A.D.I. METHODS FOR SPLINE COLLOCATION DISCRETIZATIONS OF ELLIPTIC PDEs

P. Tsompanopoulou

University of Crete
Department of Mathematics
Iraklio, Greece

February 13, 1995

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Lately, the Collocation methods based on Hermite splines ([18], [29], [1], [21], [22]) or cubic splines ([31], [1], [20], [16]) has been proved to be very important and powerful discretization tools for the solution of elliptic Partial Differential Equations (PDEs). It has been seen that both classes of methods can achieve optimal order of convergence and increased computational efficiency. Recently, a series of papers ([11], [12], [13], [2], [5]) have been devoted to the analysis, the implementation and the performance evaluation of Alternating Direction Implicit (ADI) methods applied to linear algebraic systems which arise from Hermite cubic Collocation discretizations of elliptic PDEs in two and three dimensions.

This study is the first attempt to propose, analyze and implement ADI schemes for cubic spline Collocation discretizations. Specifically, we formulate Alternating Direction Implicit Spline Collocation (ADISC) methods based on cubic spline piecewise polynomials for approximating the solution u of the second order elliptic PDE

$$Lu \equiv \sum_{i=1}^{k} \alpha_i \frac{\partial^2 u}{\partial x_i^2} + \gamma u = f \quad \text{in} \quad \Omega, \tag{1a}$$

subject to Dirichlet or Neumann boundary conditions

$$Bu = g \text{ on } \partial\Omega \equiv \text{boundary of } \Omega \tag{2a}$$

where Bu is u (or $\frac{\partial u}{\partial x_i}$, the outward normal derivative of u), $\Omega \equiv$

$\prod_{i=1}^{k} \otimes [a_i, b_i]$ is a rectangular domain in $\mathbb{R}^k$ and $\alpha_i (< 0)$, $\gamma (\geq 0)$, f and g are functions of k variables. Although the formulation and the implementation of the proposed ADISC schemes are given for the above general PDE problem we carry out most of the convergence analysis only for the three-dimensional Helmholtz problem, with Dirichlet boundary conditions and constant coefficients, that is

$$\begin{aligned} Lu &\equiv \sum_{i=1}^{3} D_{x_i}^2 u + \gamma u = f \quad \text{in } \Omega & (1b) \\ u &= g \quad \text{on } \partial\Omega \, . & (2b) \end{aligned}$$

The two-dimensional ADISC case can be treated similarly and it will not be presented here.

The rest of this paper is organized as follows. In Chapter 2 we summarize some properties of tensor products (Kronecker products) of matrices and give a brief introduction on the tensor product analysis of PDEs. In Chapter 3 we briefly define the cubic spline Collocation discretization methods and derive the linear algebraic system of equations which arises from these discretizations. The detailed derivation of the ADI schemes associated with these linear systems is given in Chapter 4 where the convergence analysis and the complexity analysis of the proposed ADISC methods are presented also. The results of our extensive numerical experiments are given in Chapter 5. Chapter 6 contains a summary of our results and some concluding remarks.

Chapter 2

Tensor Product Analysis of PDEs

A detailed introduction to the theory of tensor products of matrices is given in [17]. The use of this theory for the analysis of PDEs was first proposed in [26]. Algorithms that manipulate efficiently tensor products of matrices were given in [6]. In the three sections that follow we briefly review the above papers and extend the notation and certain theoretical results for our needs.

2.1 Definitions and properties of tensor products

Consider two functions x and y,

$$x : [a_1, b_1] \to \mathbb{R} \quad \text{and} \quad y : [a_2, b_2] \to \mathbb{R}.$$

We define the tensor product of x and y, $x \otimes y$, as

$$x \otimes y : [a_1, b_1] \times [a_2, b_2] \to \mathbb{R}$$

$$(x \otimes y)(s, t) = x(s)y(t), \quad (s, t) \in [a_1, b_1] \times [a_2, b_2].$$

Consider $\mathrm{x} \in \mathbb{R}^N$ and $\mathrm{y} \in \mathbb{R}^L$ $(N, L \in \mathbb{N})$

$$\mathrm{x} = (x_1, x_2, \ldots, x_N)^T \quad \text{and} \quad \mathrm{y} = (y_1, y_2, \ldots, y_L)^T$$

and view x and y as restrictions of the functions x and y, respectively. We define the tensor product $\mathrm{x} \otimes \mathrm{y}$ of the two vectors x and y as

$$(\mathrm{x} \otimes \mathrm{y})_{n,l} = x_n y_l \ , \quad n = 1, \ldots, N, \quad l = 1, \ldots, L$$

which can be expressed in the so-called natural ordering form as

$$\mathrm{x} \otimes \mathrm{y} = \begin{pmatrix} x_1 \mathrm{y} \\ x_2 \mathrm{y} \\ \vdots \\ x_N \mathrm{y} \end{pmatrix} \in \mathbb{R}^{NL}.$$

