JOURNAL OF COMPUTATIONAL PHYSICS 118, 348–355 (1995)

A Fast Poisson Solver for Complex Geometries

A. McKENNEY AND L. GREENGARD*

Courant Institute of Mathematical Sciences, New York University, New York, New York 10012

AND

A. MAYO

IBM, Thomas J. Watson Research Center, Yorktown Heights, New York 10598

Received July 29, 1994

Robust fast solvers for the Poisson equation have generally been limited to regular geometries, where direct methods, based on Fourier analysis or cyclic reduction, and multigrid methods can be used. While multigrid methods can be applied in irregular domains (and to a broader class of partial differential equations), they are difficult to implement in a robust fashion, since they require an appropriate hierarchy of coarse grids, which are not provided in many practical situations. In this paper, we present a new fast Poisson solver based on potential theory rather than on direct discretization of the partial differential equation. Our method combines fast algorithms for computing volume integrals and evaluating layer potentials on a grid with a fast multipole accelerated integral equation solver. The amount of work required is $O(m \log m + N)$, where m is the number of interior grid points and N is the number of points on the boundary. Asymptotically, the cost of our method is just twice that of a standard Poisson solver on a rectangular domain in which the problem domain can be embedded, independent of the complexity of the geometry.

1. INTRODUCTION

A longstanding problem in numerical analysis has been the development and implementation of a robust fast solver for the Poisson equation in complicated domains. In this paper, we present such a solver based on potential theory rather than on direct discretization of the partial differential equation. Our method combines fast algorithms for computing volume integrals [6] and evaluating layer potentials on a grid [5] with the integral equation approach of [3] and the adaptive fast multipole method [2]. Direct methods, based on Fourier analysis, cyclic reduction, or both, require regular geometries. Standard iterative procedures converge slowly. Multigrid methods, which achieve optimal efficiency in theory, require an appropriate hierarchy of coarse grids, which are not provided in many practical situations. Capacitance matrix methods, which are more closely related to the method proposed here, require the solution of a (dense) capacitance matrix rather than a boundary integral equation and cannot take advantage of fast adaptive multipole methods and highly accurate quadrature formulas. Before describing our approach, however, let us begin by describing the target problem clearly. We are interested in solving the Dirichlet problem

$$\Delta U(\mathbf{x}) = \rho(\mathbf{x}) \quad \text{in} \quad \mathcal{D} \tag{1}$$

$$U(\mathbf{x}) = f(\mathbf{x}) \quad \text{on} \quad \partial\mathcal{D}, \tag{2}$$

where $\mathcal{D}$ is either an *interior* or *exterior* domain with boundary $\partial\mathcal{D}$ in $\mathbf{R}^2$. The boundary is assumed to be smooth, but it may consist of many components; i.e., the domain is allowed to be multiply-connected. For the exterior problem, the inhomogeneous term $\rho(\mathbf{x})$ is assumed to have compact support. Our objective is to provide a tool which takes as input a simple specification of the necessary data and which returns the desired solution in nearly optimal order time without further assistance.

Remark. For interior problems, $\partial\mathcal{D}$ consists of an outer boundary, which we will denote by $\partial\mathcal{D}_0$, as well as a (possibly empty) collection of interior boundary curves $\partial\mathcal{D}_1$, ..., $\partial\mathcal{D}_M$. For exterior problems, $\partial\mathcal{D}$ consists of a collection of boundary curves $\partial\mathcal{D}_1$, ..., $\partial\mathcal{D}_M$.

The user is required to provide the following:

Interior Problems. 1. N_k points in the discretization of each boundary component $\partial\mathcal{D}_k$ for $k = 0, ..., M$.

- The Dirichlet data $f(\mathbf{x})$ at each of the boundary points.
- The inhomogeneous data $\rho(\mathbf{x})$ at each of the boundary points.

* The first author was supported by the Office of Naval Research under Contract N00014-91-J-1312. The second author was supported by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract DEFGO288ER25053, by a NSF Presidential Young Investigator Award, and by a Packard Foundation Fellowship.

0021-9991/95 $6.00

2. A box $\mathcal{B}$ which encloses $\mathcal{D}$.

- The inhomogeneous data $\rho(\mathbf{x})$ at those points of a uniform $m \times m$ grid on $\mathcal{B}$ which lie inside $\mathcal{D}$.

Exterior Problems. 1. N_k points in the discretization of each boundary component $\partial\mathcal{D}_k$ for $k = 1, \ldots, M$.

- The Dirichlet data $f(\mathbf{x})$ at each of the boundary points.
- The inhomogeneous data $\rho(\mathbf{x})$ at each of the boundary points.

