Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1984

A Tensor Product Generalized ADI Method for Ellipic Problems on Cylindrical Domains with Holes

Wayne R. Dyksen

Report Number:

84-495

A TENSOR PRODUCT GENERALIZED ADI METHOD FOR ELLIPTIC PROBLEMS ON CYLINDRICAL DOMAINS WITH HOLES

Wayne R. Dyksen

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

CSD-TR 495
October 1984

ABSTRACT

We consider solving second order linear elliptic partial differential equations together with Dirichlet boundary conditions in three dimensions on cylindrical domains (nonrectangular in x and y) with holes.

We approximate the partial differential operators by standard partial difference operators. If the partial differential operator separates into two factors, one depending on x and y, and one depending on z, then the discrete elliptic problem may be written in tensor product form as

$$(T_z \otimes I + I \otimes A_{xy})U = F.$$

We consider a specific implementation which uses a *Method of Planes* approach with unequally spaced finite differences in the xy direction and symmetric finite difference in the z direction. We establish the convergence of the *Tensor Product Generalized Alternating Direction Implicit* iterative method applied to such discrete problems. We show that this method gives a fast and memory efficient scheme for solving a large class of elliptic problems.

A Tensor Product Generalized ADI Method for Elliptic Problems on Cylindrical Domains with Holes

Wayne R. Dyksen

1. Introduction

Elliptic problems in three dimensions on nonrectangular domains present several difficulties. First is the often ignored problem of approximating the domain. This may be, in some sense, as difficult as the rest of the problem. Second, straightforward discretizations give very large linear systems, even for relatively coarse grids. Third, these systems often do not possess nice properties, and using simple band Gauss elimination is very expensive. We present a fast method for elliptic problems which separate into two factors, one depending on x and y, and one depending on z. We obtain a discrete problem of the form

$$(T_z \otimes I + I \otimes A_{xy})U = F. \tag{1.1}$$

using tensor products of matrices. We then apply a fast, tensor product ADI method to solve (1.1) efficiently.

In Section 2 we briefly introduce the *Tensor Product Generalized Alternating Direction Implicit* (TPGADI) method. We use finite differences to derive a tensor product formulation of the discrete problem in Section 3. In Sections 4 and 5, we apply the TPGADI method to this discrete problem, proving convergence for the Dirichlet problem. We explore a specific implementation in Section 6, showing that it is efficient both in time and memory.

2. The Two Directional Tensor Product Generalized ADI Methods

Let A_k and B_k be $N_k \times N_k$ matrices, and consider the linear system

$$(A_1 \otimes B_2 + B_1 \otimes A_2) C = F. \tag{2.1}$$

We wish to solve the two directional problem (2.1) by using methods employed to solve the one directional, simpler problems involving A_1, B_1, A_2 and B_2. The term *directional* is used rather than *dimensional* since one direction may encompass more than one dimension.

For a given set of positive *acceleration parameters* ρ_k, $k = 1,2,\ldots$, the two directional *Tensor Product Generalized Alternating Direction Implicit* (TPGADI) iteration method is defined by

$$C^{(0)} \text{ given}$$

$$\left[(A_1 + \rho_{k+1} B_1) \otimes B_2\right] C^{(k+1/2)} = F - \left[B_1 \otimes (A_2 - \rho_{k+1} B_2)\right] C^{(k)} \tag{2.2}$$

$$\left[B_1 \otimes (A_2 + \rho_{k+1} B_2)\right] C^{(k+1)} = F - \left[(A_1 - \rho_{k+1} B_1) \otimes B_2\right] C^{(k+1/2)}.$$

We use the following results in subsequent analysis; details are found in [Dyksen, 1984a].

Theorem 2.1. *Let A_k and B_k be matrices of order $N_k \times N_k$, and consider the linear system* (2.1) *for F given. Suppose that $B_1^{-1}A_1$ and $B_2^{-1}A_2$ have complete sets of normalized eigenvectors p_i and q_j, respectively, with corresponding positive eigenvalues λ_i and μ_j, respectively. Then, for a given set of positive acceleration parameters ρ_k, $k = 1,2,\ldots$, the two directional Tensor Product Generalized Alternating Direction Implicit iterative method, given by* (2.2) *is convergent, and C is its only solution.*

Corollary 2.2. *The TPGADI iterative method* (2.2) *can be exact (except for round-off) in a number of iterations equal to the number of unknowns in either direction; that is, in N_1 or N_2 iterations.*

Discrete elliptic problems arising from other discretizations in both two and three dimensions can be solved using the TPGADI method. In two dimensions we have considered the Method of Lines [Dyksen, 1982] and Hermite bicubic collocation [Dyksen, 1984a]. We also have solved problems on three dimensional rectangular domains using Hermite bicubic collocation in x and y, and finite differences in z [Dyksen, 1984b].

3. The Tensor Product Formulation of the Discrete Problem

Let Ω_2 be a bounded two dimensional domain contained in the rectangle $R=[a_x,b_x]\times[a_y,b_y]$. A three dimensional *cylindrical* domain Ω_3 is formed by the tensor product $\Omega_3=\Omega_2\times[a_z,b_z]$. We consider partial differential equations of the form

$$\begin{aligned} L_{xy}u+L_z u&=f \quad \text{in } \Omega_3 \\ u&=g \quad \text{on } \partial\Omega_3, \end{aligned} \tag{3.1}$$

where

$$L_{xy}u=-a(x,y)u_{xx}-b(x,y)u_{yy}+c(x,y)u_x+d(x,y)u_y+e(x,y)u, \quad a,b>0,\ e\geq 0, \tag{3.2a}$$

$$L_z u=-(p(z)u_z)_z+q(z)u,\ p>0,\ q\geq 0, \tag{3.2b}$$

and where f and g are given functions of x, y and z.

We first consider the subproblem of solving elliptic problems of the form

$$\begin{aligned} L_{xy}u&=f \quad \text{in } \Omega_2 \\ u&=g \quad \text{on } \partial\Omega_2, \end{aligned}$$

where f, g and u are functions of x and y. To solve such problems, we must approximate both the nonrectangular domain Ω_2 and the operator L_{xy}.

For given positive integers N_x and N_y, the rectangle R containing Ω_2 is subdivided by a rectangular grid defined by the grid lines

$$x_i = a_x + ih_x,\ h_x = \frac{b_x - a_x}{N_x + 1}, \quad \text{and} \quad y_j = a_y + jh_y,\ h_y = \frac{b_y - a_y}{N_y + 1}.$$

The interior of the domain Ω_2 is approximated by $\hat{\Omega}_2$, the set of grid points (x_i, y_j) in the interior of Ω_2. The boundary $\partial\Omega_2$ is approximated by $\partial\hat{\Omega}_2$, the intersection of the grid lines with $\partial\Omega_2$. Figure 3.1 shows an example of a nonrectangular domain. Note that small changes in N_x and N_y can substantially change the nature of the interior grid elements near the boundary. In practice, it is not always possible to choose the grid lines so that they intersect with $\partial\Omega_2$ in a nice way. However, as $N_x, N_y \to \infty$, these effects become less dramatic.

