1984

A Tensor Product Generalized ADI Method for the Method of Planes

Wayne R. Dyksen

Report Number:

84-494

A TENSOR PRODUCT GENERALIZED ADI METHOD FOR THE METHOD OF PLANES

Wayne R. Dyksen

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

CSD-TR 494
October 1984

ABSTRACT

We consider solving separable second order linear elliptic partial differential equations in three independent variables. If the partial differential operator separates into two factors, one depending on x and y, and one depending on z, then we use the *Method of Planes* to obtain a discrete problem which we write in tensor product form as

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

We apply a new iterative method, the *Tensor Product Generalized Alternating Direction Implicit* method, to solve the discrete problem. We study a specific implementation which uses Hermite bicubic collocation in the xy direction and symmetric finite differences in the z direction. We demonstrate that this method is a fast and accurate way to solve the large linear systems arising from three dimensional elliptic problems.

1980 *Mathematics Subject Classification*. Primary. 65F10, 65M20, 65N35.

A Tensor Product Generalized ADI Method for the Method of Planes

Wayne R. Dyksen

1. Introduction

Let R be a rectangular domain. We consider solving second order linear elliptic problems of the form

$$
(1.1)\qquad
\begin{aligned}
L_{xy}u + L_z u &= f(x,y,z) && \text{in } R \otimes [a_z, b_z] \\
u &= g_a(x,y) && \text{on } R \otimes a_z \\
\alpha(x,y)u + \beta(x,y)u_n &= g(x,y,z) && \text{on } \partial R \otimes (a_z, b_z) \\
u &= g_b(x,y) && \text{on } R \otimes b_z, \\
\alpha(x,y)\beta(x,y) &= 0 && \text{on } \partial R \\
\alpha^2(x,y) + \beta^2(x,y) &> 0 && \text{on } \partial R
\end{aligned}
$$

where

$$
(1.2)\qquad L_z = -(p(z)u_z)_z + q(z)u, \;\; p > 0, \; q \geq 0,
$$

and L_{xy} is a general elliptic operator in x and y. We express a discrete problem in terms of tensor products of matrices resulting from lower dimensional, and hence much simpler, problems. We apply a new iterative technique to solve this discrete problem. We obtain a fast method for solving a large class of elliptic problems in three dimensions.

Section 2 presents a brief introduction to the *Tensor Product Generalized Alternating Direction Implicit* (TPGADI) method. In Section 3 we extend the *Method of Lines* (Jones et al, [18]) to the *Method of Planes* to obtain a discrete problem which we write in tensor product form as

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

We present the TPGADI method for solving such discrete problems in Section 4. We apply the Method of Planes using Hermite bicubics in x and y and finite differences in z in Sections 5, 6 and 7. We consider a specific implementation in Section 8; we show that the Method of Planes together with the TPGADI method give a powerful tool for solving a large class of elliptic problems in three dimensions.

The TPGADI method has been used effectively to solve discrete elliptic problems arising from other discretizations in both two and three dimensions. We have used it in conjunction with the Method of Lines (Dyksen, [8]) and the Hermite bicubic collocation equations (Dyksen, [12]). In three dimensions, we have used it to solve elliptic problems on cylindrical domains with holes (Dyksen, [13]).

2. The Two Directional Tensor Product Generalized ADI Methods

Let A_k and B_k be matrices of order $N_k \times N_k$, 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 for one directional, simpler problems involving A_1, B_1, A_2 and B_2. We use the term *directional* rather than *dimensional* since one direction may encompass more than one dimension.

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

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

$$\left[(A_1 + \rho_{k+1} B_1) \otimes B_2\right] C^{(k+½)} = F - \left[B_1 \otimes (A_2 - \rho_{k+1} B_2)\right] C^{(k)} \tag{2.2}$$

$$\left[B_1 \otimes (A_2 + \rho_{k+1} B_2)\right] C^{(k+1)} = F - \left[(A_1 - \rho_{k+1} B_1) \otimes B_2\right] C^{(k+½)}.$$

The following results are used in subsequent analysis; for details see Dyksen, [12].

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.*

Proof. Let $E^{(k)} = C^{(k)} - C$ denote the error of the k^{th} iterate. A straightforward computation shows that the error $E^{(k)}$ may be expressed in terms of the initial error $E^{(0)}$ as

$$E^{(k)} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \prod_{l=1}^{k} \left[\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \right] E_{ij}^{(0)} p_i \otimes q_j . \tag{2.3}$$

Since by the hypothesis the eigenvalues λ_i and μ_j are positive, it follows from (2.3) that for positive acceleration parameters ρ_l

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

so that

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

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.*

Proof. Let $\lambda_1, \ldots, \lambda_{N_1}$ be the eigenvalues of $B_1^{-1}A_1$ and set $\rho_l = \lambda_l$. Then by (2.3) we have for all i

$$E_{ij}^{(N_1)} = \prod_{l=1}^{N_1} \left[\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \right] E_{ij}^{(0)} = 0. \tag{2.4}$$

Thus,

$$E^{(N_1)} = 0.$$

The analogous argument for N_2 iterations completes the proof □

3. The Tensor Product Formulation of the Method of Planes

Let Ω_s and Ω_c be the unit square and the unit cube, respectively. We consider partial differential equations of the form

$$\begin{aligned} L_{xy}u + L_z u &= f \quad \text{in } \Omega_c \\ u &= 0 \quad \text{on } \partial\Omega_c, \end{aligned} \tag{3.1}$$

where L_z has the form (1.2) and L_{xy} is a general second order linear elliptic partial differential operator in x and y, with the coefficients of u_{xx} and u_{yy} being strictly negative. For simplicity, we first consider homogeneous Dirichlet problems; we consider more general boundary conditions in Section 6.