2. A box $\mathcal{B}$ which encloses both the support of $\rho(\mathbf{x})$ and the boundary $\partial\mathcal{D}$, as well as the region of space on which the solution is desired.

- The inhomogeneous data $\rho(\mathbf{x})$ at those points of a uniform $m \times m$ grid on $\mathcal{B}$ which lie in $\mathcal{D}$.

3. The condition at infinity specified in terms of a real number C. If $C \neq 0$, then a solution $U(\mathbf{x})$ will be computed which satisfies $U(\mathbf{x}) \to C \log |\mathbf{x}|$ as $|\mathbf{x}| \to \infty$. If $C = 0$, then a solution $U(\mathbf{x})$ will be computed which is bounded as $|\mathbf{x}| \to \infty$.

On completion, the algorithm returns the value of the solution at those points of the $m \times m$ grid which lie inside $\mathcal{D}$.

By uncoupling the discretization of $\partial\mathcal{D}$ from the discretization of $\mathcal{D}$ itself, it is easy for the user to describe complex geometries. One can achieve extremely high resolution of the boundary without complicated triangulations and data structures. Furthermore, the use of boundary integral equations allows for the straightforward calculation of derivative quantities such as the Dirichlet–Neumann map (the evaluation of $\partial U/\partial n$, given Dirichlet data for U).

Our approach to solving the Poisson equation is based on a standard potential theory decomposition, namely the construction of a *particular solution* V satisfying

$$\Delta V(\mathbf{x}) = \rho(\mathbf{x}) \quad \text{in } \mathcal{D}, \tag{3}$$

followed by the solution of a new Dirichlet problem now governed by the Laplace equation

$$\begin{aligned} \Delta W(\mathbf{x}) &= 0 \quad \text{in } \mathcal{D} \\ W(\mathbf{x}) &= f(\mathbf{x}) - V(\mathbf{x}) \quad \text{on } \partial\mathcal{D}. \end{aligned} \tag{4}$$

The desired solution is then given by $U = V + W$.

The paper proceeds as follows: A fast algorithm for computing a particular solution V is briefly described in Section 2, the integral equation approach to solving (4) is reviewed in Section 3, and a fast algorithm for extending the function W to the entire domain is discussed in Section 4. Numerical results are presented in Section 5 and some concluding remarks are collected in Section 6.

2. THE RAPID EVALUATION OF A VOLUME INTEGRAL

An obvious particular solution V satisfying

$$\Delta V(\mathbf{x}) = \rho(\mathbf{x}) \quad \text{in } \mathcal{D}$$

is obtained by convolution with the free-space Green's function,

$$V(\mathbf{x}) = \frac{1}{2\pi} \int_{\mathcal{D}} \log |\mathbf{x} - \mathbf{y}| \rho(\mathbf{y})\, d\mathbf{y}.$$

With the use of appropriate quadrature rules, this integral can be computed at m^2 points in $O(m^2)$ time, using the fast multipole method. Although such an approach would be very robust, the corresponding constant is large and the method is not competitive with the approach described below [6] which yields a different particular solution.

2.1. *Notation*

Recall that our method uses two distinct grids: a regular lattice on $\mathcal{B}$ which covers the domain $\mathcal{D}$, which we refer to as the "volume grid" and whose points we refer to as "lattice points," and a separate discretization of the boundary, which we refer to as the "boundary grid" and whose points we refer to as "boundary points."

The volume grid is assumed to have sides parallel to the x and y axes and the point spacing h is assumed to be the same in both the x and y directions. The N_k boundary points on each curve $\partial\mathcal{D}_i$ are assumed to be equispaced in arclength.

The discrete (5-point) Laplacian is defined on the lattice points by

$$\Delta_h V_{ij} = (V_{i,j+1} + V_{ij-1} + V_{i+1,j} + V_{i-1,j} - 4V_{ij})/h^2, \tag{5}$$

where the $\{V_{ij}\}$ are the values of a function defined on the lattice points $\{(x_i, y_j)\}$. A grid point $(x_i, y_j) \in \mathcal{D}$ is called a *regular* grid point if the formula for Δ_h involves points entirely inside $\mathcal{D}$. Similarly, a grid point $(x_i, y_i) \in \mathcal{B}\backslash\mathcal{D}$ is called a *regular* grid point if the formula for Δ_h involves points entirely inside $\mathcal{B}\backslash\mathcal{D}$. All other grid points are called *irregular* points.

