ were first computed. The value of $r_{i^\bullet}$ was then determined such that $r_{i^\bullet} \leq \bar{r} < r_{i^\bullet+1}$. Then using the finite Fourier series on $r_{i^\bullet}$ and $r_{i^\bullet+1}$ values were obtained for $f\left(r_{i^\bullet}, \bar{\theta}\right)$ and $f\left(r_{i^\bullet+1}, \bar{\theta}\right)$. A value for $f\left(\bar{r}, \bar{\theta}\right)$ was obtained by one-dimensional linear interpolation, i.e.,

$$f\left(\bar{r}, \bar{\theta}\right) = \frac{\left(r_{i^\bullet+1} - \bar{r}\right) f\left(r_{i^\bullet}, \bar{\theta}\right) + \left(\bar{r} - r_{i^\bullet}\right) f\left(r_{i^\bullet+1}, \bar{\theta}\right)}{\Delta r}. \tag{3.3}$$

This formula is second-order accurate.

Another formula that was used was to find the three nearest values of r_i to $\bar{r}$, and use quadratic interpolation on the three approximations to $f\left(r_i, \bar{\theta}\right)$, each of which is computed using the finite Fourier series. This method was more accurate than (3.3), but was also more sensitive. It gave less accurate values if the corner singularity, which is discussed in section 5, was not treated properly.

Both the bilinear interpolation (3.1) and quadratic interpolation (3.2) were tested in place of (3.3) to interpolate values from the polar grid to the cartesian grid. Both of these methods were far less accurate than was the use of the finite Fourier series. The inaccuracy of the quadratic interpolation for this case is due to the large grid spacing in the angular direction.

To interpolate the velocity from one grid to another, the different representation of the velocity on each grid must be accounted for. We found it best to interpolate the velocity components independently, and then transform them to the proper representation. For example, in interpolating from a cartesian representation to a polar representation, the cartesian velocity components are interpolated, and then transformed to the polar representation of the velocity.

The basic data for interpolation to a given point from a grid consists of the coordinates of the nearest grid point and the quadrant in which the given point lies relative to the nearest grid point. In many applications it would be best to determine this data prior to the main calculation. In the applications employed here the basic data was computed as needed. Because of the simplicity of the two grids used, i.e., cartesian and polar, the necessary computations were relatively easy and straight-forward. For calculations being done on vector processors it would presumably be more efficient to store the interpolation data, rather than recompute it. This would also be true when using any grid for which it is difficult to determine the nearest grid point to a given point.

4. The Iterative Solution Procedure

Before discussing the finite difference approximation we give a brief description of the iterative solution procedure. This is done before the discussion of the finite difference methods themselves to highlight those aspects which relate to the considerations discussed in the section on the overlapping domains for the Stokes system.

The iterative method used on the cartesian domain is based on point successive-over-relaxation and is described in detail in [11]. The iterative method used on the polar domain is based on line successive-over-relaxation and is well suited for methods using finite Fourier series methods. This method was also used in [12] where it is described in detail. Multigrid methods are also suitable to be used, see [13].

The basic iteration step to update the solution on a domain consists of two parts. The first part consists of updating the velocity using one step of a method such as successive-over-relaxation. This step uses only the first vector equation in (2.6) or (2.7). In the second part, the pressure is updated based on the local velocity divergence, which is the second equation in (2.6) or (2.7).

For the cartesian grid, assuming equal grid spacing of h and assuming the standard second-order accurate five-point laplacian, these formulas are:

$$\begin{aligned} u_{ij} &\leftarrow u_{ij} + \omega \left(\frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{i,j} - \frac{h^2}{4}\delta_{x,r} p_{i,j} \right) \\ v_{ij} &\leftarrow v_{ij} + \omega \left(\frac{1}{4}(v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1}) - v_{i,j} - \frac{h^2}{4}\delta_{y,r} p_{i,j} \right) \end{aligned} \tag{4.1}$$

where $\delta_{x,r}$ and $\delta_{y,r}$ represent regularized central difference approximations to the derivatives with respect to x and y, respectively. Formulas for $\delta_{x,r}$ and $\delta_{y,r}$ are given in section 6. The algorithm uses immediate replacement, so that only one copy of the solution needs to be kept. The iteration parameter ω was chosen as for standard succesive-over-relaxation, see section 7.