Let now $A = \{a_{mn}\}$ and $B = \{b_{kl}\}$ be matrices of order $M \times N$ and $K \times L$ respectively. The tensor product of A and B is the following $MK \times NL$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1N}B \\ a_{21}B & a_{22}B & \cdots & a_{2N}B \\ & \vdots & & \\ a_{M1}B & a_{M2}B & \cdots & a_{MN}B \end{pmatrix}.$$

Assuming now that the sizes of the matrices, involved below, are compatible with the indicated operations, we summarize some of the properties of tensor products which are useful in analyzing ADI schemes:

$$(A_1 + A_2) \otimes B = A_1 \otimes B + A_2 \otimes B$$

$$A \otimes (B_1 + B_2) = (A \otimes B_1) + (A \otimes B_2)$$

$$(A_1 \otimes A_2)(B_1 \otimes B_2) = A_1 B_1 \otimes A_2 B_2$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)^T = A^T \otimes B^T.$$

Finally, from the above it can be easily seen that if x and y are the eigenvectors of the matrices A and B with eigenvalues λ and μ, respectively, then $\mathrm{x} \otimes \mathrm{y}$ and $\lambda\mu$ is the eigenvector and the eigenvalue of the matrix $A \otimes B$, respectively. It is worth to point out that the above statement does not require commutativity of the matrices A and B. A very useful Lemma, to be used later for the eigenvalue analysis of our ADI Collocation methods, is stated as follows.

Lemma 2.1 *Let the $N_i \times N_i$ matrices A_i, $i = 1, \ldots, k$, possess complete sets of linearly independent eigenvectors $\mathrm{y}^{(i,j_i)}$ with corresponding eigenvalues $\lambda_{A_i}^{(j_i)}$ $(j_i = 1, \ldots, N_i,\ i = 1, \ldots, k)$. Then the matrix $A \equiv A_1 \otimes A_2 \otimes \cdots \otimes A_k$ possesses the $\prod_{i=1}^{k} N_i$ linearly independent eigenvectors $y^{(\underline{j})} \equiv \mathrm{y}^{(1,j_1)} \otimes \mathrm{y}^{(2,j_2)} \otimes \cdots \otimes \mathrm{y}^{(k,j_k)}$, where $\underline{j} \equiv (j_1, \ldots, j_k)$ and $j_i = 1, \ldots, N_i$, with corresponding eigenvalues $\lambda_A^{(\underline{j})} \equiv \lambda_{A_1}^{(j_1)} \lambda_{A_2}^{(j_2)} \ldots \lambda_{A_k}^{(j_k)}$.*

Proof: See Lemma 3.5 in [15].

2.2 Tensor products applied to elliptic PDEs

In this section we present a few results obtained by applying a classical and fundamental idea to the analysis of PDEs. The idea is that certain multidimensional problems can be solved by solving a few one–dimensional problems and is the basis of the classical method of separation of variables of mathematical physics. In the case of PDEs, this idea leads to tensor product analysis of the matrices involved. Below we present the basics of such an analysis in two dimensions.

We consider the second order linear PDE

$$L_x u + L_y u = f \quad \text{in } \Omega \equiv [0,1] \times [0,1] \tag{2.1}$$

where L_x (L_y) is an operator with derivatives with respect only to x (y) and coefficients that depend only on x (y). A usual procedure for solving the PDE (2.1) equipped with some boundary conditions is to place mesh points in Ω and to replace the derivatives with divided differences. We assume that the mesh points are the intersections of $M + 1$ mesh lines parallel to the y–axis with $N + 1$ lines parallel to the x–axis. Denote by u_{mn} the value of the solution of the discrete problem on the mesh point (x_m, y_n). These values can be taken to be components of a vector u. The result of such a procedure is a system of linear algebraic equations of the form

$$M\mathrm{u} = \mathrm{g} \tag{2.2}$$

where M is a matrix and g a vector with components formed from values of the right–hand side of (2.1) and the boundary values. The exact form

of M depends on the operator $L_x + L_y$, the difference approximation and the type of the boundary conditions. For the 5-point star difference approximation of Poisson's equation it is easy to see that the associated linear algebraic system (2.2) takes the form

$$[I \otimes A + B \otimes I]\mathrm{u} = \mathrm{g} \tag{2.3}$$

where A, B are tridiagonal matrices. It is obvious that $I \otimes A$ and $B \otimes I$ commute and it is well known [32] that commutativity plays a significant role in the analysis of ADI methods. It is also known [32], [33] that for commutativity, we do not need to have the two matrices in (2.3) in this specific form.