If $p = (x_i, y_j)$ is a grid point, then its four nearest neighbors are denoted by $p_N = (x_i, y_{j+1})$, $p_S = (x_i, y_{j-1})$, $p_E = (x_{i+1}, y_j)$, and $p_W = (x_{i-1}, y_j)$. If the boundary $\partial\mathcal{D}$ crosses the segment $[p, p_k]$, $k = N, S, E, W$, then the point of intersection will be denoted by $\tilde{p}_k$ and the distance from p to $\tilde{p}_k$ by h_k. Finally, the jump in the quantity U as one crosses $\tilde{p}_k$ in passing from p to one of its neighbors p_k, $k = N, S, E, W$, is denoted by $[U]_k$.

2.2. *Interior Problems*

Consider the following Dirichlet problem:

$$\Delta V = \begin{cases} \rho & \text{on } \mathcal{D} \\ 0 & \text{on } \mathcal{B}\backslash\mathcal{D} \end{cases} \qquad (6)$$
$$V = 0 \quad \text{on } \partial\mathcal{B}.$$

Solving

$$\Delta_h \tilde{V}_{i,j} = \begin{cases} \rho_{i,j} & \text{if } (x_i, y_j) \in \mathcal{B} \\ 0 & \text{if } (x_i, y_j) \in \mathcal{B}\backslash\mathcal{D} \end{cases} \qquad (7)$$
$$\tilde{V}_{i,j} = 0 \quad \text{on } \partial\mathcal{B}$$

provides only a first-order accurate approximation to V. To see why, suppose that (x_i, y_j) is a regular grid point inside $\mathcal{D}$. Then

$$\Delta_h V = \Delta V + O(h^2) = \rho + O(h^2)$$

by a standard Taylor series expansion. At regular grid points outside $\mathcal{D}$,

$$\Delta_h V = \Delta V + O(h^2) = 0 + O(h^2),$$

again by a standard Taylor series expansion. At irregular grid points, however,

$$\Delta_h V = \Delta V + O(1)$$

because of the discontinuity in the Laplacian across the boundary. There is, therefore, an $O(1)$ error in the right-hand side of (7) at the irregular points. Since the number of irregular points is small compared to the total number of grid points, one can show that the global error is $O(h)$ and not $O(1)$. To achieve higher order accuracy, one must analyze the difference $\Delta_h V - \Delta V$ at the irregular grid points more carefully. In [6], it is shown that

$$\Delta_h V_{i,j} = \Delta V_{i,j} + C_{i,j},$$

where

$$C_{i,j} = \frac{1}{2h^2}\left(h_E^2[V_{xx}]_E + h_W^2[V_{xx}]_W + h_N^2[V_{yy}]_N + h_S^2[V_{yy}]_S\right) + O(h).$$

Note that not all jumps are nonzero. Higher order corrections are also computable, but the above is sufficient for second-order accuracy. Once we obtain the $C_{i,j}$, all we need to do is solve the linear system

$$\Delta_h \tilde{V}_{i,j} = \begin{cases} \rho_{i,j} & \text{if } (x_i, y_j) \text{ is a regular point in } \mathcal{D} \\ 0 & \text{if } (x_i, y_j) \text{ is a regular point in } \mathcal{B}\backslash\mathcal{D} \\ \rho_{ij} + C_{i,j} & \text{if } (x_i, y_j) \text{ is an irregular point in } \mathcal{D} \\ C_{i,j} & \text{if } (x_i, y_j) \text{ is an irregular point in } \mathcal{B}\backslash\mathcal{D} \end{cases} \qquad (8)$$
$$\tilde{V}_{i,j} = 0 \quad \text{on } \partial\mathcal{B}$$

by means of a standard method for inverting the discrete Laplacian on $\mathcal{B}$. In short, we first compute a right-hand side which corresponds to the discrete Laplacian of the function we want and then we apply a standard fast Poisson solver. At present, we use Buneman's method [1].

2.3. *Exterior Problems*

As in the preceding section, we begin by solving the Dirichlet problem:

$$\Delta V_1 = \begin{cases} \rho & \text{on } \mathcal{D} \\ 0 & \text{on } \mathcal{B}\backslash\mathcal{D} \end{cases} \qquad (9)$$
$$V_1 = 0 \quad \text{on } \partial\mathcal{B},$$

where $\mathcal{B}$ is the (user-provided) box which encloses the support of ρ. Further, let us extend V_1 to be identically zero outside $\partial\mathcal{B}$. Unfortunately, this function cannot play the role of a particular solution satisfying (3). The boundary $\partial\mathcal{B}$ is part of the problem domain and the particular solution we are seeking must be harmonic in its vicinity. V_1, on the other hand, has a jump in its normal derivative as one crosses $\partial\mathcal{D}$. Consider, however, the single layer potential