In order to solve (3.1), we extend the Method of Lines in a natural way to obtain the *Method of Planes*. For a fixed positive integer M, we place in Ω_c the M "horizontal" planes

$$z_j = jh_z, \;\; h_z = \frac{1}{M+1}, \;\; j = 1, \ldots, M.$$

We look for an approximate solution of (3.1) in the form of a set of M functions

$$\{U_1(x,y), U_2(x,y), \ldots, U_M(x,y)\},$$

so that U_j approximates u on plane j; that is,

$$U_j(x,y) \approx u(x,y,z_j).$$

We first discretize the z variable by applying the standard equally spaced, $O(h_z^2)$ symmetric finite difference approximation to L_z so that (1.2) becomes

$$L_z U_j = d_j^- U_{j-1} + d_j U_j + d_j^+ U_{j+1}, \ \ j = 1,\ldots,M \, ,$$

where

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

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

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

Note that $U_0(x,y) = u(x,y,0) = 0$ and $U_{M+1}(x,y) = u(x,y,1) = 0$. The problem of solving (3.1) is now replaced by that of finding suitable functions $U_1(x,y), U_2(x,y), \ldots, U_M(x,y)$, each of which satisfies

$$L_{xy} U_j(x,y) + d_j^- U_{j-1}(x,y) + d_j U_j(x,y) + d_j^+ U_{j+1}(x,y) = f(x,y,z_j), \ \text{in } \Omega_z ,$$

(3.2)

$$U_j(x,y) = 0, \qquad \text{on } \partial\Omega_z .$$

Thus, since the original operator in (3.1) separates into two factors, we are able to reduce a three dimensional problem to a coupled system of two dimensional problems.

We now choose N linearly independent functions $\{\phi_i(x,y)\}_{i=1}^N$ which are twice continuously differentiable and satisfy the boundary conditions. On each plane $z = z_j$, we set

$$U_j(x,y) = \sum_{i=1}^{N} c_{ij} \phi_i(x,y), \ \ j = 1,\ldots,M$$

for some constants c_{ij}. We determine the MN unknowns c_{ij} by choosing N distinct points $\{(x_k,y_k)\}_{k=1}^N$ in Ω_z and collocating the equations in (3.2) at these points. We obtain an $MN \times MN$ system of simultaneous linear equations in the unknowns c_{ij} which can be written in tensor product form as

(3.3) $$(T_z \otimes B_{xy} + I \otimes A_{xy})C = F \, ,$$

where

$$T_z = \text{tridiag}\left[d_j^- \; d_j \; d_j^+\right], \quad j = 1,\ldots,M,$$

I is the $M \times M$ identity matrix,

(3.4)
$$A_{ki} = L_{xy}\phi_i(x_k, y_k), \quad B_{ki} = \phi_i(x_k, y_k), \quad \begin{matrix} k = 1,\ldots,N \\ i = 1,\ldots,N, \end{matrix}$$

$$C_{ij} = c_{ij}, \quad \begin{matrix} i = 1,\ldots,N \\ j = 1,\ldots,M, \end{matrix} \quad \text{and} \quad F_{kj} = f(x_k, y_k, z_j), \quad \begin{matrix} k = 1,\ldots,N \\ j = 1,\ldots,M. \end{matrix}$$

4. The TPGADI Method for the Method of Planes

Convergent TPGADI iterative schemes are derived by adding weighted, approximate values of u to both sides of the original linear system. For the Method of Planes we observe that

$$\begin{aligned} \left[(I \otimes B_{xy})C\right]_{kj} &= \sum_{i=1}^{N} c_{ij}\phi_i(x_k, y_k) \\ &= U_j(x_k, y_k) \\ &\approx u(x_k, y_k, z_j). \end{aligned}$$

Thus, for a given set of positive acceleration parameters ρ_k, $k = 1,2,\ldots$, the TPGADI iterative method for the Method of Planes is

(4.1)
$$\begin{aligned} &C^{(0)} \text{ given} \\ &\left[(T_z + \rho_{k+1}I)\otimes B_{xy}\right]C^{(k+1/2)} = F - \left[I \otimes (A_{xy} - \rho_{k+1}B_{xy})\right]C^{(k)} \\ &\left[I \otimes (A_{xy} + \rho_{k+1}B_{xy})\right]C^{(k+1)} = F - \left[(T_z - \rho_{k+1}I)\otimes B_{xy}\right]C^{(k+1/2)}. \end{aligned}$$

The convergence of (4.1) depends on the eigenvalues of T_z and the generalized eigenvalues of $A_{xy}\mathbf{c} = \lambda B_{xy}\mathbf{c}$. The eigenvalues of T_z are distinct, real and positive. The generalized eigenvalues of $A_{xy}\mathbf{c}=\lambda B_{xy}\mathbf{c}$ are the collocation approximations to the eigenvalues of L_{xy} (de Boor and Swartz, [4], [5]), which, for a large class of operators, are distinct, real and positive, or at least have positive real parts. Thus, we assume that these eigenvalues at least have positive real parts so that Theorem 2.1 applies.

In subsequent applications, the acceleration parameters ρ_k are taken to be the eigenvalues of T_z. If the sum $\lambda+\rho_k$ in (2.4) is bounded away from zero for each generalized eigenvalue λ of $A_{xy}\mathbf{c}=\lambda B_{xy}\mathbf{c}$, then Corollary 2.2 applies so that (4.1) is "convergent" as a direct method. Experience shows that only a very small number of iterations is required to achieve "discretization" accuracy.

5. The Method of Planes with Hermite Bicubics

For given fixed positive integers N_x and N_y, the unit square Ω_x is subdivided with a rectangular, tensor product grid with N_xN_y rectangles. The grid lines, given by

$$x_{n_x}=n_xh_x,\ h_x=\frac{1}{N_x}\ ,\quad \text{and}\quad y_{n_y}=n_yh_y,\ h_y=\frac{1}{N_y}\ ,$$

are the knots of the Hermite bicubics. The Hermite bicubic basis functions are formed as tensor products of the standard one dimensional Hermite cubics. For the case of homogeneous Dirichlet boundary conditions, there are $4N_xN_y$ Hermite bicubic basis functions which we denote by

$$\{\phi_m(x,y)\}_{m=1}^{4N_xN_y}=\{\Psi_0(x),\Phi_1(x),\Psi_1(x),\ldots,\Phi_{N_x-1}(x),\Psi_{N_x-1}(x),\Psi_{N_x}(x)\}$$
$$\otimes\{\Psi_0(y),\Phi_1(y),\Psi_1(y),\ldots,\Phi_{N_y-1}(y),\Psi_{N_y-1}(y),\Psi_{N_y}(y)\}$$

so that

$$U_j(x,y)=\sum_{i=1}^{4N_xN_y}c_{ij}\phi_i(x,y)\approx u(x,y,z_j).$$

The $4N_xN_y$ unknowns c_{ij} are determined by choosing $4N_xN_y$ distinct points $\{(\tau_l,\upsilon_l)\}_{l=1}^{4N_xN_y}$ in Ω_z and collocating the coupled system of partial differential equations in (3.2) at these points. The $4N_xN_y$ collocation points are placed at the four Gauss points of each subrectangle (Houstis, [15]), (Percell and Wheeler, [25]).