After the velocity has been updated, the pressure is updated by,

$$p_{ij} \leftarrow p_{ij} - \gamma(\delta_{x,r} u_{ij} + \delta_{y,r} v_{ij}) \tag{4.2}$$

for all interior points. Here, as before, $\delta_{x,r}$ and $\delta_{y,r}$ represent regularized central difference approximations, but have a different shift than they did for the gradient terms in (4.1). The iteration parameter γ was chosen proportional to the grid spacing, see Section 7. After all the interior points were updated, the boundary values of the pressure are set by quadratic extrapolation using formulas such as

$$p_{0j} = 3(p_{1j} - p_{2j}) + p_{3j}$$

along the boundary given by $i = 0$. Similar formulas are used along the other boundaries. This extrapolation is of high enough order not to affect the overall second-order accuracy of the solution, see [14]. The stopping criteria for the iterative method are discussed in Section 7.

It is important to notice that at no time is it required that the velocity field be divergence-free or satisfy a similar numerical condition. Also, the pressure is not constrained so as to determine the unimportant additive constant. This avoids entirely problems such as those arising in the use of divergence-free finite elements, see [1].

When solving a problem on overlapping domains the procedure we have adopted is as follows. The solution on one domain, say Ω_1, is updated as in one step of the iterative method for a single domain. The values of the velocity on the boundary of Ω_2 are determined by interpolation from the latest values on Ω_1, and then the solution is updated on Ω_2 as in one step of the iterative method for a single domain. The boundary values of Ω_1 are then updated by interpolation of the velocity on Ω_2. This describes one step of the iterative method. The iteration proceeds until the solution on each domain has converged to within the specified tolerance.

There are several observations to be made about this procedure. First, the pressure fields on the two domains do not directly effect each other. In particular, their difference in the overlap is not restricted to be a constant. In fact, it is not clear how best to compare the pressure values since they are defined on different grids. Secondly, as in the proof of the uniqueness of the solution in section 2, the divergence-free property of the flow is not imposed, it is a result of the nature of the system.

These two observations can be used to give a posteriori error checks. The deviation of the velocity fields from being divergence-free and the deviation from a constant of the difference of the two pressure fields on the overlap can be used to indicate the accuracy of the final solution.

We have made no attempt to prove that this iterative method will converge. It appears that such a proof is beyond our current methods of analysis. In practice the method has performed very well. The solutions have been obtained in a reasonable number of steps.

The conditioning of the method for the second test problem was found to be related to the size of the overlap region, as suggested by the analysis of the differential system. The method converged faster, and

gave more accurate results, when the overlap region was larger.

5. The Reentrant Corner Difficulty

One difficulty which was discovered and finally surmounted is of sufficient interest to warrant special attention. This difficulty occurred with the second test problem and has to do with the corners of the cartesian grid resulting from the exclusion of portions of that grid from the computational domain. These corners, such as the points R_1 and R_2 in Figure 3, are distinguished by being reentrant corners.

For the Navier-Stokes or Stokes equations in a region with reentrant corners the solution will in general have a pressure singularity at such a corner. Usually this is studied only for flows over steps, in which the velocity is zero along the sides of the step, see [9], however, the pressure singularity exists at reentrant corners for general data.

For the overlapping domains of the second problem the true solution has no singularity at the reentrant corners since the reentrant corners are interior to the other domain. However in the numerical approximation, the data given by the interpolation introduces errors which cause a pressure singularity to appear in the error. In the test problem the graphical display of the error showed that these pressure singularities were the dominant feature of the error. The singularity in the pressure error caused a vortical flow in the velocity error. The discovery of the pressure error singularity was made possible by graphically displaying the errors.

Two approaches were used to improve the solution by decreasing the strength of these pressure error singularities. The first of these we call "rounding off the corner." It is based on the observation that the grid points at the reentrant corners can be treated as interior points of the cartesian grid. Both the pressure and velocity at these corner points were updated as interior points. This is possible because all the nearest neighbors are also part of the grid. Interpolation is done then only at points which are missing at least one neighbor in the grid. This simple procedure gave a dramatic improvement in the accuracy of the solution.

The second method was to "chop off" the corner so that the one 270° angle is replaced by two 225° angles. This method introduced more programming difficulties and did not give a significant improvement over the method of rounding off the corner, and so will not be discussed further.