We approximate $L_{xy}u$ by finite difference operators. Consider a grid point $(x_i, y_j) \in \hat{\Omega}_2$, and let the distance to its nearest neighbor to the west, east, north and south be denoted by h_W, h_E, h_N and h_S, respectively; that is, we have the following five points for the finite difference approximations.

$$(x_i, y_{j+1}) = (x_i, y_j + h_N)$$

$$(x_{i-1}, y_j) = (x_i - h_W, y_j) \qquad (x_i, y_j) \qquad (x_{i+1}, y_j) = (x_i + h_E, y_j)$$

$$(x_i, y_{j-1}) = (x_i, y_j - h_S)$$

The partial differential operators in (3.2a) are replaced by the unequally spaced partial difference operators defined by

(3.3)

$$u_{xx}|_{ij} = \frac{2}{h_W(h_E + h_W)} u_{i-1,j} - \frac{2}{h_W h_E} u_{ij} + \frac{2}{h_E(h_E + h_W)} u_{i+1,j} + O(h_x)$$

$$u_x|_{ij} = \frac{-h_E}{h_W(h_E + h_W)} u_{i-1,j} + \frac{h_E - h_W}{h_W h_E} u_{ij} + \frac{h_W}{h_E(h_E + h_W)} u_{i+1,j} + O(h_x^2)$$

$$u_{yy}|_{ij} = \frac{2}{h_S(h_N + h_S)} u_{i,j-1} - \frac{2}{h_S h_N} u_{ij} + \frac{2}{h_N(h_N + h_S)} u_{i,j+1} + O(h_y)$$

$$u_y|_{ij} = \frac{-h_N}{h_S(h_N + h_S)} u_{i,j-1} + \frac{h_N - h_S}{h_S h_N} u_{ij} + \frac{h_S}{h_N(h_N + h_S)} u_{i,j+1} + O(h_y^2)$$

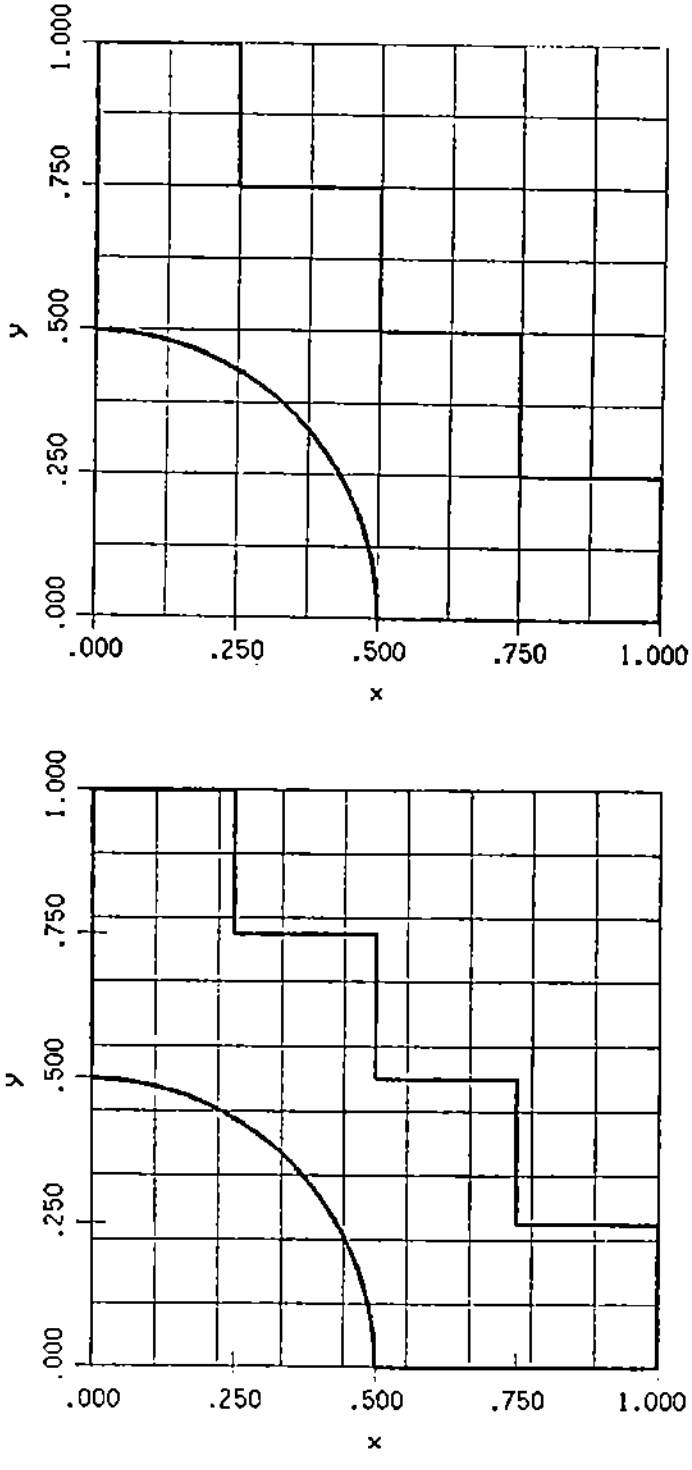

Figure 3.1 A nonrectangular domain from a problem involving heat flow in the shield of a nuclear reactor approximated with $N_x = N_y = 7$ and with $N_x = N_y = 8$ [Houstis, et. al., 1978]

where $u_{ij} = u(x_i, y_j)$ (see [Forsythe and Wasow, 1960], Theorems 20.2 and 20.4). Note that $\max(h_W, h_E) \le h_x$ and $\max(h_S, h_N) \le h_y$.

There are two distinct types of grid points in $\hat{\Omega}_2$: *regular* points which have all four of their nearest neighbors in $\hat{\Omega}_2$, and *irregular* points which have one or more nearest neighbors on the boundary $\partial\hat{\Omega}_2$. At irregular grid points, the finite difference approximations in (3.3) give only $O(h_x)$ and $O(h_y)$ approximations to u_{xx} and u_{yy}, respectively. At regular grid points, we have $h_W = h_E$ and $h_N = h_S$ so that (3.3) reduces to the standard equally spaced finite differences giving $O(h_x^2)$ and $O(h_y^2)$ approximations to u_{xx} and u_{yy}, respectively. We see from Figure 3.1 that, as $N_x, N_y \to \infty$, most of the interior grid points in $\hat{\Omega}_2$ are regular grid points. Thus, the discretization error in the finite difference approximation to $L_{xy}u$ is locally $O(h_x^2) + O(h_y^2)$ at most of the grid points in $\hat{\Omega}_2$. If u has a bounded fourth derivative in Ω_2, then the global discretization error is pointwise $O(h^2)$, where $h = \max(h_x, h_y)$ [Bramble and Hubbard, 1963, Theorem 3.1].

If we form the difference equations for each point $(x_i, y_j) \in \hat{\Omega}_2$, subtracting the boundary values on $\partial\hat{\Omega}_2$ from the right side, we obtain a system of simultaneous linear equations in the unknowns $U_{ij} \approx u(x_i, y_j)$, which we write as

$$A_{xy}\mathbf{u} = \mathbf{f}. \tag{3.4}$$

If Ω_2 is rectangular, then $u_{i+N_x(j-1)} = U_{ij}$. The matrix A_{xy} has dimension equal to the number of grid points in $\hat{\Omega}_2$ which is less than or equal to $N_x N_y$. If the grid points are ordered in a natural way (south to north, west to east), then A_{xy} has bandwidth less than or equal to N_y depending on the domain.