The system of linear equations arising from this particular instance of the Method of Planes may be written in tensor product form (3.3) where the x_k, y_k in (3.4) are replaced by τ_l, υ_l. If the collocation points and Hermite bicubic basis functions are ordered in a natural tensor product way, then A_{xy} and B_{xy} have bandwidth $4N_y+2$. For example, the pattern of non-zero elements in A_{xy} and B_{xy} is illustrated in Figure 7.1 for the case $N_x=N_y=3$.

6. Convergence of the Tensor Product Generalized ADI Method

We now apply the TPGADI iterative method (4.1) to the discrete elliptic problem (3.3) resulting from the Method of Planes using Hermite bicubic collocation in the xy direction and symmetric finite differences in the z direction. We establish the convergence of the TPGADI method for the *Model Problem*

$$\begin{aligned} -u_{xx}-u_{yy}-u_{zz} &= f \quad \text{in } \Omega_c \\ u &= 0 \quad \text{on } \partial\Omega_c . \end{aligned}$$

The *Discrete Model Problem* is given by (3.3) where

$$T_z = \text{tridiag}[-h_z^2 \;\; 2h_z^2 \;\; -h_z^2],$$

$$[A_{xy}]_{li} = -\frac{\partial^2\phi_i}{\partial x^2}(\tau_l,\upsilon_l) - \frac{\partial^2\phi_i}{\partial y^2}(\tau_l,\upsilon_l), \quad \text{and} \quad [B_{xy}]_{li} = \phi_i(\tau_l,\upsilon_l).$$

THEOREM 6.1. *For a given set of positive acceleration parameters* ρ_k, $k=1,2,\ldots$, *the TPGADI method* (4.1) *applied to the Discrete Model Problem is convergent.*

Proof. It is well known that the M eigenvalues of T_z are distinct, real and positive. We must show that the $4N_xN_y$ generalized eigenvalues of $A_{xy}\mathbf{c} = \lambda B_{xy}\mathbf{c}$ are distinct, real and positive. We first observe that the matrices A_{xy} and B_{xy} may be written in tensor product form as

$$A_{xy} = A_x \otimes B_y + B_x \otimes A_y \quad \text{and} \quad B_{xy} = B_x \otimes B_y,$$

where A_x, B_x and A_y, B_y are defined by $[A_x]_{lm} = -\Phi_m''(\tau_l)$, $[B_x]_{lm} = \Phi_m(\tau_l)$, and $[A_y]_{jn} = -\Psi_n''(\upsilon_j)$, $[B_y]_{jn} = \Psi_n(\upsilon_j)$. Note that A_x, B_x and A_y, B_y have dimensions $2N_x \times 2N_x$ and $2N_y \times 2N_y$, respectively.

Now, let $\mathbf{p}$, λ and $\mathbf{q}$, μ satisfy $A_x\mathbf{p} = \lambda B_x\mathbf{p}$ and $A_y\mathbf{q} = \mu B_y\mathbf{q}$. Then, by the properties of tensor products of matrices, it follows that $A_{xy}(\mathbf{p} \otimes \mathbf{q}) = (\lambda + \mu)B_{xy}(\mathbf{p} \otimes \mathbf{q})$. Thus, the generalized eigenvalues of $A_{xy}\mathbf{c} = \lambda B_{xy}\mathbf{c}$ are given by the sums of pairs of the generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ and $A_y\mathbf{c} = \lambda B_y\mathbf{c}$.

Dyksen [12, Theorem 5.1] has shown that the $2N_x$ generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ are

$$\lambda_0 = \frac{36}{h_x^2},$$

$$\lambda_{N_x} = \frac{12}{h_x^2},$$

$$\lambda_l^{\pm} = \frac{7d + 9 \mp 6\sqrt{d^2 + 90d + 81}}{h_x^2(4d + 3)}, \quad l = 1, \ldots, N_x - 1,$$

where

$$d = \tan^2\left(\frac{l}{N_x}\,\frac{\pi}{2}\right).$$

Since $d > 0$ for all $l = 1, \ldots, N_x - 1$, and since

$$\tan\left(\frac{l}{N_x}\,\frac{\pi}{2}\right) < \tan\left(\frac{l+1}{N_x}\,\frac{\pi}{2}\right), \quad l = 1, \ldots, N_x - 2,$$

it follows that the $2N_x - 2$ generalized eigenvalues $\lambda_i^{\pm}$ are distinct, real and positive. Hence, the $2N_x$ generalized eigenvalues of $A_x \mathbf{c} = \lambda B_x \mathbf{c}$ are distinct, real and positive. An analogous result holds for $A_y \mathbf{c} = \lambda B_y \mathbf{c}$. The proof now follows directly from Theorem 2.1 □

7. Extensions to More General Boundary Conditions

We first consider a general, second order, linear elliptic partial differential operator L_{xy} on a rectangular domain R together with *uncoupled boundary conditions*; that is, an elliptic problem of the form

$$
\begin{aligned}
L_{xy} u &= f && \text{in } R \\
\alpha(x,y)u + \beta(x,y)u_n &= g(x,y) && \text{on } \partial R \\
\alpha(x,y)\beta(x,y) &= 0 && \text{on } \partial R \\
\alpha^2(x,y) + \beta^2(x,y) &> 0 && \text{on } \partial R.
\end{aligned}
\tag{7.1}
$$

The Hermite bicubic collocation equations resulting from (7.1) contain $4N_xN_y$ equations and unknowns associated with the partial differential operator and $4(N_x + N_y + 1)$ corresponding to the boundary conditions. Since the Hermite bicubic polynomials are the dual basis with respect to function and derivative evaluation at the grid points, the equations and unknowns associated with the boundary conditions uncouple from the resulting linear system. To illustrate this, we show in Figure 7.1 the pattern of non-zero elements in the Hermite bicubic collocation equations arising from (7.1) with $\beta = 0$ using the tensor product ordering of the collocation points (equations) and basis functions (unknowns) for the case $N_x = N_y = 3$. Figure 7.2 shows the same linear system, rearranged with the boundary collocation points and basis functions ordered first. The first 28 equations and unknowns are associated with the boundary conditions, and uncouple from the 64×64 linear system, leaving the 36 equations and unknowns associated with the interior collocation points. The unknowns associated with the boundary conditions can be computed and eliminated from the resulting linear system entirely, during the discretization phase.

