A Tensor Product Generalized ADI Method for Elliptic 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 (nonrectangu-
lar in x and y) with holes.

We approximate the partial differential operators by standard partial difference opera-
tors. 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

\[
(T_x \otimes I + I \otimes A_y)U = F.
\]

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

\[ (A_1 \otimes B_2 + B_1 \otimes A_2)C = F. \]

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 $\rho_k$, $k = 1, 2, \ldots$, the two directional Tensor Product Generalized Alternating Direction Implicit (TPGADI) iteration method is defined by

\[ C^{(0)} \text{ given} \]

\[ \begin{align*}
(A_1 + \rho_k + B_1) \otimes B_2)C^{(k+1)} &= F - [B_1 \otimes (A_2 - \rho_k + B_2)]C^{(k)} \\
B_1 \otimes (A_2 + \rho_k + B_2)]C^{(k+1)} &= F - [(A_1 - \rho_k + B_1) \otimes B_2]C^{(k+1)}.
\end{align*} \]

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 $\lambda_i$ and $\mu_j$, respectively. Then, for a given set of positive acceleration parameters $\rho_k$, $k = 1, 2, \ldots$, 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 \( \Omega_2 \) 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 \( \Omega_3 \) is formed by the tensor product \( \Omega_3 = \Omega_2 \times [a_z, b_z] \). We consider partial differential equations of the form

\[
L_{\Omega_3} u + L_u = f \quad \text{in } \Omega_3 \\
u = g \quad \text{on } \partial \Omega_3,
\]

where

\[
L_{\Omega_3} u = -a(x, y)u_{xx} - b(x, y)u_{yy} + c(x, y)u_x + d(x, y)u_y + e(x, y)u, \quad a, b > 0, \quad e \geq 0,
\]

\[
L_u = -(p(x)u_x)_x + q(x)u, \quad p > 0, \quad q \geq 0,
\]

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

\[
L_{\Omega_2} u = f \quad \text{in } \Omega_2 \\
u = g \quad \text{on } \partial \Omega_2,
\]

where \( f, g \) and \( u \) are functions of \( x \) and \( y \). To solve such problems, we must approximate both the nonrectangular domain \( \Omega_2 \) and the operator \( L_{\Omega_2} \).

For given positive integers \( N_x \) and \( N_y \), the rectangle \( R \) containing \( \Omega_2 \) is subdivided by a rectangular grid defined by the grid lines...
\[ x_i = a_x + l h_x, \quad h_x = \frac{b_x - a_x}{N_x + 1}, \quad \text{and} \quad y_j = a_y + f h_y, \quad h_y = \frac{b_y - a_y}{N_y + 1}. \]

The interior of the domain \( \Omega_2 \) is approximated by \( \tilde{\Omega}_2 \), the set of grid points \((x_i, y_j)\) in the interior of \( \Omega_2 \). The boundary \( \partial \Omega_2 \) is approximated by \( \partial \tilde{\Omega}_2 \), 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 \((x_i, y_j) \in \tilde{\Omega}_2\), 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.

\[
\begin{align*}
(x_i, y_{j+1}) &= (x_i, y_j + h_N) \\
(x_{i-1}, y_j) &= (x_i - h_w, y_j) \\
(x_{i+1}, y_j) &= (x_i + h_E, y_j) \\
(x_i, y_{j-1}) &= (x_i, y_j - h_S)
\end{align*}
\]

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

\[
\begin{align*}
L_{xx} u_{ij} &= \frac{2}{h_w (h_E + h_w)} u_{i-1,j} - \frac{2}{h_w h_E} u_{ij} + \frac{2}{h_E (h_E + h_w)} u_{i+1,j} + O(h_x) \\
L_{yy} u_{ij} &= \frac{-h_N}{h_N (h_S + h_N)} u_{i,j-1} + \frac{h_N - h_N}{h_N h_S} u_{ij} + \frac{h_N}{h_S (h_S + h_N)} u_{i,j+1} + O(h_y) \\
L_{xy} u_{ij} &= \frac{2}{h_S (h_N + h_S)} u_{i,j-1} - \frac{2}{h_S h_N} u_{ij} + \frac{2}{h_N (h_N + h_S)} u_{i,j+1} + O(h_x^2)
\end{align*}
\]
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 
$$\max(h_w, h_E) \leq h_x \quad \text{and} \quad \max(h_S, h_N) \leq h_y.$$ 