We now return to the original problem of solving three dimensional elliptic problems of the form (3.1) by using a "Method of Planes" approach. For a given positive integer M, we approximate the cylindrical domain Ω_3 by $M+2$ two dimensional cross sections defined by the planes

$$z_j = a_z + jh_z,\ \ h_z = \frac{b_z - a_z}{M+1}\ ,\ \ j = 0, 1, \ldots, M+1.$$

On each interior two dimensional domain $\hat{\Omega}_2 \otimes z_j$, we approximate L_{xy} at each point in the interior of $\hat{\Omega}_2 \otimes z_j$ by the partial difference operators (3.3). We approximate L_z by the standard symmetric finite differences. If we now let $U_{ij} \approx u(x_i, y_i, z_j)$, then our finite difference approximation to (3.1) results in a system of linear equations in the unknowns U_{ij}, which can be written in tensor product form as

(3.5) $$(T_z \otimes I + I \otimes A_{xy})U = F\ ,$$

where T_z is the symmetric tridiagonal matrix of order $M \times M$ defined by

$$T_z = \text{tridiag}\left[d_j^- \ d_j \ d_j^+\right],$$

where

(3.6) $$d_j^- = \frac{-p((j-½)h_z)}{h_z^2}$$

$$d_j = \frac{p((j-½)h_z) + p((j+½)h_z)}{h_z^2} + q(jh_z)$$

$$d_j^+ = \frac{-p((j+½)h_z)}{h_z^2}\ ,$$

and A_{xy} is defined in (3.4). Note that we use I to denote the identity matrix of possibly different orders.

4. The Tensor Product Generalized ADI Method for Cylindrical Domains

For a given set of positive acceleration parameters ρ_k, $k = 1, 2, \ldots$, the TPGADI method for the partial difference equations in (3.5) is given by

$U^{(0)}$ given

$$\left[(T_z+\rho_{k+1}I)\otimes I\right]U^{(k+1/2)}=F-\left[I\otimes(A_{xy}-\rho_{k+1}I)\right]U^{(k)} \tag{4.1}$$

$$\left[I\otimes(A_{xy}+\rho_{k+1}I)\right]U^{(k+1)}=F-\left[(T_z-\rho_{k+1}I)\otimes I\right]U^{(k+1/2)}.$$

This special case of the TPGADI method (2.1) with $B_1=B_2=I$ is similar in nature to the Peaceman-Rachford method [Young, 1971, Chapter 17]. In traditional three dimensional ADI applications, the partial differential operator is required to separate into three factors, and the domain is required to be a rectangular right prism. The resulting discrete elliptic problem is

$$(A\otimes I\otimes I+I\otimes B\otimes I+I\otimes I\otimes C)U=F,$$

which is solved using a three directional ADI scheme [Varga, 1962, Section 7.4]. By combining the x and y dimensions into one factor, we can solve a considerably larger class of problems while still using an efficient TPGADI method.

5. Convergence of the Tensor Product Generalized ADI Method

We now establish the convergence of the TPGADI iterative method (4.1) if applied to the discrete elliptic Dirichlet problem (3.5).

THEOREM 5.1. *For h_x, h_y and h_z sufficiently small, the TPGADI method* (4.1) *is convergent if applied to the discrete elliptic problem* (3.5).

Proof. Let $E^{(k)}=U^{(k)}-U$ denote the error of the k^{th} iterate. A straightforward computation shows that the components of the error satisfy

$$E_{ij}^{(k)}=\prod_{l=1}^{k}\left[\frac{\lambda_i-\rho_l}{\lambda_i+\rho_l}\;\frac{\mu_j-\rho_l}{\mu_j+\rho_l}\right]E_{ij}^{(0)} \tag{5.1}$$

where λ_i and μ_j denote the eigenvalues of T_z and A_{xy}, respectively.

Now, for h_z sufficiently small, the tridiagonal matrix T_z resulting from the z direction symmetric finite difference approximation to $L_z u$ is symmetric positive definite so that its eigenvalues are real and positive. Since the acceleration parameters ρ_l are always taken to be real and positive, it follows that for all i, l

(5.2) $$\left|\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l}\right| \le \epsilon < 1.$$

For h_x, h_y and h_z sufficiently small, A_{xy} is diagonally dominant [Forsythe and Wasow, 1960, Section 20.7]. Moreover, since the domain Ω_2 is connected, and since strict inequality holds for the boundary equations, it follows that A_{xy} is irreducibly diagonally dominant. Hence, since A_{xy} has positive diagonal elements, its eigenvalues satisfy $\mathrm{Re}\mu_j > 0$ [Varga, Theorem 1.8]. Thus, for all j, l we have

(5.3) $$\left|\frac{\mu_j - \rho_l}{\mu_j + \rho_l}\right| \le \epsilon < 1$$

Combining the inequalities in (5.2) and (5.3), we see from (5.1) that

$$\lim_{k\to\infty} |E_{ij}^{(k)}| = \lim_{k\to\infty} \prod_{l=1}^{k} \left|\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l}\;\frac{\mu_j - \rho_l}{\mu_j + \rho_l}\; E_{ij}^{(0)}\right| = 0$$

which implies that

$$\lim_{k\to\infty} \|E^{(k)}\| = 0 \;\Box$$

We note that the latter part of the proof of Theorem 5.1 is merely a proof of the fact that Theorem 2.1 is still valid under the weaker assumptions that $\mathrm{Re}\lambda_i > 0$ and $\mathrm{Re}\mu_j > 0$.

6. Computer Implementation and Performance Evaluation

We use some of the advanced features of ELLPACK[†] [Rice and Boisvert, 1985] to

[†]ELLPACK is a very high level computer language developed at Purdue University for solving second order linear elliptic partial differential equations.

implement our numerical method for elliptic problems on cylindrical domains. The implementation takes the form of an ELLPACK program together with supplemental Fortran subprograms. ELLPACK automatically discretizes the two dimensional domain Ω_2 and partial differential operator L_{xy}. Thus, we use existing software parts in a novel way to solve at least two difficult subproblems of the original problem. The supplemental subprograms discretize $L_z u$ and solve the resulting discrete problem using the TPGADI method. ELLPACK "thinks" that we are solving a two dimensional problem. A sample ELLPACK program is given in Appendix A.

We now consider briefly the computational complexity of the TPGADI method derived for the discrete elliptic problem $(T_z \otimes I + I \otimes A_{xy})U = F$. Recall that the tridiagonal matrix T_z has dimension $M \times M$. We assume that A_{xy} has dimension $N_{xy} \times N_{xy}$ with bandwidth K_{xy}; recall that N_{xy} and K_{xy} depend on the two dimensional nonrectangular domain Ω_2, with $N_{xy} \le N_x N_y$ and $K_{xy} \le N_y$. Moreover, we assume that $M = O(N_x) = O(N_y)$ since this is a most likely case for typical applications. The work required to compute one iteration of the TPGADI method (2.2) is in [Dyksen, 1984a]. For the special case $B_1 = B_2 = I$ as in (4.1), the work per iteration is summarized in Table 6.1.