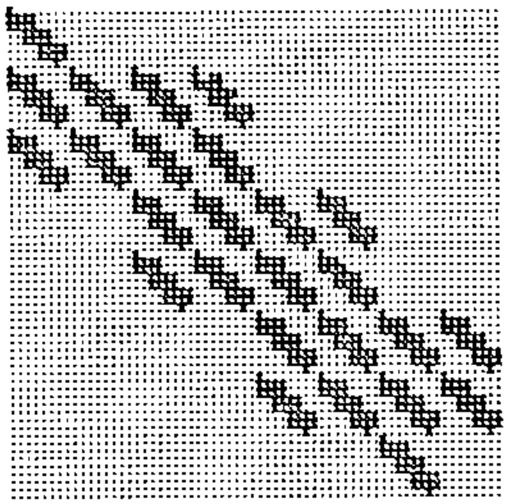


Figure 7.1 The pattern of non-zero elements in the Hermite bicubic collocation equations with the tensor product ordering for the case $N_x, N_y = 3$

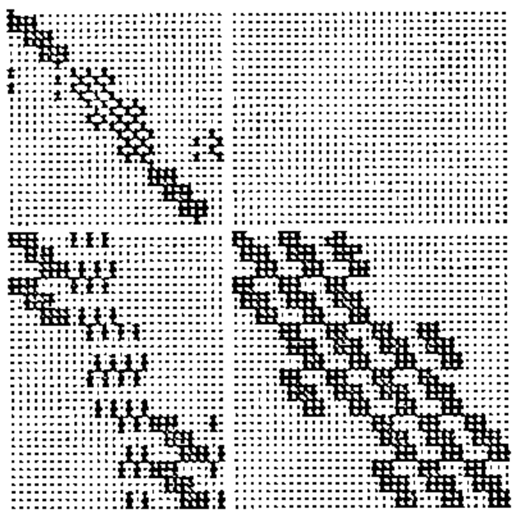


Figure 7.2 The pattern of non-zero elements in the Hermite bicubic collocation equations with the boundary equations and unknowns ordered first for the case $N_x, N_y = 3$

Now, we use this idea with the Method of Planes using Hermite bicubic collocation in the xy direction and symmetric finite differences in the z direction together with the TPGADI method to solve elliptic problems of the form

$$\begin{aligned} L_{xy}u + L_z u &= f(x,y,z) && \text{in } R \otimes [a_z, b_z] \\ u &= g_a(x,y) && \text{on } R \otimes a_z \\ \alpha(x,y)u + \beta(x,y)u_n &= g(x,y,z) && \text{on } \partial R \otimes (a_z, b_z) \\ u &= g_b(x,y) && \text{on } R \otimes b_z, \end{aligned}$$

where L_{xy} and L_z are defined in (3.1), and α and β satisfy

$$\alpha(x,y)\beta(x,y) = 0 \quad \text{and} \quad \alpha^2(x,y) + \beta^2(x,y) > 0 \text{ on } \partial R.$$

If we apply the Method of Planes in a straightforward manner, including the equations of the boundary conditions in the discrete problem, and if we order the equations and unknowns associated with the boundary conditions first, then the matrix in the linear system has the general form

$$\begin{bmatrix} D & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ D_1 & A_{xy} + d_1 B_{xy} & d_1^+ B_{xy} & & & & 0 \\ D_2 & d_2^- B_{xy} & A_{xy} + d_2 B_{xy} & d_2^+ B_{xy} & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ D_{M-1} & & & & d_{M-1}^- B_{xy} & A_{xy} + d_{M-1} B_{xy} & d_{M-1}^+ B_{xy} \\ D_M & 0 & & & & d_M^- B_{xy} & A_{xy} + d_M B_{xy} \end{bmatrix}.$$

After solving for the boundary unknowns ("inverting" D) and eliminating the D_j, the block tridiagonal matrix involving A_{xy} and B_{xy} can be split up into tensor product form giving

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

It is important that the D_j are eliminated **without** modifying the block tridiagonal matrix involving A_{xy} and B_{xy}; otherwise, its tensor product form would be destroyed.

In practice, the equations of the boundary conditions are eliminated from the discrete problem in a way similar to that for the Method of Lines. The Dirichlet boundary conditions on $R \otimes a_x$ and $R \otimes b_x$ are subtracted from the right side F in the usual way for finite differences. The $4(N_x + N_y + 1)$ "precalculated" unknowns c_{lJ} associated with the uncoupled boundary conditions are eliminated for each plane $R \otimes z_J$. Each unknown c_{lJ} interacts with three planes, $z = z_{J-1}, z_J, z_{J+1}$. Thus, to eliminate c_{lJ} at the l^{th} collocation point (τ_l, ν_l) on the J^{th} plane $z = z_J$, the right side F must be modified on each of these three planes by

$$F_{l,J+1} \leftarrow F_{l,J+1} - d_J^{+} c_{lJ} \phi_l(\tau_l, \nu_l)$$

$$F_{lJ} \leftarrow F_{lJ} - c_{lJ}[L_{xy}\phi_l(\tau_l, \nu_l) + d_J \phi_l(\tau_l, \nu_l)]$$

$$F_{l,J-1} \leftarrow F_{l,J-1} - d_J^{-} c_{lJ} \phi_l(\tau_l, \nu_l).$$

8. Computer Implementation and Performance Evaluation

We use some of the advanced features of ELLPACK† to graft an experimental version of the Method of Planes and the TPGADI method into ELLPACK (Rice and Boisvert, [27]). ELLPACK automatically discretizes the xy-direction operator by generating the Hermite bicubic collocation equations and computing the unknowns associated with the boundary conditions (Houstis et al, [16], [17]). ELLPACK "thinks" that it's solving a two dimensional problem. We supplement ELLPACK with Fortran subprograms which discretize the z-direction operator and solve the resulting discrete problem using the TPGADI method. A sample ELLPACK program is given in Appendix A.

The computational complexity of the TPGADI iterative method (4.1) derived for the Method of Planes can be estimated from the results given by Dyksen [12]. We assume that $h_z = \frac{1}{M+1}$ and that $N_x = N_y = N$ so that $h_x = h_y = h = \frac{1}{N}$. Thus, in $(T_z \otimes B_{xy} + I \otimes A_{xy})$, T_z

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

has dimension $M \times M$, and A_{xy} and B_{xy} each has dimension $4N^2 \times 4N^2$ with approximate bandwidth $2N$. We give in Table 8.1 the work required to compute the z-direction and the xy-direction sweep of the TPGADI method.