$$V_2(\mathbf{x}) = \frac{1}{2\pi}\int_{\mathcal{B}} \log|\mathbf{x} - \mathbf{y}|\sigma(\mathbf{y})\, d\mathbf{y},$$

where σ is an unspecified but continuous density function. Then V_2 is continuous in $\mathbf{R}^2$, harmonic in $\mathbf{R}^2\backslash\mathcal{B}$, and satisfies the jump relation

$$\left[\frac{\partial V_2}{\partial \nu}\right] = \sigma \quad \text{on } \partial\mathcal{B},$$

where $\partial/\partial\nu$ is the unit outward normal derivative. If we set $\sigma = -[\partial V_1/\partial n]$, then the function $V = V_1 + V_2$ is harmonic on both sides of $\partial\mathcal{B}$ and continuous across the boundary with continuous normal derivative. It is, therefore, harmonic in a neighborhood of the boundary and is a suitable particular solution.

To obtain the values of V_2 at all points inside $\mathcal{B}$, we first evaluate the single layer potential on the box boundary $\mathcal{B}$ by the fast multipole method and then extend the solution to the interior of the box by means of a standard fast solver.

3. THE SOLUTION OF LAPLACE'S EQUATION

We now consider the solution of the auxiliary Laplace equation (4), following the treatment of [3, 8]. Note, however, that we first require the modified Dirichlet data $\tilde{f} = f - V$ on the boundary. While the original data f is given, we must interpolate V from the volume grid to the boundary mesh. For this, we use a third-order accurate formula which takes into account the fact that the volume integral V has jumps in its second derivative across the boundary $\partial\mathcal{D}$ (see, for example, [6]).

In the case of a bounded domain $\mathcal{D}$ with M interior boundary curves $\partial\mathcal{D}_k$, we seek a solution in the form of a double layer potential combined with M singular sources,

$$W(\mathbf{x}) = \frac{1}{2\pi}\int_{\partial\mathcal{D}} \mu(y)\frac{\partial}{\partial \nu_y}\ln|\mathbf{y}-\mathbf{x}|\,d\mathbf{y} + \sum_{k=1}^{M} A_k \ln|\mathbf{x}-S_k|, \quad (10)$$

where S_k is a point inside $\partial\mathcal{D}_k$. Applying the standard jump relations for double layer potentials [8], we obtain the integral equation

$$\mu(\mathbf{y}_0) + \frac{1}{\pi}\int_{\partial\mathcal{D}} \mu(\mathbf{y})\frac{\partial}{\partial \nu_y}\ln|\mathbf{y}-\mathbf{y}_0|\,d\mathbf{y} + 2\sum_{k=1}^{M} A_k \ln|\mathbf{y}_0 - S_k| = 2\tilde{f}(\mathbf{y}_0). \quad (11)$$

We subject this system to the constraint equations

$$\int_{\partial\mathcal{D}_k} \mu(\mathbf{y})\,d\mathbf{y} = 0, \quad k = 1, \ldots, M. \quad (12)$$

Remark. The coefficients of the singular sources $A_1, \ldots, A_M$ in the representation (10) are, from a fluid dynamics viewpoint, the circulations around the boundary components $\partial\mathcal{D}_1, \ldots, \partial\mathcal{D}_M$. Without these coefficients, the integral equation (11) is singular in mutliply connected domains, but it can be shown that the system of Eqs. (11) and (12) is invertible. For a complete discussion, see [3, 8].

For exterior problems, we seek a solution in the form

$$W(\mathbf{x}) = \frac{1}{2\pi}\int_{\partial\mathcal{D}} \mu(\mathbf{y})\frac{\partial}{\partial \nu_y}\ln|\mathbf{y}-\mathbf{x}|\,d\mathbf{y} + \frac{1}{2\pi}\int_{\partial\mathcal{D}} \mu(\mathbf{y})\,d\mathbf{y} + \sum_{k=1}^{M} A_k \ln|\mathbf{x}-S_k|, \quad (13)$$

where S_k is a point inside the boundary curve $\mathcal{D}_k$. We require that

$$\sum_{k=1}^{M} A_k = C, \quad (14)$$

where C is the user-specified constant describing the desired behavior at ∞. If $C = 0$, the solution will be bounded at ∞. Otherwise, $W(\mathbf{x}) \to C \log|\mathbf{x}|$.