Table 6.1
Work to compute one sweep of the TPGADI method for partial difference equations on cylindrical domains

z-direction sweep		xy-direction sweep	
Operation	Work	Operation	Work
$W_2 = A_{xy} - \rho_{k+1} I$	$\frac{1}{2} N_{xy}$	$W_1 = T_z - \rho_{k+1} I$	$\frac{1}{2} M$
$W = (I \otimes W_2) U^{(k)}$	$2MK_{xy}N_{xy}$	$W = (W_1 \otimes I) U^{(k+\frac{1}{2})}$	$2MN_{xy}$
$W = F - W$	MN_{xy}	$W = F - W$	MN_{xy}
$W_1 = T_z + \rho_{k+1} I$	$\frac{1}{2} M$	$W_2 = A_{xy} + \rho_{k+1} I$	$\frac{1}{2} N_{xy}$
$U^{(k+\frac{1}{2})} = (W_1 \otimes I)^{-1} W$	$2M + 3MN_{xy}$	$U^{(k+1)} = (I \otimes W_2)^{-1} W$	$2Kxy^2 N_{xy} + 3MK_{xy}N_{xy}$

Thus, the work required per iteration for the z-direction sweep is $O(2MK_{xy}N_{xy})$ and for the

xy-direction sweep is $O(3MK_{xy}N_{xy})$ so that the total work per iteration is $O(5MK_{xy}N_{xy})$. Since the TPGADI iterative method can be a direct method in M iterations, it follows that the total work to solve the discrete problem (3.5) is $O(5M^2K_{xy}N_{xy})$. For the simple "worst case" $N=M=N_x=N_y$, $N_{xy}=N_xN_y$ and $K_{xy}=N_{xy}$, this simplifies to $O(5N^5)$.

By contrast, $(T_x \otimes I + I \otimes A_{xy})$ has dimension $MN_{xy} \times MN_{xy}$ and approximate bandwidth N_{xy}. The work to factor it using band Gauss elimination with partial pivoting is $O(MN_{xy}^3)$ operations. For the simple worst case considered above, this simplifies to $O(N^7)$. Thus, even as a direct method, the TPGADI method is asymptotically much faster than band Gauss elimination.

The memory required by the TPGADI method is nearly optimal, $O(3MN_{xy}+6K_{xy}N_{xy})$ words. For the simple worst case considered above, this simplifies to $O(9N^3)$ words, nine times the number of unknowns. To factor $(T_x \otimes I + I \otimes A_{xy})$ using band Gauss elimination, $O(3MN_{xy}^2)$ words are required; $O(3N^5)$ words if $M=N_x=N_y$, $N_{xy}=N_xN_y$ and $K_{xy}=N_{xy}$. Thus, the TPGADI method gives a potential for using a relatively large number of grid lines to solve three dimensional elliptic problems.

The following numerical results were computed on a VAX 11/780 (UNIX, 4.1BSD) with a floating-point accelerator using the Fortran compiler f77 with optimizer in single precision. The acceleration parameters ρ_k are computed to be the eigenvalues of the symmetric positive definite matrix T_x by the EISPACK routine IMTQL1 [Smith, et. al., 1976], [Wilkinson, 1962]; the time required to compute these eigenvalues is always included in timings of the TPGADI method. They are used in increasing order [Lynch and Rice, 1968]. The initial iterate, $U^{(0)}$, is always taken to be zero.

EXAMPLE 6.1. Performance of the TPGADI Method with N Varied

Let Ω_2 be the two dimensional circular domain defined by

$$\Omega_2=\{(x,y) \mid (x-½)^2+(y-½)^2<½\},$$

and let Ω_3 be the right circular cylinder defined by $\Omega_3 = \Omega_2 \otimes [0,1]$. We consider the Model Dirichlet Problem

$$-u_{xx} - u_{yy} - u_{zz} = f \quad \text{in } \Omega_3 \qquad (6.1)$$
$$u = g \quad \text{on } \partial\Omega_3,$$

where f and g are chosen so that $u(x,y,z) = x^2y^2z^3$. We solve (6.1) with $1/h = 4, 8, 16, 32$ where $N = M = N_x = N_y$ so that $h = h_z = h_x = h_y = \frac{1}{N+1}$. The maximum relative error at the grid points interior to Ω_3 is computed. The results are summarized in Table 6.2.

Table 6.2
The TPGADI method applied to the partial difference equations arising from the Model Dirichlet Problem on a cylindrical domain

$N+1=1/h$	K_{xy}	N_{xy}	Number of Unknowns	Number of Iterations	Solution Time (Secs)	Maximum Error
4	3	9	27	3	0.12	9.4190e-08
8	7	45	315	7	3.03	7.2661e-07
16	15	193	2895	15	111.45	5.5426e-06
32	31	793	24583	31	4075.00	7.9324e-05

A logarithmic fit of this timing data shows that Time $\approx 6.80 * 10^{-4} * N^{4.48}$ which agrees with the worst case theoretical work estimate of $O(5N^5)$ operations. Note that we are using the TPGADI method as a direct method; in practice, one would use many fewer than N sweeps. The partial difference operators in (3.3) and (3.6) are theoretically exact on the Model Dirichlet Problem with solution $u(x,y,z) = x^2y^2z^3$. Machine round-off is achieved, and the round-off errors do not grow significantly since Error $\approx 3.34 * 10^{-9} * N^{2.85}$.

The TPGADI method uses a relatively modest amount of memory to solve this three dimensional problem. For the case $1/h = 32$ ($N = 31$), we use on the order of 220,000 words of memory. The matrix $(T_z \otimes I + I \otimes A_{xy})$ has dimension 24583×24583 with approximate bandwidth 793. The amount of memory required to store and factor it using band Gauss elimination is approximately 58.5 million words.

Example 6.2. The TPGADI Method Applied to the Partial Difference Equations Arising from Problem 18

Let Ω_2 be the two dimensional nonrectangular domain given in Figure 3.1 and let Ω_3 be the cylindrical domain defined by $\Omega_3 = \Omega_2 \otimes [0,1]$. We extend to three dimensions the two dimensional elliptic operator of **Problem 18** of the population of partial differential equations in [Rice, et. al., 1981]; in particular, we consider

$$(6.2) \qquad \begin{aligned} -u_{xx} - (1+xy)u_{yy} - (\sin(z)u_z)_z - \cos(x)u_x + e^{-x}u_y + (3+z^2)u &= f \quad \text{in } \Omega_3 \\ u &= g \quad \text{on } \partial\Omega_3, \end{aligned}$$

where f and g are chosen so that $u(x,y,z) = \sin(2\pi x)\cos(4\pi y)e^z$.

We solve (6.2) using $h = h_z = h_x = h_y = \dfrac{1}{N+1}$. The smallest $(N+1)/2$ eigenvalues of T_z are used as the acceleration parameters. The results are given in Table 6.3.

Table 6.3
The TPGADI method applied to the partial difference equations arising from Problem 18 on a cylindrical domain

$N+1=1/h$	K_{xy}	N_{xy}	Number of Unknowns	Number of Iterations	Solution Time (Secs)	Maximum Error
4	2	3	9	2	0.04	6.4266e-01
8	5	17	119	4	0.53	1.2656e-01
16	11	88	1320	8	18.80	3.0372e-02
32	23	394	12214	16	706.24	7.9741e-03

We obtain Error $\approx 110h^{2.11}$ which agrees with the theoretical convergence rate of $O(h^2)$. The number of iterations is chosen *a priori*, somewhat arbitrarily; it is possible that fewer iterations would produce satisfactory results. Figure 6.1 contains contour plots of two cross sections of the error in solving Problem 18 for the case $h = 1/16$. Note that the errors on some of the contours are larger than the maximum error given in Table 6.3; this is due to the error in the interpolation scheme used by ELLPACK near the boundary.