Table 8.1
Work to compute one sweep of the TPGADI method for the Method of Planes

z-direction sweep		xy-direction sweep	
Operation	Work	Operation	Work
$W_2 = A_{xy} - \rho_{k+1} B_{xy}$	$16N^3$	$W_1 = T_z - \rho_{k+1} I$	$\frac{1}{2}M$
$W = (I \otimes W_2) C^{(k)}$	$16MN^3$	$W = (W_1 \otimes B_{xy}) C^{(k+\frac{1}{2})}$	$8MN^2 + 16MN^3$
$W = F - W$	$4MN^2$	$W = F - W$	$4MN^2$
$W_1 = T_z + \rho_{k+1} I$	$\frac{1}{2}M$	$W_2 = A_{xy} + \rho_{k+1} B_{xy}$	$16N^3$
$C^{(k+\frac{1}{2})} = (W_1 \otimes B_{xy})^{-1} W$	$2M + 32N^4 + 12MN^2 + 24MN^3$	$C^{(k+1)} = (I \otimes W_2)^{-1} W$	$32N^4 + 24MN^3$

The work required per iteration for the each direction sweep is $O(32N^4 + 40MN^3)$ so that the total work per iteration is $O(64N^4 + 80MN^3)$. Since the TPGADI iterative method can be a direct method (depending on the choice of the acceleration parameters) in $\min(M, 4N^2)$ iterations, it follows that the total work is $O\left((64N^4 + 80MN^3)\min(M, 4N^2)\right)$.

The matrix $(T_z \otimes B_{xy} + I \otimes A_{xy})$ has dimension $4MN^2 \times 4MN^2$ and approximate bandwidth $4N^2$ so that the work to factor it using band Gauss elimination with partial pivoting is $O(128MN^6)$. Since the xy direction collocation discretization error is $O(h^4)$, whereas the z direction finite difference discretization error is $O(h_z^2)$, one would usually require M to be much larger than N. For example, if $M = N^2$, then the work to solve $(T_z \otimes B_{xy} + I \otimes A_{xy})C = F$ by the TPGADI method and by Gauss elimination is $O(80N^7)$ and $O(128N^8)$, respectively. Hence, even if one uses the TPGADI method as a **direct method**, it is asymptotically much faster than the simple approach of applying band Gauss elimination.

Our implementation of the TPGADI method requires $O(12MN^2+24N^3)$ words of memory, whereas $O(48MN^4)$ words are required just to store and factor $(T_z \otimes B_{xy} + I \otimes A_{xy})$ using band Gauss elimination. If $M = N^2$, then these estimates simplify to $O(12N^4)$ and $O(48N^6)$, respectively. Note that the memory used by the TPGADI method is only three times the number of unknowns, $4N^4$, and hence represents a considerable savings for three dimensional 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 eigenvalues of the symmetric tridiagonal matrix T_z as computed by the EISPACK routine IMTQL1 (Smith et al, [28]) (Wilkinson, [30]) are used as the acceleration parameters ρ_k; the time required to compute these eigenvalues is always included in timings of the TPGADI method. The acceleration parameters are used in increasing order (Lynch and Rice, [21]), and the initial iterate, $C^{(0)}$, is always taken to be zero.

EXAMPLE 8.1. Performance of the TPGADI Method with M and N Varied

The three dimensional Model Dirichlet Problem is defined by

$$\begin{aligned} -u_{xx} - u_{yy} - u_{zz} &= f \quad \text{in } \Omega_c \\ u &= g \quad \text{on } \partial\Omega_c, \end{aligned}$$

where f and g are given functions of x, y and z. We solve the Model Dirichlet Problem in which f and g are chosen so that $u(x,y,z) = x^3y^3z^3$. We compute the maximum relative error at the grid points on every interior plane. The results are summarized in Table 8.2.

Table 8.2
The Method of Planes and the TPGADI method applied to the Model Dirichlet Problem

M	N	Number of Unknowns	Number of Iterations	Solution Time (Secs)	Maximum Error
4	4	256	4	5.92	6.5484e-08
8	8	2048	8	149.77	6.5772e-07
12	12	6912	12	914.41	4.1562e-06
16	16	16384	16	4127.95	4.3164e-06

A logarithmic fit of this timing data gives Time $\approx 0.00879 N^{4.69}$ which agrees with the theoretical work estimate of $O(144 N^5)$ operations. This method is theoretically exact for this problem, and we see that machine round-off is achieved and that the round-off errors do not grow significantly as M and N increase.

Example 8.2. The Method of Planes and the TPGADI Method applied to Problem 18

We prove in Section 6 that the TPGADI method derived for the Method of Planes converges if applied to the Discrete Model Problem. We now solve a discrete problem arising from a more general elliptic operator. We extend to three dimensions the two dimensional elliptic operator of **Problem 18** of the population of partial differential equations of Rice et al [26]; in particular, we consider

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

where f and g are chosen so that $u(x,y,z) = \sin(2\pi x)\cos(4\pi y)e^z$. We use $h_z = h^2$ to balance the errors between the xy and z direction discretizations. The results are given in Table 8.3.

Table 8.3
The Method of Planes and the TPGADI method with $2N$ iterations applied to Problem 18

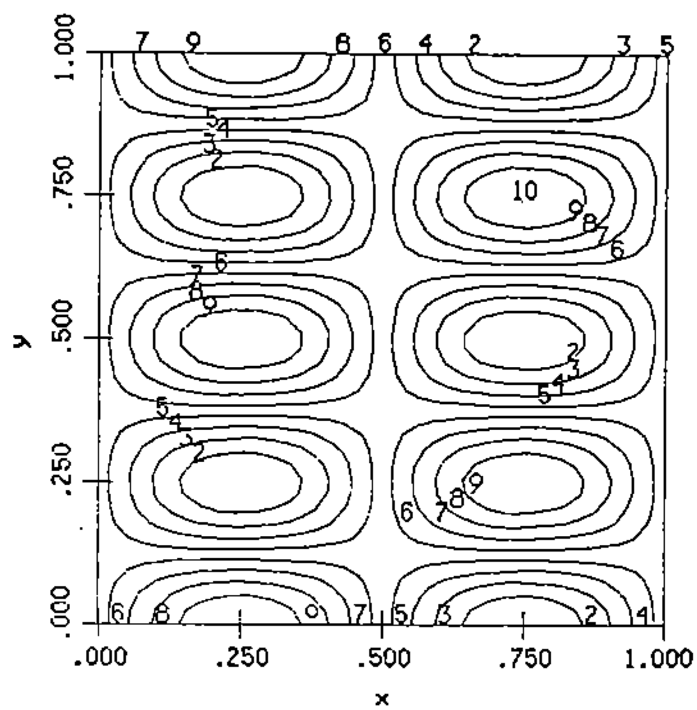
M	N	Number of Unknowns	Number of Iterations	Solution Time (Secs)	Maximum Error
15	4	960	8	27.00	1.4691e-01
35	6	5040	12	245.10	1.8959e-02
63	8	16128	16	1499.40	7.1656e-03
143	12	82368	24	14298.65	1.3471e-03