The corresponding integral equation is

$$-\mu(\mathbf{y}_0) + \frac{1}{\pi}\int_{\partial\mathcal{D}} \mu(\mathbf{y})\frac{\partial}{\partial \nu_y}\ln|\mathbf{y}-\mathbf{y}_0|\,d\mathbf{y} + \frac{1}{\pi}\int_{\partial\mathcal{D}} \mu(\mathbf{y})\,d\mathbf{y} + 2\sum_{k=1}^{M} A_k \ln|\mathbf{y}_0 - S_k| = 2f(\mathbf{y}_0), \quad (15)$$

which we subject to the constraints

$$\int_{\partial\mathcal{D}_k} \mu(\mathbf{y})\,d\mathbf{y} = 0, \quad k = 1, \ldots, M-1. \quad (16)$$

In order to solve the systems (11) and (12) or (14), (15), and (16), we use a Nyström algorithm based on the trapezoidal rule, since it achieves superalgebraic convergence for smooth data on smooth domains. In more detail, we assume that we are given N_k points equispaced in arclength on each boundary component $\partial\mathcal{D}_k$ and associate with each such point, denoted $\mathbf{y}_i^k$, an unknown density value μ_i^k. The step in arclength in the discretization of $\partial\mathcal{D}_k$ will be denoted $h_k = |\partial\mathcal{D}_k|/N_k$, where $|\partial\mathcal{D}_k|$ is the length of the curve $\partial\mathcal{D}_k$. The total number of discretization points is denoted by

$$N = \sum_{k=1}^{M} N_k \quad \text{for the exterior problem}$$

$$N = \sum_{k=0}^{M} N_k \quad \text{for the interior problem.}$$

We replace the integral operators in (11) and (12) by

$$\int_{\partial\mathcal{D}} \mu(\mathbf{y})\frac{\partial}{\partial \nu_y}\ln|\mathbf{y}-\mathbf{y}_0|\,d\mathbf{y} \approx \sum_{k=0}^{M} h_k \sum_{j=1}^{N_k} \frac{\partial}{\partial \nu_{y_j^k}}\ln|\mathbf{y}_j^k - \mathbf{y}_i^l|\mu_j^k$$

and the constraint equations in (12) and (16) by

$$\int_{\partial\mathcal{D}_k} \mu(\mathbf{y})\,d\mathbf{y} \approx \sum_{j=1}^{N_k} \mu_j^k h_k.$$

Remark. In the preceding linear systems, the diagonal terms of the form $(\partial/\partial\nu_{y_j^k})\ln|\mathbf{y}_j^k - \mathbf{y}_i^l|$ should be replaced by their analytical limit, $\kappa(\mathbf{y}_i^l)/2$, where $\kappa(\mathbf{y}_i^l)$ denotes the curvature at the point $\mathbf{y}_i^l$.

The discrete equations for the interior and exterior problems may be written in block form as

$$\begin{pmatrix} I + K^i & B^i \\ C^i & D^i \end{pmatrix}\begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} 2\mathbf{f} \\ 0 \end{pmatrix}, \quad (17)$$

TABLE I

Computational Results for Example 1

m_x	m_y	N	K	E_∞	E_2	T	T_g	T_v	T_b	T_r	T_d
32	32	240	10	2.2×10^{-2}	7.9×10^{-3}	37	*	*	26	10	*
64	64	480	10	5.7×10^{-3}	1.9×10^{-3}	64	*	1	36	25	1
128	128	960	10	1.2×10^{-3}	4.6×10^{-4}	106	1	3	60	37	3
256	256	1920	10	3.1×10^{-4}	1.1×10^{-4}	202	5	11	101	64	11
512	512	3840	10	7.5×10^{-5}	2.7×10^{-5}	456	18	50	175	124	50
1024	1024	7680	10	2.1×10^{-5}	7.5×10^{-6}	1309	69	219	385	253	217

Note. Times marked by an asterisk (*) were negligible.

$$\begin{pmatrix} I - K^e & B^e \\ C^e & D^e \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} -2\mathbf{f} \\ 0 \end{pmatrix}, \tag{18}$$

where, for the interior problem,

$$\boldsymbol{\mu} = (\mu_1^0, \ldots, \mu_{N_0}^0, \mu_1^1, \ldots, \mu_{N_1}^1, \ldots, \mu_1^M, \ldots, \mu_{N_M}^M)^{\mathrm{T}}$$

is the vector of unknown density values,

$$\mathbf{a} = (A_1, \ldots, A_M)^{\mathrm{T}}$$

is the vector of unknown coefficients, and

$$\mathbf{f} = (f_1^0, \ldots, f_{N_0}^0, f_1^1, \ldots, f_{N_1}^1, \ldots, f_1^M, \ldots, f_{N_M}^M)^{\mathrm{T}}$$

is the vector of given boundary values. For the exterior problem,

$$\boldsymbol{\mu} = (\mu_1^1, \ldots, \mu_{N_1}^1, \ldots, \mu_1^M, \ldots, \mu_{N_M}^M)^{\mathrm{T}}$$

and

$$\mathbf{f} = (f_1^1, \ldots, f_{N_1}^1, \ldots, f_1^M, \ldots, f_{N_M}^M)^{\mathrm{T}}.$$

The matrices K^i and K^e in (17) and (18) represent the integral operators, the matrices B^i and B^e represent the logarithmic terms coupling the density values to the coefficients $A_1, \ldots, A_M$, and the matrices C^i, C^e, D^i, and D^e represent the constraint equations.