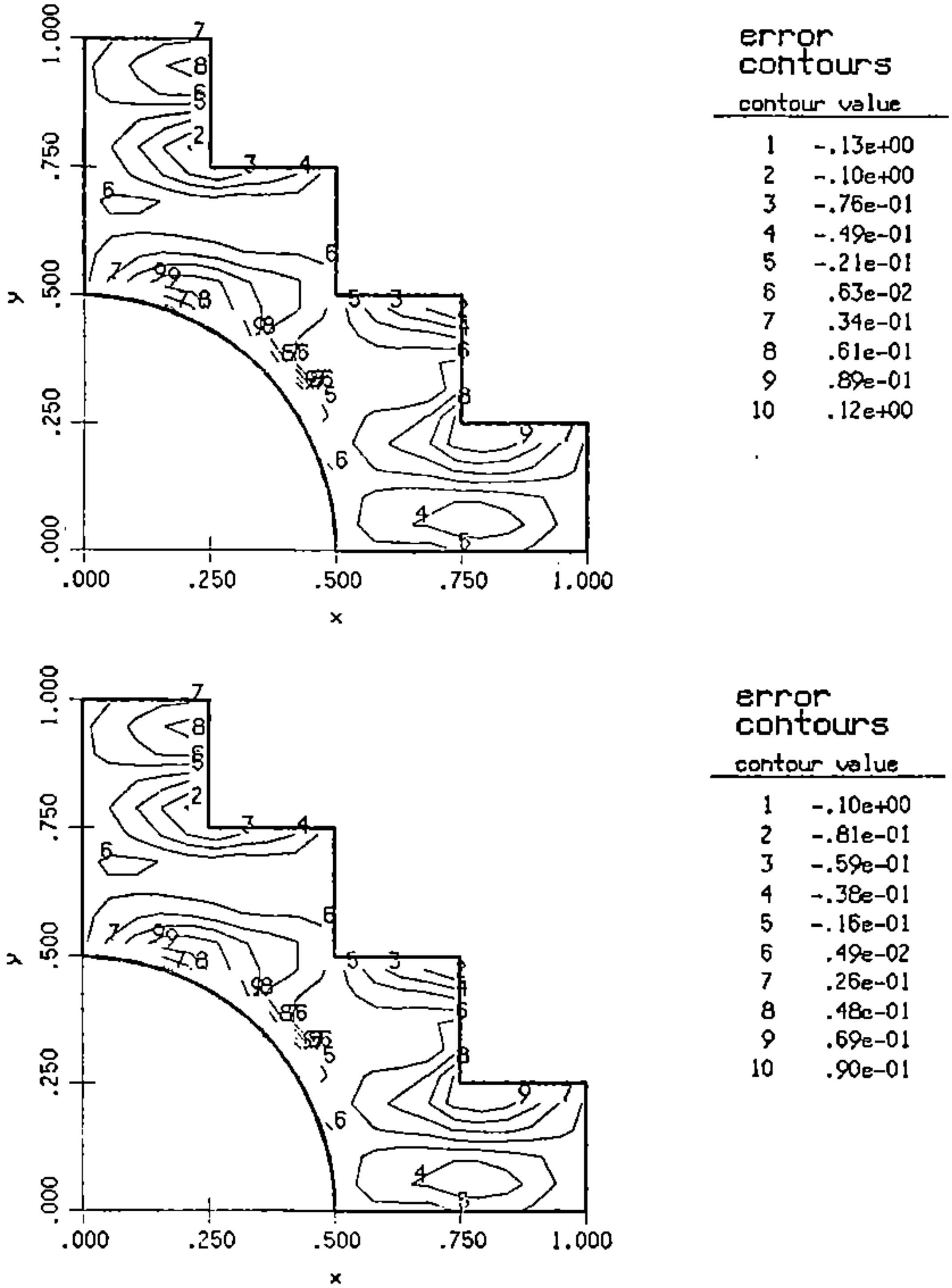

Figure 6.1 Contour plots of two cross sections of the error in solving Problem 18 on the planes $z = 1/2$ (top) and $z = 1/4$ (bottom) for the case $h = 1/16$

EXAMPLE 6.3. The TPGADI Method Applied to the Partial Difference Equations Arising from a Cylindrical Domain with a Hole

ELLPACK provides a so-called ***HOLE segment*** which defines a hole to be removed from a two dimensional nonrectangular domain defined by a BOUNDARY segment. The discretization module 5 POINT STAR is designed to handle such domains. As a result, our implementation allows cylindrical domains with holes. Let Ω_2 be the two dimensional nonrectangular domain with a hole given in Figure 6.2. Let Ω_3 be the cylindrical domain defined by $\Omega_3 = \Omega_2 \otimes [0,1]$. We consider the heat conduction problem defined by

$$(6.3) \qquad \begin{aligned} -u_{xx} - u_{yy} - u_{zz} &= 0 \quad \text{in } \Omega_3 \\ u &= g \quad \text{on } \partial\Omega_3, \end{aligned}$$

where g is defined by

$$g(x,y,z) = \begin{cases} 1600[z(1-z)]^2 & \text{if } (x-1/4)^2 + (y-1/4)^2 = 1/16 \\ 0 & \text{elsewhere.} \end{cases}$$

The solution u to (6.3) can be interpreted as the steady state temperature distribution within Ω_3, given that the boundary is kept at the temperatures defined by g.

We see from Figure 6.1 that the domain Ω_2 is difficult to approximate particularly since the hole is so close to the left boundary $x = 0$. For example, even if $h_x = 1/16$, there would be only one grid line between them. We solve (6.3) using $h_x = h_y = 1/32$, $h_z = 1/16$ and $h_x = 1/64$, $h_y = 1/32$, $h_z = 1/16$, giving 8400 and 17,190 unknowns to compute, respectively. We use 8 iterations of the TPGADI method resulting in solution times of 364.74 seconds and 769.28 seconds, respectively. Figure 6.3 and Figure 6.4 are contour plots of a cross section of the computed solution on the planes $z = 1/2$ and $z = 1/4$. The computed solution shows the heat flowing out from the hole through Ω_3 to the outer cool boundaries of Ω_3. Note that even though the contour plots look similar on the two different planes, the maximum value in the solution differs by a factor of two.