For this case, the experimental rate of convergence is Error $\approx 45h^{4.21} \approx 45h_z^{2.11}$ which agrees with the theoretical rate of convergence, $O(h^4)$. We give in Figure 8.1 a typical contour plot of a cross section of the computed solution and the error.

Although the solution time for the case $N = 12$ and $M = 143$ is almost four hours, moderate accuracy is achieved by taking a relatively small number of iterations (24) compared to the number of unknowns (82368). This is because each eigenvalue of T_z annihilates 576 components of the error. Herein lies the power of the TPGADI method. By contrast, it would take approximately 12 days to solve this problem using band Gauss elimination.

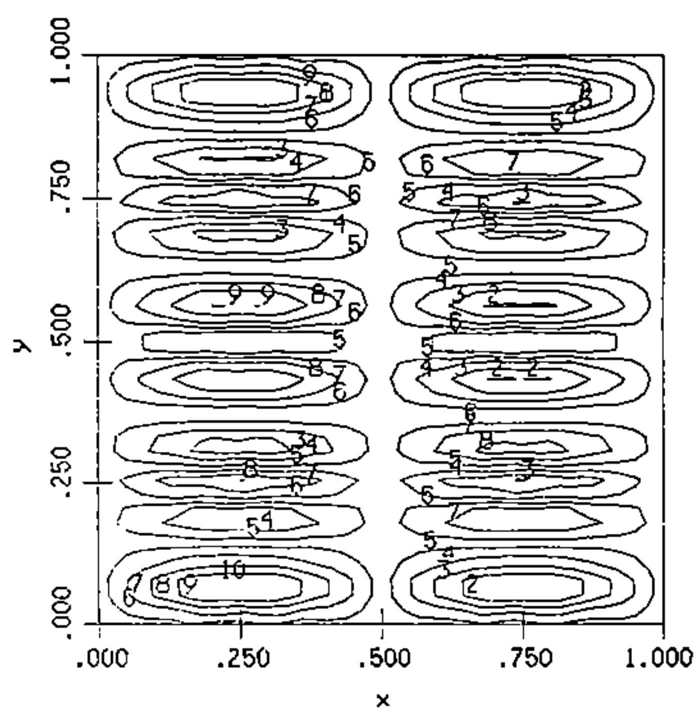
The memory efficiency of the TPGADI method is striking for three dimensional problems. For example, with $N = 12$ and $M = 143$, our implementation of the Method of Planes and the TPGADI method requires approximately 300,000 words of computer memory. By contrast, the number of words required in this case just to store $(T_z \otimes B_{xy} + I \otimes A_{xy})$ to factor it using band Gauss elimination is on the order of 140,000,000.

We believe that our implementation could be made more efficient by using a discretization method in the z direction which achieves an $O(h_z^4)$ discretization error such as a HODIE type method (Boisvert, [3]), (Lynch and Rice, [22]). We could then reduce M from N^2 to N, so that the work reduces by a factor of N^2 from $O(80N^7)$ to $O(80N^5)$.



u contours

contour	value
1	-1.65e+00
2	-1.29e+00
3	-9.18e-01
4	-5.51e-01
5	-1.83e-01
6	1.84e-01
7	5.51e-01
8	9.18e-01
9	1.29e+00
10	1.65e+00



error contours

contour	value
1	-1.87e-02
2	-1.46e-02
3	-1.04e-02
4	-6.26e-03
5	-2.11e-03
6	2.04e-03
7	6.20e-03
8	1.04e-02
9	1.45e-02
10	1.87e-02

Figure 8.1 A contour plot of a cross section on the plane $z = 1/2$ of the computed solution and the error for Problem 18 for the case $h = 1/8$ and $h_r = 1/64$

EXAMPLE 8.3. The Method of Planes and the TPGADI Method Applied to a Problem in Heat Conduction

Let us consider the *steady state* temperature distribution on the unit cube Ω_c with a heat source at the center of the cube together with certain boundary conditions. In particular, we consider

$$-u_{xx}-u_{yy}-u_{zz}=6400x(1-x)y(1-y)[z(1-z)]^2 \text{ in } \Omega_c$$

$$\begin{aligned} u&=0 \text{ on } z=0,1 \\ \chi_r(y)u+\chi_l(y)u_x&=0 \text{ on } x=0 \\ \chi_r(1-y)u+\chi_l(1-y)u_x&=0 \text{ on } x=1 \\ \chi_r(x)u+\chi_l(x)u_y&=0 \text{ on } y=0 \\ \chi_r(1-x)u+\chi_l(1-x)u_y&=0 \text{ on } y=1, \end{aligned} \tag{8.1}$$

where χ_r and χ_l are defined by

$$\chi_r(s)=\begin{cases}0 & s<\tfrac{1}{2}\\ 1 & s\ge\tfrac{1}{2}\end{cases} \quad\text{and}\quad \chi_l(s)=\begin{cases}1 & s\le\tfrac{1}{2}\\ 0 & s>\tfrac{1}{2}\end{cases}.$$

The boundary conditions on a cross section of the cube are illustrated below in Figure 8.2.

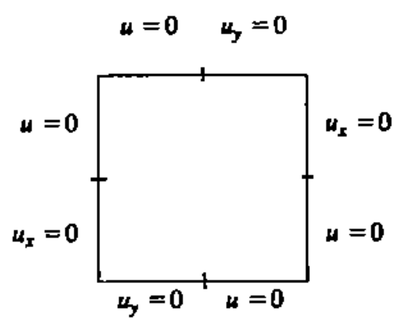


Figure 8.2 The boundary conditions on a cross section of the unit cube for a heat conduction problem

The boundary conditions in (8.1) correspond physically to perfectly insulating two opposite "corners" of the cube while maintaining the top ($z=1$) and bottom ($z=0$) and the other two corners at 0°. These boundary conditions are of the uncoupled type discussed in Section 7 so that our implementation of the Method of Planes with INTERIOR COLLOCATION and the TPGADI method applies.