It is important to remember that the reentrant corner pressure error singularity is inherent in the use of overlapping domains. This can be seen by considering the overlapping domain problem for the differential equations as discussed in section 2. If we assume that the specification of boundary values on the overlap portions of the boundary, i.e., $(\partial\Omega_2) \cap \Omega_2$ and $(\partial\Omega_2) \cap \Omega_1$, is not exact but contains some error, then in general there will be a pressure singularity at each reentrant corner. The strength of the singularity will be directly related to the amount of error in the boundary specification.

6. The Finite Difference Method for the Stokes Equations

The finite difference methods used in the numerical experiments with the overlapping domain method are variations of the regularized difference schemes introduced by the first author [10]. This class of schemes is the only one for which published results are given to demonstrate the second-order accuracy of the solution in non-simple domains.

In cartesian coordinates the Stokes equations are

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial p}{\partial x} = 0 \tag{6.1}$$

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} - \frac{\partial p}{\partial y} = 0 \tag{6.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{6.3}$$

and in polar coordinates they are

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} - \frac{u}{r^2} + \frac{2}{r^2}\frac{\partial v}{\partial \theta} - \frac{\partial p}{\partial r} = 0 \tag{6.4}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 v}{\partial \theta^2} - \frac{v}{r^2} - \frac{2}{r^2}\frac{\partial u}{\partial \theta} - \frac{1}{r}\frac{\partial p}{\partial \theta} = 0 \tag{6.5}$$

$$\frac{1}{r}\frac{\partial (ru)}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial \theta} = 0. \tag{6.6}$$

In (6.1), (6.2), and (6.3) the variables (u, v) denote the velocity components in the cartesian representation, while in (6.4), (6.5), and (6.6) they denote the components in the polar representation. No confusion should result from not distinguishing between the choice of representation since the two coordinate systems are treated separately.

The first test problem is on the domain consisting of the rectangle $-1 \leq x \leq 1, 0 \leq y \leq 1$, with a right triangle with legs of length 0.8 and 0.6 attached to the rectangle along the hypotenuse at $-0.5 \leq x \leq 0.5$, see Figure 2. The two grids used to compute the solution were each cartesian grids. On each domain the equations have the form (6.1)-(6.3), but the velocity components on one grid must be rotated by the angle of the tilt between the grids to obtain the velocity components on the other grid.

The region for which the second set of numerical experiments were run is the rectangle $-1 \leq x \leq 1, 0 \leq y \leq 1$ with the exclusion of a circle of radius $r_{\min}$ centered at the origin. This region is displayed in Figure 3. The grid system consisted of an evenly spaced cartesian grid with equal spacing in each direction and a polar grid with uniform spacing in each of the radial and angular directions.

The domain covered by the cartesian grid was the large rectangle with the exclusion of a smaller rectangle around the excluded circle as displayed in Figure 3. The grid spacings on this, the cartesian domain, were

given by

$$\Delta x = \Delta y = 1/M \tag{6.7}$$

for some positive integer M. The omitted grid points were those in the rectangle

$$-i_0\Delta x \ < \ x \ < \ i_0\Delta x, \quad 0 \ \leq \ y \ < \ i_0\Delta y. \tag{6.8}$$

The domain covered by the polar grid was the region displayed in the lower right of Figure 3, given by

$$r_{\min} \ \leq \ r \ \leq \ r_{\max}, \quad 0 \ \leq \ \theta \ \leq \ \pi. \tag{6.9}$$

Various values of $r_{\min}$ and $r_{\max}$ were used in the studies. Typical values were 0.2 for $r_{\min}$ and 0.9 for $r_{\max}$. The grid spacings for this domain, the polar domain, were given by

$$\Delta r = (r_{\max} - r_{\min})/N \quad \text{and} \quad \Delta\theta = \pi/J \tag{6.10}$$

for some positive integers N and J.

In all the tests an analytical solution of the Stokes equations was used to assess the accuracy of the method. The first problem used Dirichlet boundary conditions, in which the velocity was specified along the boundary.