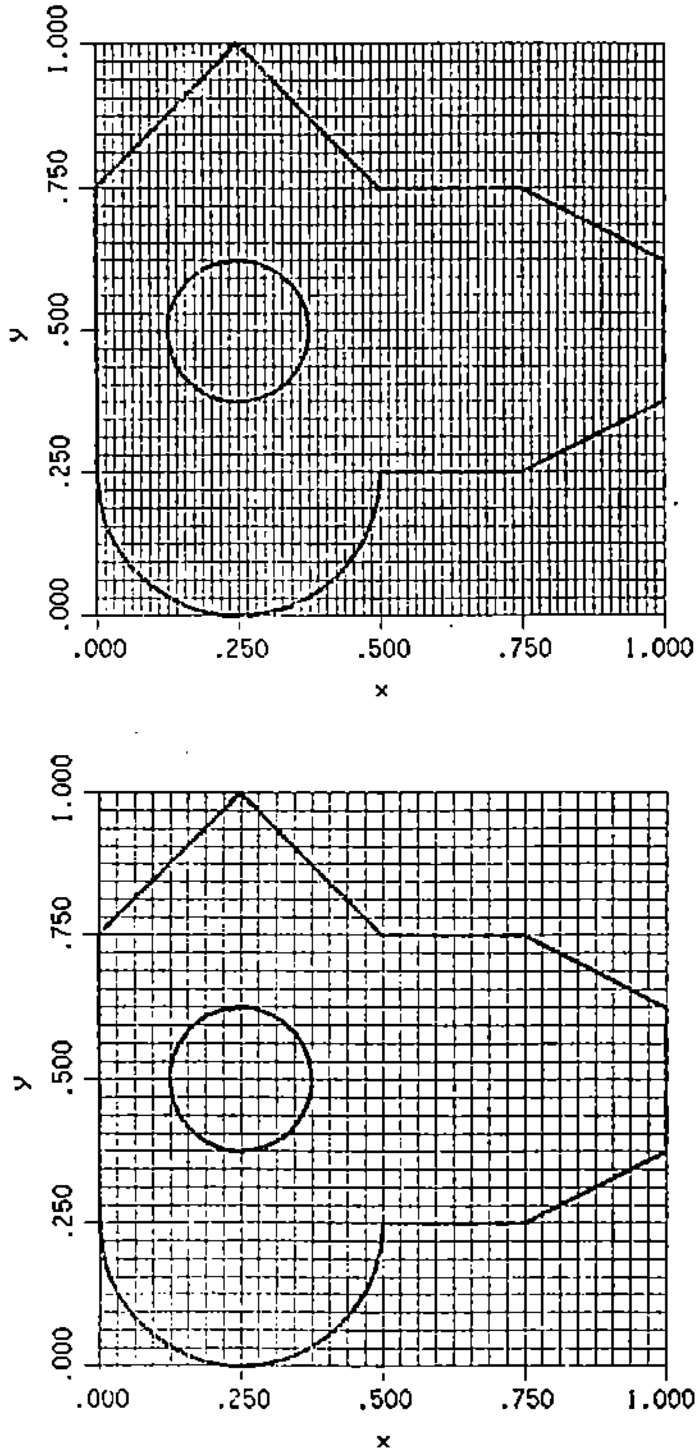

Figure 6.2 A graph of a cross section of the domain for Example 6.2 with the grid lines for the cases $h_x = h_y = 1/32$ (top) and $h_x = 1/64$, $h_y = 1/32$ (bottom)

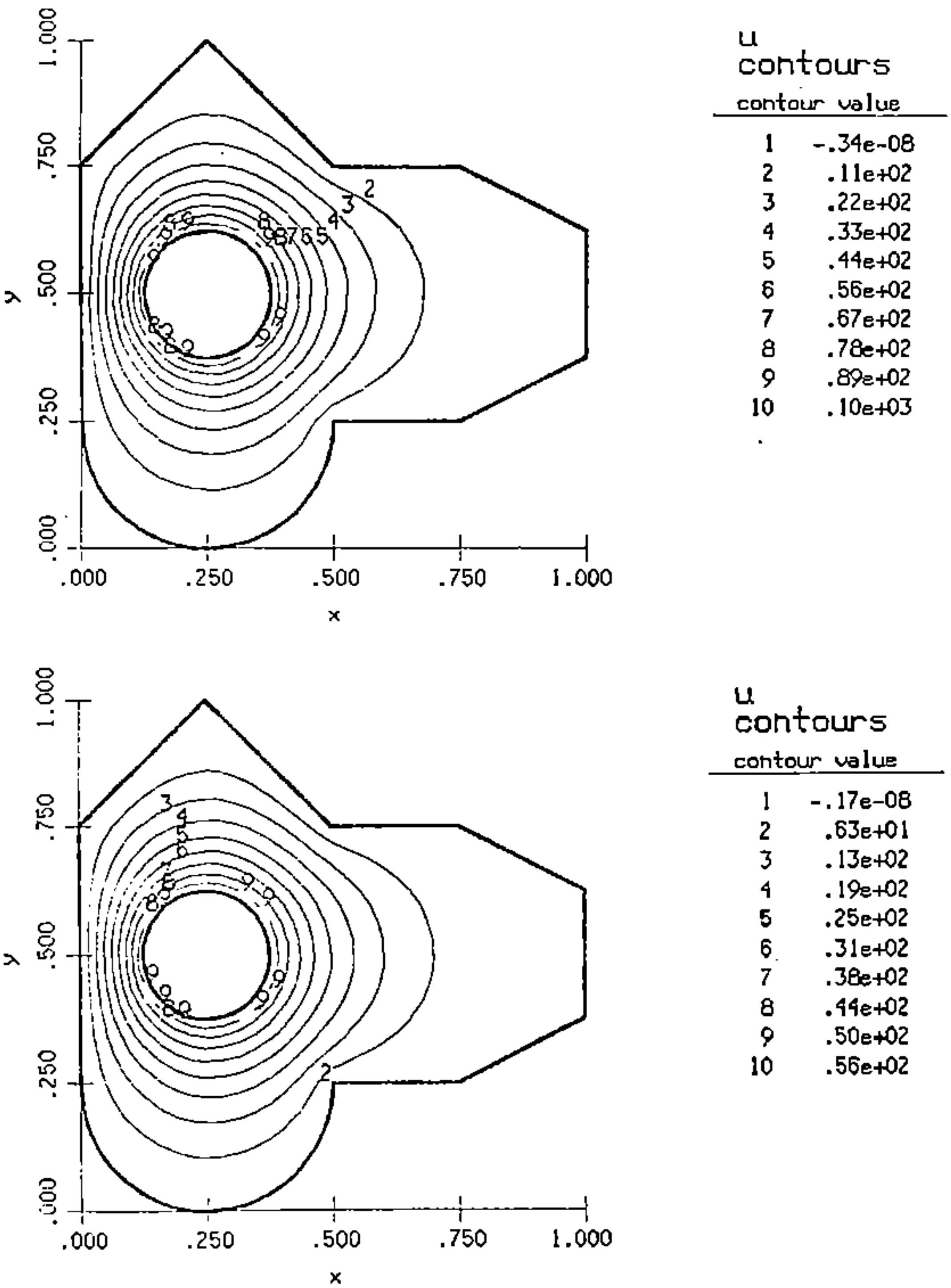

Figure 6.3 Contour plots of two cross sections of the computed solution to a heat conduction problem on the planes $z = 1/2$ (top) and $z = 1/4$ (bottom) for the case $h_x = h_y = 1/32$, $h_z = 1/16$