We solve (8.1) using $h = 1/8$, $h_z = 1/64$, giving 16,128 unknowns to compute. We use 16 iterations of the TPGADI method resulting in a solution time of 1538.87 seconds. Figure 8.3 shows contour plots of the computed solution on the planes $z = 1/4$ and $z = 1/2$. The temperature is greater on the plane $z = 1/2$ which is closer to the heat source. Moreover, the heat flows out of the "upper right" and "lower left" corners of the cross sections, and not through the insulated corners.

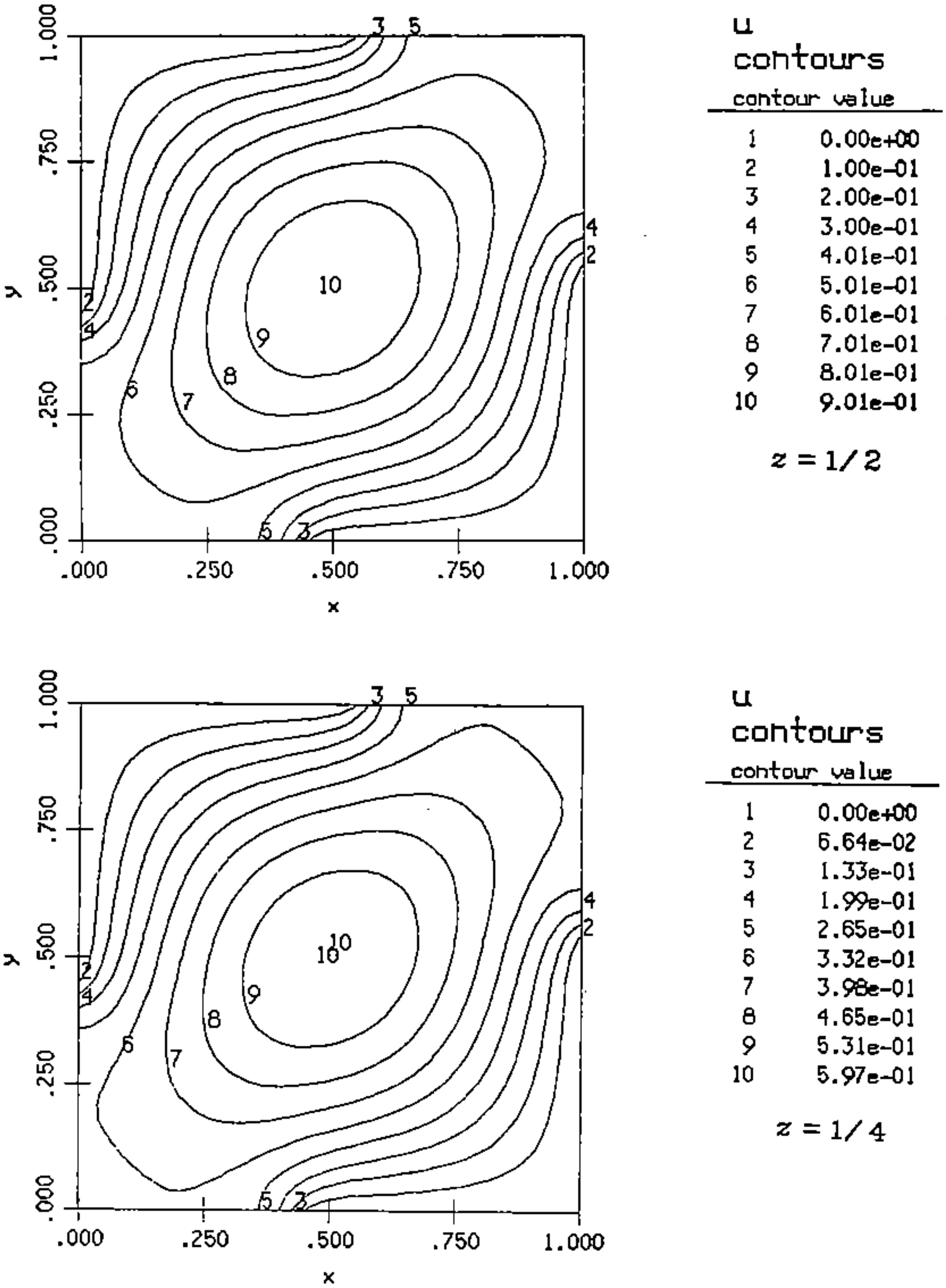


Figure 8.3 A contour plot of the cross section on the planes $z = 1/2$ (top) and $z = 1/4$ (bottom) of the computed solution to a heat conduction problem for the case $h = 1/8$ and $h_z = 1/64$

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

5. C. de Boor and B. Swartz, "Collocation approximation to eigenvalues of an ordinary differential equation: The principle of the thing", *Math. Comp.*, 35(1980), 679-694.

6. C. de Boor and B. Swartz, "Collocation approximation to eigenvalues of an ordinary differential equation: Numerical illustrations", *Math. Comp.*, 36(1981), 1-19.

7. 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.

8. 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.

9. W. R. Dyksen and J. R. Rice, "A new ordering scheme for the Hermite bicubic collocation equations", in *Elliptic Problem Solvers III*, (G. Birkhoff and A. Schoenstadt, eds.), Academic Press, 1984, pp. 467-480.

10. W. R. Dyksen and J. R. Rice, "Scale factors for the Hermite bicubic collocation equations", to appear, *Advances in Computer Methods for Partial Differential Equations-V*, (R. Vichnevetsky and R. Stepleman, eds.), IMACS, New Brunswick, NJ, 1984.

11. W. R. Dyksen, R. E. Lynch, J. R. Rice and E. N. Houstis, "The performance of the collocation and Galerkin methods with Hermite bi-cubics", *SIAM J. Numer. Anal.*, 21(1984), 695-715.

12. 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.

13. W. R. Dyksen, "A tensor product generalized ADI method for elliptic problems on cylindrical domains", Purdue University, Computer Science Department Report CSD-TR 495, October 1984, submitted for publication.