For the second problem the solutions which were used were symmetric about the x axis. The boundary conditions imposed were that the velocity was specified along the boundaries given by $x = -1$, $x = 1$, $y = 1$, and also $r = r_{\min}$. Along the boundary $y = 0$ symmetry conditions were used. These were

$$\frac{\partial u}{\partial y} = 0 \quad \text{and} \quad v = 0.$$

For the cartesian grid the finite difference method was based on second-order accurate central differences for all derivatives. However, the use of the standard central difference for the divergence equation and for the pressure gradient results in non-smooth solutions. The loss of smoothness can be avoided by using regularized central differences as introduced in [10], see also [11]. These approximations are

$$\frac{\partial p}{\partial x} \simeq \delta_{x,r}p = \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} - \frac{p_{i+2,j} - 3p_{i+1,j} + 3p_{i,j} - p_{i-1,j}}{6\Delta x} \tag{6.11}$$

$$\frac{\partial p}{\partial y} \simeq \delta_{y,r}p = \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y} - \frac{p_{i,j+2} - 3p_{i,j+1} + 3p_{i,j} - p_{i,j-1}}{6\Delta y} \tag{6.12}$$

$$\frac{\partial u}{\partial x} \simeq \delta_{x,r}u = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} - \frac{u_{i+1,j} - 3u_{i,j} + 3u_{i-1,j} - u_{i-2,j}}{6\Delta x} \tag{6.13}$$

$$\frac{\partial v}{\partial y} \simeq \delta_{y,r}v = \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} - \frac{v_{i,j+1} - 3v_{i,j} + 3v_{i,j-1} - v_{i,j-2}}{6\Delta y}. \tag{6.14}$$

Notice that the additional regularizing terms are third-order divided differences which are shifted forward for the pressure and shifted backward for the velocity derivatives. At grid points near the boundaries, where the approximations (6.11)-(6.14) can not be applied, the third-order difference is shifted the other way.

For the polar grid of the second problem, a method was employed which uses both finite differences and Fourier methods. The solutions were assumed to be symmetric about the line $y = 0$, and thus the components of the velocity in polar representation and the pressure can be expressed as

$$u_{ij} = u(r_i, \theta_j) = \frac{1}{2}a_0(r_i) + \sum_{k=1}^{J-1} a_k(r_i) \cos k\theta_j + \frac{1}{2}(-1)^j a_J(r_i) \tag{6.15}$$

$$v_{ij} = v(r_i, \theta_j) = \sum_{k=1}^{J-1} b_k(r_i) \sin k\theta_j \tag{6.16}$$

$$p_{ij} = p(r_i, \theta_j) = \frac{1}{2}c_0(r_i) + \sum_{k=1}^{J-1} c_k(r_i) \cos k\theta_j + \frac{1}{2}(-1)^j c_J(r_i). \tag{6.17}$$

In these formulas $r_i = r_{\min} + i \cdot \Delta r$ and $\theta_j = j\Delta\theta$. The formulas (6.15), (6.16), and (6.17) express the function in terms of their finite Fourier series for a fixed value of r_i. The determination of the coefficients a_k, b_k, and c_k are easily determined from the function values. Because of the assumed symmetry in the solution, u and p are even functions in θ and v is an odd function.

Approximations to the derivatives with respect to θ are obtained by regarding the variable θ as a continuously varying variable on the right-hand side of (6.15), (6.16), and (6.17). For example,

$$\frac{\partial v}{\partial \theta}(r_i, \theta_j) \simeq \sum_{k=1}^{J-1} b_k(r_i) k \cos k\theta_j. \tag{6.18}$$

The finite Fourier representation is used only to obtain the approximations to the derivatives with respect to θ. The derivatives are used to evaluate the left-hand side of equations (6.4), (6.5) and (6.6) as part of the iterative solution technique.

The importance of using regularized differences also applies to the Fourier approximation. In this case, the coefficients $a_J(r_i)$ and $c_J(r_i)$ must be constrained (see also [12]). These coefficients are related through equation (6.4), but some relation other than given by the equations (6.4), (6.5), and (6.6) is needed to determine these values. To control the magnitude of the coefficients $c_J(r_i)$ in (6.17) the approximation of some derivative must be altered. Note that the derivative of p with respect to θ occurs only in equation (6.5) which relates the coefficients of p, i.e., the $c_k(r_i)$, with those of v, i.e., the $b_k(r_i)$, but there is no mode for k equal to J in the representation for v. It would be possible to allow for a coefficient $b_J(r_i)$ in the representation for v. This coefficient $b_J(r_i)$ would be proportional to $c_J(r_i)$ and could be constrained through equation (6.6). Note that since $\sin J\theta_j$ is identically zero, this coefficient would have no direct effect