There are two distinct types of grid points in $\tilde{\Omega}_2$: regular points which have all four of their nearest neighbors in $\tilde{\Omega}_2$, and irregular points which have one or more nearest neighbors on the boundary $\partial \tilde{\Omega}_2$. 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 $h_S = h_N$ 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 $N_xN_y \rightarrow \infty$, most of the interior grid points in $\tilde{\Omega}_2$ are regular grid points. Thus, the discretization error in the finite difference approximation to $L_u u$ is locally $O(h_x^2) + O(h_y^2)$ at most of the grid points in $\tilde{\Omega}_2$. If $u$ has a bounded fourth derivative in $\Omega$, 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 $(x_i, y_j) \in \tilde{\Omega}_2$, subtracting the boundary values on $\partial \tilde{\Omega}_2$ 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 

$$(3.4) \quad A_{xy} u = f.$$ 

If $\Omega_2$ is rectangular, then $u_{i+N_x, j} = U_{ij}$. The matrix $A_{xy}$ has dimension equal to the number of grid points in $\tilde{\Omega}_2$ which is less than or equal to $N_xN_y$. 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 $\Omega$ by $M+2$ two dimensional cross sections defined by the planes
\[ x_j = a_x + jh_x, \quad h_x = \frac{b_x - a_x}{M + 1}, \quad j = 0, 1, \ldots, M + 1. \]

On each interior two dimensional domain \( \mathcal{D}_2 \otimes x_j \), we approximate \( L_{xy} \) at each point in the interior of \( \mathcal{D}_2 \otimes x_j \) by the partial difference operators (3.3). We approximate \( L_x \) by the standard symmetric finite differences. If we now let \( U_{ij} = u(x_i, y_i, z_j) \), 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

\[
(T_x \otimes I + I \otimes A_{xy})U = F,
\]

where \( T_x \) is the symmetric tridiagonal matrix of order \( M \times M \) defined by

\[
T_x = \text{tridiag}[d_j^-, d_j, d_j^+],
\]

where

\[
d_j^- = -p((j - \frac{1}{2})h_x),
\]

\[
d_j = \frac{p((j - \frac{1}{2})h_x) + p((j + \frac{1}{2})h_x)}{h_x^2} + q(jh_x),
\]

\[
d_j^+ = -p((j + \frac{1}{2})h_x),
\]

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 \( p_k, k = 1, 2, \ldots \), the TPGADI method for the partial difference equations in (3.5) is given by
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
\[(A \otimes I + I \otimes B \otimes I + I \otimes I \otimes C)U = F,
\]
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
\[(5.1) E_j^{(k)} = \prod_{l=1}^{k} \left[ \frac{\lambda_l - \rho_l}{\lambda_l + \rho_l} \frac{\mu_l - \rho_l}{\mu_l + \rho_l} \right] E_j^{(0)}\]
where $\lambda_l$ and $\mu_l$ denote the eigenvalues of $T_x$ and $A_{xy}$, respectively.
Now, for \( h \) sufficiently small, the tridiagonal matrix \( T \) resulting from the \( x \) direction symmetric finite difference approximation to \( L_x \) is symmetric positive definite so that its eigenvalues are real and positive. Since the acceleration parameters \( \rho \) are always taken to be real and positive, it follows that for all \( i, t \)

\[
\left| \frac{\lambda_i - \rho_t}{\lambda_i + \rho_t} \right| \leq \epsilon < 1. \tag{5.2}
\]

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 \( \Omega_2 \) 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 \( \text{Re} \mu_j > 0 \) [Varga, Theorem 1.8]. Thus, for all \( j, t \) we have

\[
\left| \frac{\mu_j - \rho_t}{\mu_j + \rho_t} \right| \leq \epsilon < 1. \tag{5.3}
\]

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

\[
\lim_{k \to -\infty} \| E_i^{(k)} \| = \lim_{k \to -\infty} \prod_{i=1}^{k} \left| \frac{\lambda_i - \rho_t}{\lambda_i + \rho_t} \frac{\mu_j - \rho_t}{\mu_j + \rho_t} E_i^{(p)} \right| = 0
\]

which implies that

\[
\lim_{k \to -\infty} \| E_i^{(k)} \| = 0 \square
\]

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 \( \text{Re} \lambda_i > 0 \) and \( \text{Re} \mu_j > 0 \).