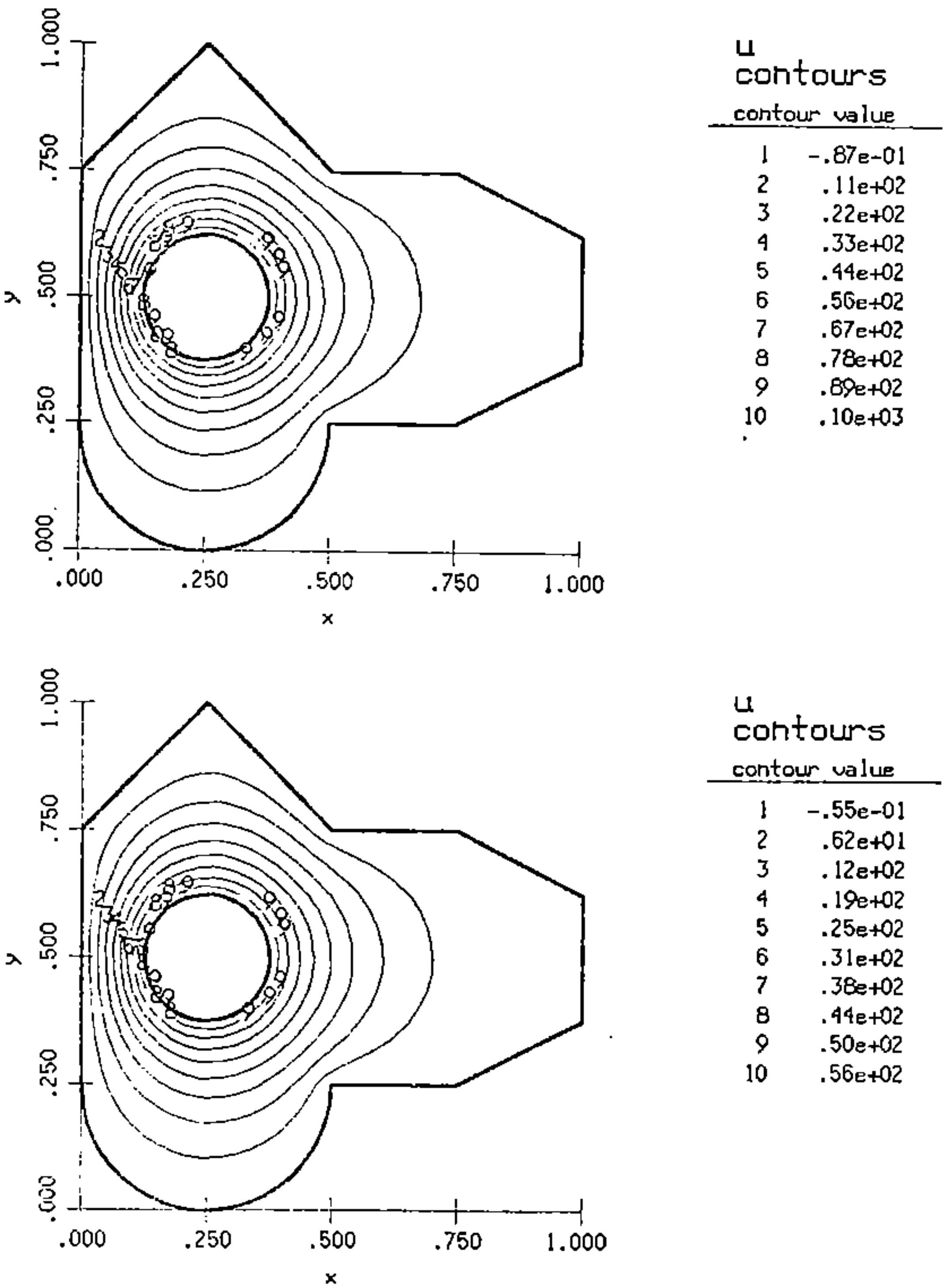

Figure 6.4 Contour plots of two cross sections of the computed solution to a heat conduction problem on the planes $z = 1/2$ (top) and $z = 1/4$ (bottom) for the case $h_x = 1/64$, $h_y = 1/32$, $h_z = 1/16$

7. References

1. G. Birkhoff, R. S. Varga and D. M. Young, "Alternating direction implicit methods", *Advances in Computers*, Academic Press, New York, 3(1962), 189-273.

2. G. Birkhoff and R. E. Lynch, *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia, 1985.

3. R. F. Boisvert, "Families of high order accurate discretizations of some elliptic problems", *SIAM J. Sci. and Stat. Comp.*, 2(1981), 268-284.

4. J. H. Bramble and B. E. Hubbard, "A theorem on error estimation for finite difference analogues of the Dirichlet problem for elliptic equations", *Contributions to Differential Equations*, 2(1963), 319-340.

5. C. de Boor, "Efficient computer manipulation of tensor products", *ACM Trans. Math. Software*, 5(1979), 173-182.

6. J. Douglas, Jr. and H. H. Rachford, Jr., "On the numerical solution of heat conduction problems in two or three space variables", *Trans. Amer. Math. Soc.* 82(1956), 421-439.

7. W. R. Dyksen, "Tensor Product Generalized Alternating Direction Implicit Methods For Solving Separable Second Order Linear Elliptic Partial Differential Equations", Ph.D. Thesis, Purdue University, 1982.

8. W. R. Dyksen, "Tensor product generalized ADI methods for elliptic problems", Purdue University, Computer Science Department Report CSD-TR 493, September 1984, submitted for publication.

9. W. R. Dyksen, "A tensor product generalized ADI method for the method of planes", Purdue University, Computer Science Department Report CSD-TR 494, October 1984, submitted for publication.

10. W. R. Dyksen and J. R. Rice, "Symmetric versus nonsymmetric differencing", *SIAM J. Sci. Stat. Comp.*, 6(1985), 45-48.

11. G. E. Forsythe and W. R. Wasow, *Finite Difference Methods for Partial Differential Equations*, John Wiley and Sons, New York, 1960.

12. E. N. Houstis, R. E. Lynch, J. R. Rice and T. S. Papatheodorou, "Evaluation of Numerical Methods for Elliptic Partial Differential Equations", *J. Comput. Phys.*, 27(1978), 323-350.

13. R. E. Lynch, J. R. Rice and D. H. Thomas, "Direct solution of partial difference equations by tensor product methods", *Numer. Math.*, 6(1964), 185-199.

14. R. E. Lynch, J. R. Rice and D. H. Thomas, "Tensor product analysis of alternating direction implicit methods", *J. Soc. Indust. Appl. Math.*, 13(1965), 995-1006.

15. R. E. Lynch, and J. R. Rice, "Convergence rates of ADI methods with smooth initial error", *Math. Comp.*, 22(1968), 311-355.

16. R. E. Lynch, and J. R. Rice, "High accuracy approximations to solutions of elliptic partial differential equations", *Proc. Nat. Acad. of Sci.*, 75(1978), 2541-2544.

17. R. S. Martin and J. H. Wilkinson, "The implicit QL algorithm", *Numer. Math.*, 12(1968), 377-383.

18. D. W. Peaceman and H. H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations", *J. Soc. Indust. Appl. Math.*, 3(1955), 28-41.

19. J. R. Rice, E. N. Houstis and W. R. Dyksen, "A population of linear, second order, elliptic partial differential equations on rectangular domains, Parts 1 and 2", *Math. Comp.*, 36(1981), 475-484.

20. J. R. Rice, "Numerical computation with general two dimensional domains", *ACM Trans. Math. Software*, 10(1984).

21. J. R. Rice and R. F. Boisvert, *Solving Elliptic Problems using ELLPACK*, Springer-Verlag, New York, 1985.

22. B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema and C. B. Moler, *Matrix Eigensystem Routines - EISPACK Guide*, Springer-Verlag, New York, 1976.

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

24. J. H. Wilkinson, "Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection", *Numer. Math.*, 4(1962), 362-367.

25. D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

8. Appendix A - A Sample ELLPACK Program

Our numerical method for elliptic problems on cylindrical domains with holes is implemented within the ELLPACK system [Rice and Boisvert, 1985]. We use an ELLPACK program supplemented with Fortran subprograms. The two dimensional domain Ω_2 and partial differential operator L_{xy} are discretized by ELLPACK. The discretization of L_z and the TPGADI solution of the discrete problem is done by the supplemental non-ELLPACK subprograms. Note that ELLPACK "thinks" that we are solving a two dimensional problem. A sample ELLPACK program is given Figure 8.1 for the Poisson problem on a right circular cylinder.

The ELLPACK language provides a simple and natural way to express a two dimensional nonrectangular domain by specifying a sequence of parameterized sides together with boundary conditions. For example, the domain in Figure 3.1 is defined in ELLPACK by the following so-called *BOUNDARY segment*:

```
BOUNDARY.
          U = 0.0 ON X = 0.5*SIN(T), Y = 0.5*COS(T) FOR T=0. TO PI/2.
                  ON LINE  0.50,0.00 TO 1.00,0.00 TO 1.00,0.25 TO
                           0.75,0.25 TO 0.75,0.50 TO 0.50,0.50 TO
                           0.50,0.75 TO 0.25,0.75 TO 0.25,1.00 TO
                           0.00,1.00 TO 0.00,0.50
```

In this BOUNDARY segment, homogeneous Dirichlet boundary conditions are specified on all sides of the domain.