on v. Equivalently, $b_J(r_i)$ was not explicitly used, instead a term equal to

$$-\frac{1}{4}(-1)^j c_J(r_i)$$

was added to the approximation of the divergence equation (6.6). The form of the iterative method then forced the coefficient $c_J(r_i)$ to decrease. The rate of decrease is $1-\gamma/2$, where γ is the iterative parameter in (4.2). Some such constraint on the coefficients $c_J(r_i)$ is necessary; without it, the dominant pressure errors are in the coefficient $c_J(r_i)$. Since this coefficient relates to the highest frequency supported by the grid there is no concern with consistency in treating this term.

7. The Numerical Experiments

The first set of numerical experiments were run using the domain shown in Figure 2. On each domain the grid is a cartesian grid. For the cases shown in Table 1, the grid spacing was $1/M$ in both directions for the larger grid and was $.8/M$ for Δx and $.6/M$ for Δy on the smaller tilted grid. On each domain the Stokes equations have the form (6.1)-(6.3), but the components of the velocity vectors are related by the rotation of the coordinate system. After interpolating the velocity vector components from one grid to the other, the components in the new coordinate system have to computed using the appropriate rotation.

The second set of numerical experiments were run using the domain shown in Figure 3. The grid was generated by constructing a cartesian grid on the domain shown in the lower left of Figure 3, and a polar grid on the domain shown in the lower right of Figure 3. In the top domain shown in Figure 3, the dotted lines show the overlap betwee the two domains. In almost all the cases tested Δx and Δy were equal and i_0 and j_0 were also equal.

Several exact solutions were used to test the numerical method. A very useful solution was, in the cartesian representation,

$$u = x, \qquad v = -y, \qquad p = 0, \tag{7.1}$$

or, in polar representation,

$$u = r\cos 2\theta, \qquad v = -r\sin 2\theta, \qquad p = 0. \tag{7.2}$$

This solution has the property that it exactly satisfies the difference approximations to the Stokes equations on both grids of the second problem. Thus the only errors introduced by the method were due to the interpolation. With the use of the finite Fourier expansion for the interpolation from the polar to the cartesian domain, this simple exact solution had essentially no error. The only errors measured were on the order of the arithmetic accuracy of the computer.

Table 1

	Main grid errors				Tilted grid errors			
M	u	v	p	d	u	v	p	d
10	5.43(−4)	4.29(−4)	1.64(−2)	1.7(−3)	5.83(−4)	2.17(−4)	1.28(−2)	3.6(−5)
16	1.88(−4)	1.58(−4)	4.97(−3)	2.7(−4)	1.72(−4)	9.20(−5)	5.09(−3)	−1.3(−6)
24	6.03(−5)	5.26(−5)	1.69(−3)	6.5(−5)	5.06(−5)	2.74(−5)	2.00(−3)	−4.2(−7)
30	3.44(−5)	3.29(−5)	9.48(−4)	3.0(−5)	2.96(−5)	1.71(−5)	1.19(−3)	−1.6(−7)

The solution for which most of the tests of the first problem were made is,

$$u = \frac{1}{2}\left((x+0.5)^2-(y+0.5)^2\right)\Big/\left((x+0.5)^2+(y+0.5)^2\right)$$

$$-\frac{1}{2}\log\left((x+0.5)^2+(y+0.5)^2\right)$$

$$v = (x+0.5)(y+0.5)\Big/\left((x+0.5)^2+(y+0.5)^2\right)$$

$$p = 2(x+0.5)\Big/\left((x+0.5)^2+(y+0.5)^2\right)+p_0,$$

in cartesian representation. In Table 1 are shown the L^2 norms of the errors for several cases. For these cases, the iteration parameters ω and γ were chosen as $\omega = 2/(1+4\Delta x)$ and $\gamma = 4\Delta x$ where Δx is the x grid spacing on the larger grid. Table 1 also shows the values of d_i, the average divergence, for each domain, see equation (2.4). The values of d_i are on the order of the error for the computed solution, and thus are acceptable.