The linear systems (17) and (18) are solved iteratively, using the CGS algorithm [9] in preconditioned form:

$$\begin{pmatrix} I & B^i \\ C^i & D^i \end{pmatrix}^{-1} \begin{pmatrix} I + K^i & B^i \\ C^i & D^i \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} I & B^i \\ C^i & D^i \end{pmatrix}^{-1} \begin{pmatrix} 2\mathbf{f} \\ 0 \end{pmatrix}, \tag{19}$$

and

$$\begin{pmatrix} I & B^e \\ C^e & D^e \end{pmatrix}^{-1} \begin{pmatrix} I - K^e & B^e \\ C^e & D^e \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} I & B^e \\ C^e & D^e \end{pmatrix}^{-1} \begin{pmatrix} -2\mathbf{f} \\ 0 \end{pmatrix}. \tag{20}$$

The bulk of the work at each iteration lies in applying the full matrix to a vector. The product

$$\begin{pmatrix} I \pm K & B \\ C & D \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{pmatrix}$$

can be computed in time $O(N + M)$ using the adaptive fast multipole method [2, 10].

Remark. In actual practice, we solve the interior problem by introducing an additional source term,

$$A_0 \ln|\mathbf{x} - S_0|,$$

where S_0 lies outside the outer boundary $\partial\mathcal{D}_0$, and an additional constraint

$$\sum_{k=0}^{M} A_k = 0.$$

(This has no effect on the original integral equation, but is necessary to ensure the invertibility of the preconditioner.)

4. EVALUATING THE DOUBLE LAYER POTENTIAL ON THE GRID

Once we have obtained a solution to Eq. (19) or (20), we need to evaluate the double layer potential W at all grid points in $\mathcal{B}$. We do this using the method developed in [5], which is closely related to the technique described in Section 2. Specifically, let

TABLE II

Computational Results for Example 2

m_x	m_y	N	K	E_∞	E_2	T	T_g	T_v	T_b	T_r	T_d
64	64	986	23	1.4×10^{-2}	3.8×10^{-3}	222	1	1	158	61	1
128	128	2000	23	3.3×10^{-3}	8.5×10^{-4}	414	2	4	308	93	4
256	256	4000	23	7.4×10^{-4}	1.5×10^{-4}	759	9	19	557	148	19
512	512	8000	23	1.9×10^{-4}	4.4×10^{-5}	1529	31	86	1029	268	85

$$W_{\mathcal{D}} = \begin{cases} W & \text{in } \mathcal{D} \\ 0 & \text{in } \mathcal{B}\backslash\mathcal{D}. \end{cases}$$

Observe now that at regular grid points in $\mathcal{B}$,

$$\Delta_h W_{\mathcal{D}} = \Delta W_{\mathcal{D}} + O(h^2) = 0 + O(h^2).$$

The discrete Laplacian of $W_{\mathcal{D}}$ at irregular points can, of course, be obtained directly from the definition (5). All that is required are the values of $W_{\mathcal{D}}$ at the irregular points and their neighbors. These can be computed in optimal order time by using the fast multipole method and high order quadrature [7]. Thus, we solve the system

$$\begin{aligned} \Delta_h(\tilde{W}_{\mathcal{D}})_{i,j} &= \begin{cases} 0 & \text{if } (x_i, y_j) \text{ is a regular point in } \mathcal{B} \\ C_{i,j} & \text{if } (x_i, y_j) \text{ is an irregular point in } \mathcal{B} \end{cases} \\ (\tilde{W}_{\mathcal{D}})_{i,j} &= W_{\mathcal{D}} \quad \text{on } \partial\mathcal{B}, \end{aligned} \tag{21}$$

where

$$C_{i,j} = \Delta_h W_{\mathcal{D}},$$

using the same Poisson solver as for the volume integral. Note that for interior problems, $W_{\mathcal{D}} = 0$ on $\partial\mathcal{B}$. For exterior problems, $W_{\mathcal{D}}$ is computed on $\partial\mathcal{B}$ by the fast multipole method.