In order to comment on the analysis presented in the sequel we list below a definition of the standard tridiagonal form and an equivalence theorem found in [26].

Definition 2.1 *We say that the matrices H and V are of standard tridiagonal form if H is block diagonal and each of the non-zero blocks is a tridiagonal matrix and V is a block tridiagonal matrix where each non-zero block is a diagonal matrix.*

Theorem 2.1 *If the $MN \times MN$ matrices H and V are of standard tridiagonal form, then a necessary and sufficient condition for the relation $HV = VH$ is that there exist a nonsingular diagonal $MN \times MN$ matrix D and tridiagonal $M \times M$ and $N \times N$ matrices E and F such that $H = D^{-1}(I \otimes E)D$ and $V = D^{-1}(F \otimes I)D$.*

We assume from now on, that the matrices A and B are non–singular and have linearly independent eigenvectors. So we have that there exist matrices P and Q such that

$$Q^{-1}AQ = \Lambda(A) \quad \text{and} \quad P^{-1}BP = \Lambda(B)$$

where $\Lambda(X)$ is a diagonal matrix with the eigenvalues of X on its diagonal.

It then follows that

$$P^{-1} \otimes Q^{-1}(I \otimes A + B \otimes I)P \otimes Q = I \otimes \Lambda(A) + \Lambda(B) \otimes I$$

and hence

$$(I \otimes A + B \otimes I)^{-1} = P \otimes Q(I \otimes \Lambda(A) + \Lambda(B) \otimes I)^{-1} P^{-1} \otimes Q^{-1}.$$

The matrix $I \otimes \Lambda(A) + \Lambda(B) \otimes I$ is diagonal and its inverse is obtained trivially. From the above discussion we easily see that the solution of (2.3) is given by:

$$\mathrm{u} = \left[P \otimes Q \left(I \otimes \Lambda(A) + \Lambda(B) \otimes I \right)^{-1} P^{-1} \otimes Q^{-1} \right] \mathrm{g}. \tag{2.4}$$

This relation gives the exact solution of the discrete PDE in terms of tensor products of quantities associated with ordinary difference equations. That is to say that the $NM \times NM$ matrix $I \otimes A + B \otimes I$ is directly inverted in terms of the eigenvectors and eigenvalues of A and B which are only $N \times N$ and $M \times M$ matrices. The formula (2.4) is of familiar form and indicates that we have merely obtained the discrete Green function for the partial difference equation. For our case, that we have the Poisson equation, where the Green function is known, the similarity is clear. In fact the matrix operator of (2.4) tends to the Green function operator of the continue problem as the space discretization tends to zero.

There is another interesting interpretation of (2.4). We consider the matrix $[P \otimes Q(I \otimes \Lambda(A) + \Lambda(B) \otimes I)^{-1} P^{-1} \otimes Q^{-1}]$ as a sequence of three operators. Each of these is represented by a matrix and has a separate interpretation. The first matrix $P^{-1} \otimes Q^{-1}$ projects the function g onto a set of orthonormal coordinates. Thus it determines the components of g in terms of a system of unit vectors naturally associated with the problem, namely the eigenvectors of $I \otimes A + B \otimes I$. The second matrix $[I \otimes \Lambda(A) + \Lambda(B) \otimes I]^{-1}$ represents the effect of the operator $(I \otimes A + B \otimes I)^{-1}$ acting on these unit vectors. The third matrix $P \otimes Q$ is the inverse of the first matrix and transforms the results back into a function in the coordinate system of g, and this is the solution. The above discussion can be graphically illustrated as follows:

$$\begin{array}{lll} \text{data} & \mathrm{g} & \mathrm{u} \ \text{solution} \\ P^{-1} \otimes Q^{-1} & \downarrow & \uparrow \ P \otimes Q \\ & H & \longrightarrow \quad H \\ & & [I \otimes \Lambda(A) + \Lambda(B) \otimes I]^{-1} \end{array}$$

where H is the vector space with unit vectors the above eigenvectors. The usefulness of the process is that in H the inverse of the partial difference operator is diagonal and easily computed.