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 $\Omega_2$ 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 $L_{x}u$ 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 $(T_x \otimes I + I \otimes A_{xy})u = F$. Recall that the tridiagonal matrix $T_x$ 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 non-rectangular domain $\Omega_2$, with $N_{xy} \leq N_x N_y$ and $K_{xy} \leq N_y$. 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**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Work</th>
<th>Operation</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_2 = A_{xy} - \rho_{k+1}I$</td>
<td>$\mathbb{R}N_{xy}$</td>
<td>$W_1 = T_x - \rho_{k+1}I$</td>
<td>$\mathbb{R}M$</td>
</tr>
<tr>
<td>$W = (I \otimes W_2)U^{(k)}$</td>
<td>$2MK_{xy}N_{xy}$</td>
<td>$W = (W_1 \otimes I)U^{(k+1)}$</td>
<td>$2MN_{xy}$</td>
</tr>
<tr>
<td>$W = F - W$</td>
<td>$MN_{xy}$</td>
<td>$W = F - W$</td>
<td>$MN_{xy}$</td>
</tr>
<tr>
<td>$W_1 = T_x + \rho_{k+1}I$</td>
<td>$\mathbb{R}M$</td>
<td>$W_2 = A_{xy} + \rho_{k+1}I$</td>
<td>$\mathbb{R}N_{xy}$</td>
</tr>
<tr>
<td>$U^{(k+1)} = (W_1 \otimes I)^{-1}W$</td>
<td>$2M + 3MN_{xy}$</td>
<td>$U^{(k+1)} = (I \otimes W_2)^{-1}W$</td>
<td>$2K_{xy}N_{xy} + 2M_{xy}N_{xy}$</td>
</tr>
</tbody>
</table>

Thus, the work required per iteration for the $z$-direction sweep is $O(2MK_{xy}N_{xy})$ and for the
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” $N = M = N_x = N_y$, $N_{xy} = N_x N_y$ 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 $O(MN_{xy}^2)$ 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; $O(3N^3)$ words if $M = N_x = N_y$, $N_{xy} = N_x N_y$ 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 $p_k$ 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 $\Omega_2$ be the two dimensional circular domain defined by

$$
\Omega_2 = \{(x, y) \mid (x - 4)^2 + (y - 4)^2 < 4\}.
$$
and let $\Omega_3$ be the right circular cylinder defined by $\Omega_3 = \Omega_2 \otimes [0,1]$. We consider the Model Dirichlet Problem

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

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 $h = h_x = h_y = \frac{1}{N + 1}$. The maximum relative error at the grid points interior to $\Omega_3$ 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

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

A logarithmic fit of this timing data shows that $\text{Time} \approx 6.80 \times 10^{-6} N^{-4.45}$ which agrees with the worst case theoretical work estimate of $O(N^5)$ 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 (3.2) 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 $\text{Error} = 3.34 \times 10^{-9} N^{2.85}$.

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 $(T_z \otimes f + I \otimes A_{xy})$ has dimension $24583 \times 24583$ 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 \( \Omega_2 \) be the two dimensional nonrectangular domain given in Figure 3.1 and let \( \Omega_3 \) 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

\[
-\frac{\partial^2 u}{\partial x^2} - (1+xy)\frac{\partial^2 u}{\partial y^2} - \sin(x)u_x - \cos(x)u_y + e^{-x}u_x + (3+z^2)u = f \quad \text{in } \Omega_3
\]

\[
u = g \quad \text{on } \partial \Omega_3,
\]

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 \( h = h_x = h_y = h_z = \frac{1}{N+1} \). The smallest \((N+1)/2\) eigenvalues of \( T \) are used as the acceleration parameters. The results are given in Table 6.3.