5. NUMERICAL RESULTS

The fast solver described above has been implemented in double precision in Fortran on a Sun SPARCstation 2. In this section, we evaluate its performance on four multiply-connected interior and exterior problems.

For each example, a smooth resampling of the boundary, equispaced in arclength, was obtained using an algorithm due to Rokhlin [11]. Tangential derivatives of the curve were obtained by Fourier differentiation, from which it was straightforward to compute normal vectors and curvature. We then solved the Poisson equation on a sequence of refined meshes in order to examine the convergence rate and CPU time requirements of the algorithm.

Our results are summarized in Tables I–IV. In these tables, m_x and m_y are the numbers of points in each direction of the volume mesh, N is the number of points used to represent the boundary, and K is the number of CGS iterations required to achieve six digits of accuracy in the solution of the integral equation. E_2 and E_∞ denote the relative L_2 and L_∞ errors, and the remaining columns indicate the computational times required by the various steps of the method. T is the total CPU time (in seconds) required for the calculation, T_g is the time required for locating irregular points, T_v is the time required for computing the volume integral, T_b is the time required for solving the boundary integral equation, T_r is the time required for evaluating the double layer potential at the irregular points, and T_d is the time required for evaluating the double layer potential on the volume mesh.

EXAMPLE 1. We first consider the interior region with eight holes depicted in Fig. 1. The left-hand figure shows the boundary of the domain as well as the boundary of the volume mesh. The right-hand figure is a contour plot of the computed solution when the input data corresponds to the exact solution

$$u(r, \theta) = r^2 \sin(8\theta).$$

EXAMPLE 2. Our second example is the interior domain

TABLE III

Computational Results for Example 3

m_x	m_y	N	K	E_∞	E_2	T	T_g	T_v	T_b	T_r	T_d
256	256	3600	34	4.1×10^{-2}	1.4×10^{-2}	921	5	10	719	17	10
512	512	7200	34	6.1×10^{-3}	2.1×10^{-3}	1816	18	46	1411	26	46
1024	1024	14400	34			3561	71	199	2482	449	200

TABLE IV

Computational Results for Example 4

m_x	m_y	N	K	E_∞	E_2	T	T_g	T_v	T_b	T_r	T_d
108	64	1250	29	3.7×10^{-2}	9.6×10^{-3}	319	1	1	251	64	1
216	128	2500	29	4.0×10^{-3}	9.1×10^{-4}	605	2	4	454	136	4
432	256	5000	28	9.2×10^{-4}	1.8×10^{-4}	1212	9	17	932	219	17
864	512	10000	29	1.3×10^{-4}	2.1×10^{-5}	2382	29	79	1751	379	79
1728	1024	20000	32			5027	113	342	3293	664	340

Note. Here, T_r includes the time to evaluate the single layer potential on the boundary of the box (see Section 2.3).

with 20 holes depicted in Fig. 2. The input data here also corresponds to the exact solution

$$u(r, \theta) = r^2 \sin(8\theta).$$

EXAMPLE 3. Our third example is the interior domain with 11 holes depicted in Fig. 3. For this geometry, we have solved the problem

$$\Delta u = -(x^2 + y^2) \sin(xy),$$

with $u = 0$ on the outer boundary and u alternating between 1 and 2 on the interior boundary components. Since the exact solution is unknown, we have used the computed solution on the finest mesh as our exact solution for evaluating the errors on coarser meshes.

EXAMPLE 4. Our final example is the domain exterior to the 25 foil-shaped regions depicted in Fig. 4. In fluid mechanics terminology, we have solved

$$\Delta \Psi = -\omega,$$

where the vorticity ω is supported within the bounding volume mesh and consists of 40 separate patches. Thirty-two of the patches are of radius 0.06, centered near the tips of the foils and alternating in sign. The remaining eight are located farther away and have radius 0.1. The long dimension of each foil is approximately 0.22 and the dimensions of the embedding box are 3.2×1.9. The stream function Ψ has been set to a different constant on each boundary component. As for the third example, the exact solution is unknown and we have used the computed solution on the finest mesh as a standard for evaluating the errors on coarser meshes.