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

15. E. N. Houstis, "Collocation methods for linear elliptic problems", *BIT*, 18(1978), 301-310.

16. E. N. Houstis, W. F. Mitchell and J. R. Rice, "Algorithms INTCOL and HERMCOL: Collocation on rectangular domains with bicubic Hermite polynomials', Purdue University, Computer Science Department Report CSD-TR 445, June 1983.

17. E. N. Houstis, W. F. Mitchell and J. R. Rice, "Collocation software for second order elliptic partial differential equations", Purdue University, Computer Science Department Report CSD-TR 446, June 1983.

18. D. J. Jones, J. C. South and E. B. Klunker, "On the numerical solution of elliptic partial differential equations by the method of lines", *J. Comput. Phys.*, 9(1972), 496-527.

19. 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.

20. 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.

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

22. 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.

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

24. 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.

25. P. Percell and M. F. Wheeler, "A C^1 finite collocation method for elliptic equations", *SIAM J. Numer. Anal.*, 17(1980), 605-622.

26. 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.

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

28. 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.

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

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

10. Appendix A - A Sample ELLPACK Program

The experimental version of the Method of Planes and the TPGADI method is implemented within the ELLPACK system (Rice and Boisvert, [27]). We use an ELLPACK program supplemented with Fortran subprograms. A sample ELLPACK program is given Figure 10.1 for the Poisson problem on the unit cube. Note that ELLPACK "thinks" that we are solving a two dimensional problem. The *discretization* module INTERIOR COLLOCATION generates the Hermite bicubic collocation equations and computes the unknowns associated with the boundary conditions (Houstis et al, [16], [17]). The ELLPACK version of INTERIOR COLLOCATION was modified to compute the Gramian matrix Bxy, and to eliminate the equations of the boundary conditions **on each plane**. The former is trivial whereas the latter is rather substantial. The z direction operator, $Lz = -(p(z)u_z)_z + q(z)u$, is specified in the function subprograms ZPCOE and ZQCOE. The z variable is made available to all subprograms through *global common*. The matrix Tz approximating Lzu 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 discrete problem is solved by TPGADI which implements the TPGADI method (4.1). The routine BLDAXY interfaces the output from INTERIOR COLLOCATION for input to TPGADI. The acceleration parameters ρ_k are computed to be the eigenvalues of the symmetric positive definite matrix Tz by SETRHO which uses the EISPACK routine IMTQL1 (Smith et al, [28]), (Wilkinson, [30]). They are used in increasing order (Lynch and Rice, [21]). The initial iterate, $U^{(0)}$, is always taken to be zero. Although the source for these supplementary program could be included in the *SUBPROGRAMS* segment of the ELLPACK program, we automatically load them from a separate, precompiled library.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                               *
*    SAMPLE ELLPACK PROGRAM FOR THE METHOD OF PLANES WITH INTERIOR              *
*    COLLOCATION AND THE TPGADI ITERATIVE METHOD                                *
*                                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

GLOBAL.
      COMMON / TPZZZZ / Z

DECLARATIONS.
      PARAMETER (NGDZMX = 10)
      PARAMETER (NPLNMX = NGDZMX-2)
      PARAMETER (NBDMAX = 2*$I1NGRY + 3)
      PARAMETER (NCOLMX = 2*NBDMAX + 1)
      PARAMETER (NWXLXY = $I1MNEQ*(NBDMAX + 1))
      COMMON / TPRSID / TPRSID($I1MNEQ,NPLNMX)
      COMMON / TPUNKN / TPUNKN($I1MNEQ,NPLNMX)
      COMMON / TPBUNK / TPBUNK(4,$I1MNEQ,NPLNMX)
      COMMON / TPGRAM / TPGRAM($I1MNEQ,$I1MNCO)
      COMMON / TPTZZZ / TZ(NPLNMX,2)
      COMMON / GRIDZZ / GRIDZ(NGDZMX)
      REAL
     A      TZ(NPLNMX,2),
     B      AXY($I1MNEQ,NCOLMX),
     C      BXY($I1MNEQ,NCOLMX),
     D      BXYFCT($I1MNEQ,NCOLMX),
     E      WORKMM(NPLNMX,2),
     F      WORKNN($I1MNEQ,NCOLMX),
     G      WORKMN(NPLNMX,$I1MNEQ),
     H      WRKBXY(NWXLXY),
     I      WORK(NWXLXY),
     J      RHO(NPLNMX)

EQUATION.  - UXX - UYY = -6.0 * (X*Y**3*Z**3 + X**3*Y*Z**3 + X**3*Y**3*Z)

BOUNDARY.  U = TRUE(X,Y) ON X = 0.0
                         ON X = 1.0
                         ON Y = 0.0
                         ON Y = 1.0

GRID.      5 X POINTS $ 5 Y POINTS

FORTRAN.
C
C     DEFINE Z GRID
C
      AZ = 0.0
      BZ = 1.0
      NGRIDZ = 5
      HZ = (BZ-AZ)/(NGRIDZ-1)
      NGRDZD = NGDZMX
      NGDZM2 = NGRIDZ-2
      GRIDZ(1) = AZ
      DO 10 KZ = 2, NGRIDZ-1
         GRIDZ(KZ) = AZ + (KZ-1)*HZ
   10 CONTINUE
      GRIDZ(NGRIDZ) = BZ
```

Figure 10.1 Sample ELLPACK program implementing the Method of Planes with INTERIOR COLLOCATION and the TPGADI iterative method

```
C
C     DISCRETIZE X,Y OPERATOR, BUILD THE RIGHT SIDE, TPRSID
C
DISCRETIZATION.  INTERIOR COLLOCATION
INDEXING.        NATURAL
FORTRAN.
C
C     DISCRETIZE THE Z OPERATOR -(P(Z)U ) + Q(Z)U
C                                     Z Z
C
      CALL BILDTZ (TZ,NPLNMX)
C
C     INTERFACE INTERIOR COLLOCATION OUTPUT FOR INPUT TO TPGADI
C
      CALL BDABXY (R1COEF,TPGRAM,AXY,BXY,I1IDCO,I1MNEQ,I1MNCO,
     A             I1ENDX,I1UNDX,NBANDU,NBANDL)
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     GUESS TPUNKN
C
      CALL GUESSC (TPUNKN,MXNEQ,NUMBEQ,NGDZM2)
C
C     SOLVE ( TZ X BXY  +  I X AYX ) 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 THE 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**3 * Y**3 * Z**3
      RETURN
      END
END.
```

Figure 10.1 (Continued)