**Table 6.3**

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

We obtain Error = \(110h^{2.11}\) which agrees with the theoretical convergence rate of \(O(h^2)\).

The number of iterations is chosen \textit{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 \( \Omega_2 \) be the two dimensional nonrectangular domain with a hole given in Figure 6.2. Let \( \Omega_3 \) be the cylindrical domain defined by \( \Omega_3 = \Omega_2 \otimes [0,1] \). We consider the heat conduction problem defined by

\[
-u_{xx} - u_{yy} - u_{zz} = 0 \quad \text{in} \quad \Omega_3 \\
u = g \quad \text{on} \quad \partial \Omega_3, 
\]  

(6.3)

where \( g \) is defined by

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

The solution \( u \) to (6.3) can be interpreted as the steady state temperature distribution within \( \Omega_3 \), given that the boundary is kept at the temperatures defined by \( g \).

We see from Figure 6.1 that the domain \( \Omega_2 \) 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 \( \Omega_3 \) to the outer cool boundaries of \( \Omega_2 \). 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 6.1 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 $h_x = 1/64$, $h_y = 1/32$, $h_z = 1/16$. 

<table>
<thead>
<tr>
<th>contour value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>
7. References


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 $\Omega_2$ 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 \textit{BOUNDARY} segment:

\begin{verbatim}
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
  0.75,0.75 TO 0.25,0.75 TO 0.25,1.00 TO
  0.00,1.00 TO 0.00,0.50
\end{verbatim}

In this \textit{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.
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.
DISCRETIZE X,Y OPERATOR, BUILD THE RIGHT SIDE TPRSID AND GUESS THE SOLUTION TPKUN

DISCRETIZATION. 5 POINT STAR
FORTRAN.
C INTERFACE 5 POINT STAR OUTPUT FOR INPUT TO TPGADI
C
CALL BLAXY (RICOEF,AXY,IMIDCO,IMNEQ,IMNCO,
A, IIMENK,IIUNDK,NBAND,NBANDL)
C
DISCRETIZE THE Z OPERATOR -(P(Z)U ) + Q(Z)U
Z Z
C
CALL BLAYZ (TZ,NPLNWK)
C
COMPUTE THE ITERATION PARAMETERS RHO(K)
C
IRHO = 1
NITERS = NGRIDZ-2
CALL SETRHO (IRHO,RHO,NGRIDZ,NITERS,TZ,NPLNWK,WORK)
C
SOLVE ( TZ X I + I X AXY ) TPKUN = TPRSID
C
NZBAND = 1
MOCBND = MAX0(NBANDL,NBAND)
CALL TPGADI (TZ,B2Z,NPLNWK,NODZ2,NZBAND,AXY,BXY,IMNEQ,IMNEQN,
A, MOCBND,TPRSID,TPKUN,BZFACT,BXYFACT,WORKMN,WORKNN,
B WORKMN,WORKNN,BWSXYZ,WRKXBY,WORK,NITERS,RHO)
C
EVALUATE SOLUTION AND ERROR ON EACH PLANE
C
DO 20 KZ = 1, NODZ2
   Z = GRIDZ(KZ+1)
   PRINT *, '**** PLANE Z = ', Z
   INITL = 1
   OUTPUT. MAX(TRUE) $ MAX(ERROR)
   CONTINUE
   20 CONTINUE

SUBPROGRAMS.
C COEFFICIENTS OF Z DIRECTION OPERATOR
C
FUNCTION ZPOCE(Z)
ZPOCE = - 1.
RETURN
END
FUNCTION ZPOCE(Z)
ZPOCE = 0.
RETURN
END
C TRUE SOLUTION
C
FUNCTION TRUE(X,Y)
COMMON / TBPZZZ / 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 \( \Omega_2 \) of the three dimensional cylindrical domain \( \Omega_3 \). 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 \( L_zu \) is computed by a BILDTZ. The \( z \) direction operator, \( L_z = -(\rho(z)u_z) + q(z)u \), 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 \( \rho_z \) 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.