From the above, it is easy to verify that the algorithm is second-order accurate and that the time required grows approximately linearly with the number of unknowns. It is also evident that solving the boundary integral equation is the most expensive part of the calculation for small values of N, m_x, and m_y. As N, m_x, and m_y grow, however, the cost of the two fast Poisson solvers (T_v and T_d) will eventually dominate. In our examples, we made no attempt to optimize the number of boundary points, simply doubling all the parameters to study the convergence rate. In fact, we could have solved the integral equations with fewer boundary points and retained the same accuracy in the solution. In Example 1, for instance, using $N = 1920$ points in the discretization of the boundary with $m_x = m_y = 1024$ yields the same accuracy as using $N = 7680$. The cost of solving the integral equation in this case is only 10% of the total cost. Finally, we note that the time T_r for evaluating the double layer potential at irregular points could be substantially reduced by using an adaptive fast multipole method. The algorithm we used is based on a nonadaptive code [7].

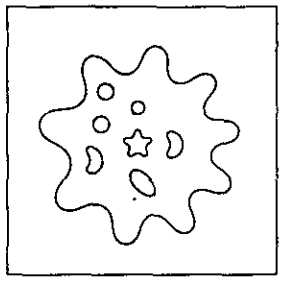
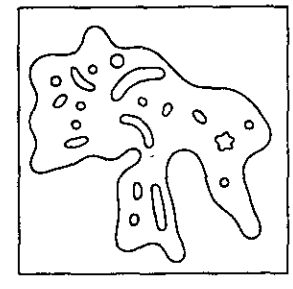


FIG. 1. The interior domain of Example 1 and a contour plot of the computed solution.



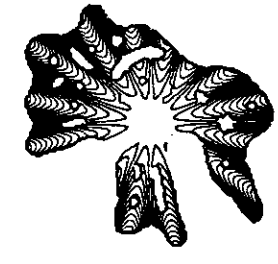


FIG. 2. The interior domain of Example 2 and a contour plot of the computed solution.

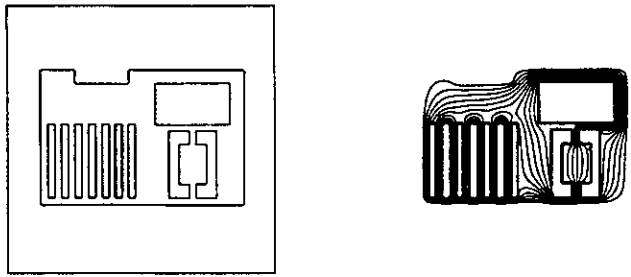


FIG. 3. The interior domain of Example 3 and a contour plot of the computed solution.

6. CONCLUSIONS

We have developed and implemented a robust fast Poisson solver for irregular, multiply-connected regions in the plane. The algorithm has a simple user interface and handles interior and exterior problems with equal ease. While limited at present to Dirichlet boundary conditions, the extension to Neumann, Robin, and mixed problems is relatively straightforward. The solution of the auxiliary Laplace equation in each of these cases can be expressed as a single layer potential or as a combination of single and double layer potentials. The resulting integral equation can be solved by a fast multipole-accelerated iterative procedure.

Extension of this work to three-dimensional problems has begun and will be reported at a later date. For related work on boundary integral equations in three dimensions, see [4].

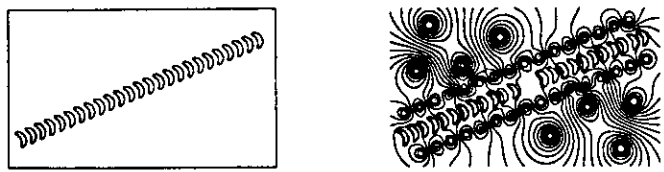


FIG. 4. The exterior domain of Example 4 with the enclosing box and a contour plot of the computed solution.

REFERENCES

1. B. L. Buzbee, G. H. Golub, and C. W. Nielson, *SIAM J. Numer. Anal.* **7,** 627 (1970).
2. J. Carrier, L. Greengard, and V. Rokhlin, *SIAM J. Sci. Statist. Comput.* **9,** 669 (1988).
3. A. Greenbaum, L. Greengard, and G. B. McFadden, *J. Comput. Phys.* **105,** 267 (1993).
4. T. Korsmeyer and J. White, "Multipole-Accelerated Preconditioned Iterative Methods for Three-Dimensional Potential Integral Equations," in *Boundary Element Methods 15 (BEM15), Worcester, MA, August 1993.*
5. A. Mayo, *SIAM J. Sci. Stat. Comput.* **6,** 144 (1985).
6. A. Mayo, *J. Comput. Phys.* **100,** 236 (1992).
7. A. McKenney, in preparation.
8. S. G. Mikhlin, *Integral Equations* (Pergamon, New York, 1957).
9. P. Sonneveld, *SIAM J. Sci. Stat. Comput.* **10,** 36 (1989).
10. V. Rokhlin, *J. Comput. Phys.* **60,** 187 (1985).
11. V. Rokhlin, unpublished result.