Table 2 gives the convergence rates for successive runs. The convergence rate for runs with values of M equal to M_1 and M_2, with M_1 less than M_2, is

$$\log\left(\text{error}\,(M_1)/\text{error}\,(M_2)\right)/\log\left(M_2/M_1\right).$$

All convergence rates are close to 2.0 indicating second-order accuracy of the overall method.

Table 2

	Main grid errors			Tilted grid errors		
M_1/M_2	u	v	p	u	v	p
10/16	2.26	2.13	2.54	2.60	1.83	1.96
16/24	2.80	2.71	2.66	3.02	2.99	2.30
24/30	2.52	2.10	2.59	2.40	2.11	2.33

The solution for which most of the tests of the second problem were made is,

$$u = \frac{1}{2}\left(x^2 - y^2\right)/\left(x^2 + y^2\right) - \frac{1}{2}\log\left(x^2 + y^2\right)$$

$$v = xy/\left(x^2 + y^2\right)$$

$$p = 2x/\left(x^2 + y^2\right) + p_0,$$

in cartesian representation. In Table 3 are shown the L^2 norms of the errors for a representative series of runs. For these runs M and N, as defined by (6.7) and (6.10), were equal. The values of $r_{\min}$ and $r_{\max}$ were 0.2 and 0.9, respectively. The value J was 14 for all runs. For these cases, the iteration parameters ω and γ were chosen as $\omega = 2/(1+9\Delta x)$ and $\gamma = 9\Delta x$ where Δx is the x grid spacing on the cartesian grid.

Table 3

	Cartesian errors				Polar errors			
M	u	v	p	d	u	v	p	d
10	5.31(−3)	4.73(−3)	2.28(−1)	−3.1(−4)	2.50(−3)	1.12(−2)	2.17(−1)	2.8(−4)
20	1.21(−3)	9.47(−4)	2.19(−2)	−2.5(−5)	8.19(−4)	2.79(−3)	4.83(−2)	3.3(−5)
30	5.47(−4)	4.31(−4)	7.81(−3)	−8.9(−6)	3.99(−4)	1.22(−3)	2.11(−2)	1.3(−5)
40	3.12(−4)	2.45(−4)	4.60(−3)	−2.4(−6)	2.25(−4)	6.99(−4)	1.22(−2)	7.0(−6)

Table 4 displays the convergence rates for successive runs. The convergence rates were computed as for Table 2. All convergence rates are, again, close to 2.0 indicating second-order accuracy of the overall method. The worst convergence rate in Table 4 is for the radial velocity component on the polar grid. Since this quantity has the lowest overall error, see Table 3, it is not considered significant that its convergence rate is so low. It must be remembered in assessing the accuracy of methods for systems of equations, such as the Stokes equations, that the accuracy of all of the components are interconnected. Hence, it is difficult to give one number to express the overall error or the overall accuracy.

Because the norms of the pressure error in Table 3 are larger than the norms of the velocity errors, it would seem that the pressure is determined with less accuracy than is the velocity, Although this is probably correct, the numbers themselves are somewhat misleading because the norm of the pressure is larger than the norm of the velocity. The numbers in Table 4, giving the order of accuracy, are the more significant quantities.

The pressure error norms are computed in a way similar to the computation of a statistical variance, that is, the pressure error norm measures the deviation of the pressure from its mean value. The algorithm

Table 4						
	Cartesian errors			Polar errors		
M_1/M_2	u	v	p	u	v	p
10/20	2.13	2.32	3.38	1.61	2.01	2.17
20/30	1.96	1.94	2.54	1.77	2.04	2.05
20/40	1.96	1.94	2.25	1.86	1.97	1.99
30/40	1.95	1.96	1.84	1.99	1.94	1.90

that was used is due to West [18]. In this way, the effect of an additive constant is canceled. The norm of the difference between the divergence and d_i was also computed using this algorithm. As used here, d_i corresponds to the statistical average, and the norm of the difference is obtained as the standard deviation.

The velocity field for the run given in Table 3 with M equal to 20 is shown in Figure 5, and in Figure 6 are shown the pressure contours. As shown by Fig. 5 the solution provides a good test problem by being nontrivial. Figure 6 was obtained by overlaying the contour plots from each of the domains.