A two dimensional, nonrectangular domain is discretized within ELLPACK using the scheme described in Section 3 [Rice, 1984]. The domain processor overlays the rectangular grid of points on the domain, determines which grid points are inside and outside of the domain, determines which interior grid points are next to the boundary, and finds the intersections of the grid lines with the boundary of the domain. The boundary intersection points must be determined accurately relative to the discretization error so that the Dirichlet boundary data is evaluated accurately.

```
******************************************************************
*                                                                *
*   SAMPLE ELLPACK PROGRAM FOR PARTIAL DIFFERENCE EQUATIONS ON   *
*   CYLINDRICAL DOMAINS AND THE TPGADI ITERATIVE METHOD          *
*                                                                *
******************************************************************

GLOBAL.
      COMMON / TPZZZZ / Z

DECLARATIONS.
      PARAMETER (NGDZMX = 9)
      PARAMETER (NPLNMX = NGDZMX-2)
      PARAMETER (NBDMAX = $I1NGRY - 2)
      PARAMETER (NCOLMX = 2*NBDMAX + 1)
      PARAMETER (NWXLXY = $I1MNEQ*(NBDMAX + 1))
      COMMON / TPRSID / TPRSID($I1MNEQ,NPLNMX)
      COMMON / TPUNKN / TPUNKN($I1MNEQ,NPLNMX)
      COMMON / GRIDZZ / GRIDZ(NGDZMX)
      REAL
     A      TZ(NPLNMX,2),
     B      AXY($I1MNEQ,NCOLMX),
     C      WORKNN($I1MNEQ,NCOLMX),
     D      WORKMN(NPLNMX,$I1MNEQ),
     E      WORK(NWXLXY),
     F      RHD(NPLNMX)
*                               2 3      2 3      2 2
*     - U    - U    - U   = - (2Y Z  + 2X Z  + 6X Y Z)
*        XX     YY     ZZ

EQUATION.
      - UXX - UYY = - (2.*Y**2*Z**3 + 2.*X**2*Z**3 + 6.*X**2*Y**2*Z)

BOUNDARY.
      U=X**2 * Y**2 * Z**3 ON X=COS(T), Y=SIN(T) FOR T=0.0 TO 2.*PI

GRID.
      9 X POINTS -1.0 TO 1.0
      9 Y POINTS -1.0 TO 1.0

FORTRAN.
C
C     DEFINE Z GRID
C
      AZ = -1.0
      BZ =  1.0
      NGRIDZ = 9
      HZ = (BZ-AZ)/(NGRIDZ-1)
      NGDZM2 = NGRIDZ-2
      GRIDZ(1) = AZ
      DO 10 KZ = 2, NGRIDZ-1
         GRIDZ(KZ) = AZ + (KZ-1)*HZ
   10 CONTINUE
      GRIDZ(NGRIDZ) = BZ
```

Figure 8.1 Sample ELLPACK program for partial difference equations on cylindrical domains and the TPGADI iterative method. Supplementary Fortran program are loaded from a precompiled library.

```
C
C      DISCRETIZE X,Y OPERATOR, BUILD THE RIGHT SIDE TPRSID
C      AND GUESS THE SOLUTION TPUNKN
C
DISCRETIZATION.    5 POINT STAR
FORTRAN.
C
C      INTERFACE 5 POINT STAR OUTPUT FOR INPUT TO TPGADI
C
       CALL BLDAXY (R1COEF,AXY,I11DCO,I1MNEQ,I1MNCO,
     A              I1ENDX,I1UNDX,NBANDU,NBANDL)
C
C      DISCRETIZE THE Z OPERATOR -(P(Z)U ) + Q(Z)U
C                                        Z Z
       CALL BILDTZ (TZ,NPLNMX)
C
C      COMPUTE THE ITERATION PARAMETERS RHO(K)
C
       IRHO = 1
       NITERS = NGRIDZ-2
       CALL SETRHO (IRHO,RHO,NGRIDZ,NITERS,TZ,NPLNMX,WORK)
C
C      SOLVE ( TZ X I   +  I X AXY ) TPUNKN = TPRSID
C
       NZBAND = 1
       MXYBND = MAX0(NBANDL,NBANDU)
       CALL TPGADI (TZ,BZZ,NPLNMX,NGDZM2,NZBAND,AXY,BXY,I1MNEQ,I1NEQN,
     A              MXYBND,TPRSID,TPUNKN,BZFACT,BXYFCT,WORKMM,WORKNN,
     B              WORKMN,WORKBZ,WRKBXY,WORK,NITERS,RHO)
C
C      EVALUATE SOLUTION AND ERROR ON EACH PLANE
C
       DO 20 KZ = 1, NGDZM2
          Z = GRIDZ(KZ+1)
          PRINT *, '*** PLANE Z =', Z
          INITL = 1
OUTPUT.  MAX(TRUE) $ MAX(ERROR)
FORTRAN.
   20 CONTINUE

SUBPROGRAMS.
C
C      COEFFICIENTS OF Z DIRECTION OPERATOR
C
       FUNCTION ZPCOE(Z)
       ZPCOE = - 1.
       RETURN
       END
       FUNCTION ZQCOE(Z)
       ZQCOE = 0.
       RETURN
       END
C
C      TRUE SOLUTION
C
       FUNCTION TRUE(X,Y)
       COMMON / TPZZZZ / Z
       TRUE = X**2 * Y**2 * Z**3
       RETURN
       END
END.
```

Figure 8.1 (Continued)

Given the graph of a domain and the grid lines as in Figure 3.1, the task of "processing" a nonrectangular two dimensional domain is easy to do "by eye". However, the automation of this process within a computer program is nontrivial. The domain processor consists of approximately 1450 lines of executable Fortran. By contrast, the totality of subprograms which construct and solve the discrete elliptic problem contain approximately 1200 lines of code. Hence, to implement our numerical method on cylindrical domains, the problem of approximating the domain is in some sense as difficult (as measured by the amount of Fortran code) as that of approximating the solution of the elliptic problem.

The ELLPACK discretization module 5 POINT STAR uses the output from the domain processor to construct the matrix A_{xy} in (3.5); that is, 5 POINT STAR approximates $L_{xy}u$ on a two dimensional cross section Ω_2 of the three dimensional cylindrical domain Ω_3. The original version of 5 POINT STAR was modified slightly to evaluate the right side of the partial differential equation and eliminate the Dirichlet boundary conditions on each cross section. The matrix T_z approximating $L_z u$ is computed by a BILDTZ. The z direction operator, $L_z = -(p(z)u_z)_z + q(z)u$, is specified in the function subprograms ZPCOE and ZQCOE. The z variable is made available to all subprograms through so-called *global common*.

The discrete problem is solved by TPGADI which implements the TPGADI method (4.1). The routine BLDAXY interfaces the output from 5 POINT STAR for input to TPGADI. The acceleration parameters ρ_k are computed to be the eigenvalues of the symmetric positive definite matrix T_z by SETRHO which uses the EISPACK routine IMTQL1 [Smith et al., 1976], [Wilkinson, 1962]. They are used in increasing order [Lynch and Rice, 1968]. The initial iterate, $U^{(0)}$, is always taken to be zero. Although the source for these supplementary programs could be included in the *SUBPROGRAMS* segment of the ELLPACK program, we automatically load them from a separate, precompiled library.