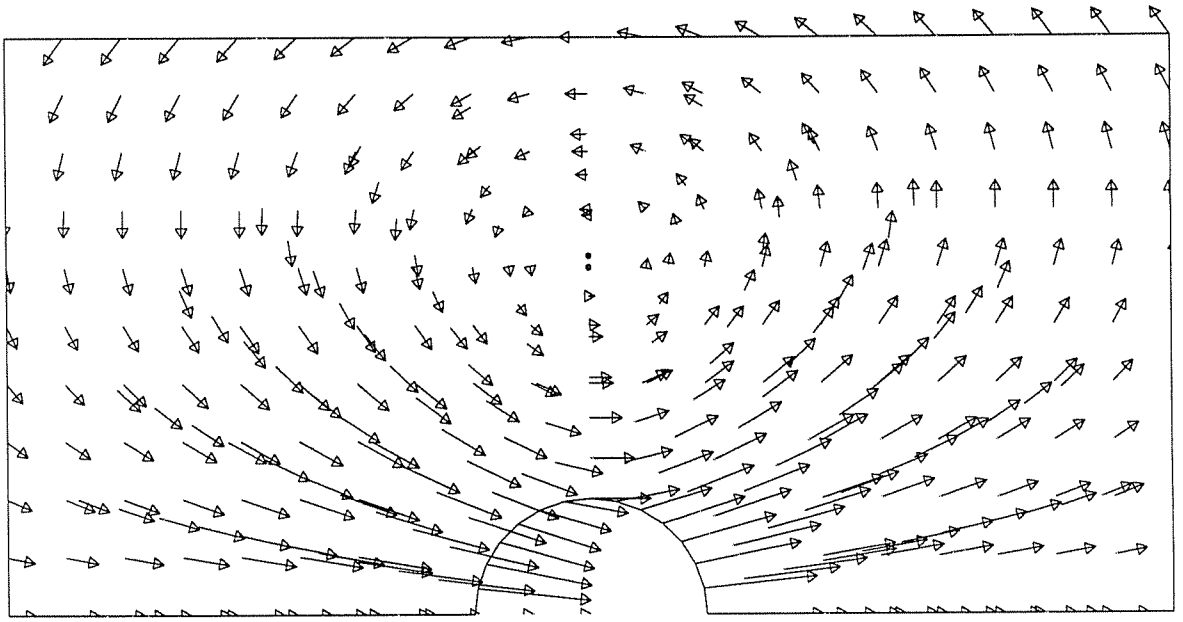

Figure 5

The stopping criterion for the iterative methods was to stop when the norms of the changes in the velocities and the norm of the deviation between the divergence of the velocity field and d_i were all sufficiently small. The number of iterations for the cases shown in Table 1 were 179, 327, 527, and 677, respectively, when the L^2 norms of the updates were less than 10^{-5}.

As mentioned in section 2 the size of the overlap between the domains was related to the conditioning of

the problem. For example if the value of r_{max} was decreased from 0.9 to 0.6 the errors increased by about 50 percent. The efficiency was also affected by decreasing r_{max}. The number of iterations required to achieve a given convergence criterion was more than four times as great for r_{max} of 0.6 as for r_{max} of 0.9.

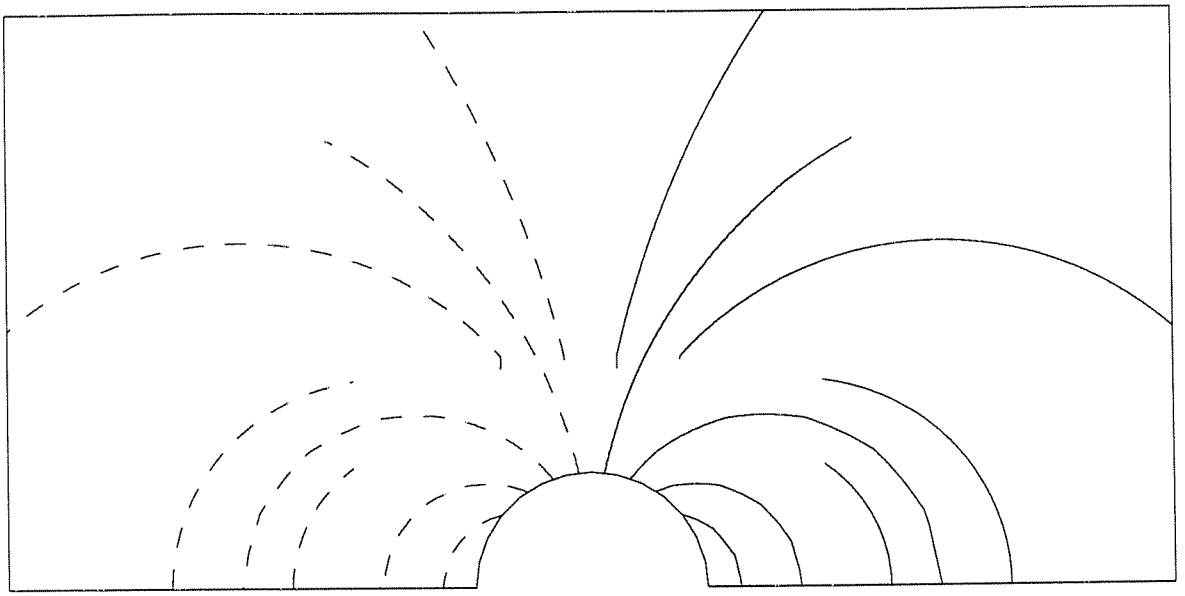

Figure 6

8. Conclusions

The domain decomposition method presented here has been demonstrated to compute accurate solutions to the equations describing incompressible viscous flow. The method described here can be extended to compute flows of engineering significance involving the Navier-Stokes equations. It is anticipated that there will be few difficulties encountered in extending this method from the Stokes equations to the Navier-Stokes equations for low Reynolds number flows.

REFERENCES

[1] J. Cahouet, *On some difficulties occurring in the simulation of incompressible fluid flows by domain decomposition methods, in Proc. First Intern. Symp. on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, Phila. PA, 1988, pp. 313-332.

[2] G. S. Chesshire, *Composite grid construction and applications*, Ph.D. thesis, Dept. of Applied Math., California Institute of Technology, Pasadena, CA, 1986.

[3] Q. V. Dinh, R. Glowinski, J. Périaux, and G. Terrasson, *On the coupling of viscous and inviscid models for incompressible fluid flows via domain decomposition methods, in Proc. First Intern. Symp. on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, Phila. PA, 1988, pp. 350-369.

[4] M. Fortin and R. Aboulaich, *Schwarz's decomposition method for incompressible flow problems, in Proc. First Intern.*

Symp. on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, Phila. PA, 1988, pp. 333-349.

[5] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., *Proc. First Intern. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Phila. PA, 1988.

[6] W. D. Henshaw and G. Chesshire, *Multigrid on Composite Meshes*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 914-923.

[7] B. Kreiss, *Construction of curvilinear grids*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 270-279.

[8] P. L. Lions, *On the Schwarz alternating method. I, in Proc. First Intern. Symp. on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, Phila. PA, 1988, pp. 1-42.

[9] H. J. Lugt and E. W. Schwiderski, *Flow around dihedral angles I, eigenfunction analysis*, Proc. Roy. Soc. Lond. A, 285 (1965), pp. 382-399.

[10] J. C. Strikwerda, *Finite difference methods for the Stokes and Navier-Stokes equations*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 56-68.

[11] J. C. Strikwerda, *An iterative method for solving finite difference approximations to the Stokes equations*, SIAM J. Numer. Anal., 21 (1984), pp. 447-458.

[12] J. C. Strikwerda and Y. M. Nagel, *A numerical method for the incompressible Navier-Stokes equations in three-dimensional cylindrical geometry*, J. Comp. Phys., 78 (1988), pp. 64-78.

[13] J. C. Strikwerda, *A multigrid method for incompressible viscous flow*, in preparation.

[14] J. C. Strikwerda, B. A. Wade, K. P. Bube, *Regularity estimates up to the boundary for elliptic systems of difference equations*, SIAM J. Numer. Anal. (1990 to appear).

[15] Thompson, J. F., ed., *Numerical Grid Generation*, Elsevier/North Holland, New York, NY, 1982..

[16] M. C. Thompson and J. H. Ferziger, *An adaptive multigrid technique for the incompressible Navier-Stokes equations*, J. Comp. Phys., 82 (1989), pp. 94-121.

[17] S. P. Vanka, *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comp. Phys., 65 (1986), pp. 138-158.

[18] D. H. D. West, *Updating mean and variance estimates: an improved method*, Comm. ACM, 22 (1979), pp. 532-535.