We can also find the solution of the equation (2.3) with the following ADI scheme, for an $\mathrm{u}^{(0)}$ given

$$(L_x + \rho_{s+1} I)\mathrm{u}^{(s+1/2)} = \mathrm{g} - (L_y - \rho_{s+1} I)\mathrm{u}^{(s)}$$

$$(L_y + \rho_{s+1} I)\mathrm{u}^{(s+1)} = (L_y - \omega\rho_{s+1} I)\mathrm{u}^{(s)} + (1+\omega)\rho_{s+1}\mathrm{u}^{(s+1/2)} \qquad (2.5)$$

where $\rho_{s+1}, s = 0, \ldots,$ and ω are appropriate iteration parameters.

If e_i, f_j are orthonormal eigenvectors and λ_i, μ_j are the corresponding eigenvalues of the matrices A and B, respectively, then (see Section 2.1) the vectors $f_j \otimes e_i$ form a complete set of orthonormal eigenvectors for both $I \otimes A$ and $B \otimes I$. Let u denote the solution of (2.3) and define $\mathrm{z}^{(s)} = \mathrm{u}^{(s)} - \mathrm{u}$. The error $\mathrm{z}^{(s)}$ can be now expanded in terms of the eigenvectors as

$$\mathrm{z}^{(s)} = \sum_{i,j} a_{ij}^{(s)} f_i \otimes e_j.$$

Substituting $\mathrm{u}^{(s)} = \mathrm{z}^{(s)} + \mathrm{u}$ into (2.5) we can easily see that

$$a_{ij}^{(s+1)} = a_{ij}^{(s)} \left[1 - \frac{(1+\omega)\rho_{s+1}(\lambda_i + \mu_j)}{(\lambda_i + \rho_{s+1})(\mu_j + \rho_{s+1})} \right].$$

From the above it is easy to see that positive eigenvalues λ_i, μ_j , positive ρ_{s+1} and $0 \le \omega \le 1$ imply that

$$\left| \frac{a_{ij}^{(s+1)}}{a_{ij}^{(s)}} \right| < 1.$$

Therefore the iteration scheme (2.5) is convergent and u is its only solution.

Note that for $\omega = 1$ (the known as Peaceman–Rachford scheme) a_{ij} can be made zero for all i by choosing $\rho_{s+1} = \mu_j$, the eigenvalues of matrix B, for some j. This shows the power of this scheme: a large number of components of the error can be annihilated simultaneously. Furthermore this is accomplished without increasing any other components of the error vector if $\lambda_i, \mu_j, \rho_{s+1}$ are positive, as is usually the case.

In contrast the SOR method can annihilate at most one component of the error, and Douglas–Rachford scheme with $\omega = 0$ cannot annihilate any error component. Therefore the Peaceman–Rachford scheme can be exact (except of round-off) in a number of iterations equal to the number of grid points taken in either the x or y direction. Nevertheless it is usually more efficient to use "optimum parameters" and this is because we can obtain satisfactory approximate solutions with a smaller total number of iterations than the $\min(M, N)$.

Tensor products can be similarly applied to other PDE problems and other discretization schemes. The following are examples of the tensor product formulation of common PDE problems defined on a rectangular domain:

- 9–point approximation of Poisson equation

$$[6I \otimes A + 6B \otimes I + B \otimes A]\mathrm{u} = \mathrm{g}$$

- 5–point approximation of Helmholtz equation

$$[I \otimes A + B \otimes I + \sigma I \otimes I]\mathrm{u} = \mathrm{g},\ \sigma\ constant$$

- 13–point approximation of the biharmonic equation

$$[I \otimes A + B \otimes I]^2\mathrm{u} = \mathrm{g}$$

- The Poisson equation in spherical coordinates in 3 dimensions

$$[(C \otimes B_1 + I \otimes B_2) \otimes I + I \otimes I \otimes A]\mathrm{u} = \mathrm{g}.$$

The tensor product formulation can also be used to solve parabolic finite difference equations. A typical scheme is the implicit Crank-Nicolson method for the heat equation which results in a partial difference equation of the form

$$[I \otimes A + B \otimes I][\mathrm{u}(t + \Delta t) + \mathrm{u}(t)] = 2\sigma[\mathrm{u}(t + \Delta t) - \mathrm{u}(t)]$$

where σ depends on the grid spacing and the time stepping. Finally we point out that tensor product analysis can also be applied to other problems such as difference-differential equations, integral equations in more than one variable, integro-differential equations, etc.

2.3 Efficient manipulation and complexity

The fact that a particular matrix factors into the tensor product of two or more matrices is of no computational value unless we have algorithms for the efficient manipulation of tensor products. For example, to compute $(\prod_{i=1}^{k} \otimes A_i)$u we desire a procedure which uses only the factors A_i and which avoids explicitly forming the matrix $A = \prod_{i=1}^{k} \otimes A_i$. Such procedures are given in [6].

Before we discuss the techniques used for computer manipulations of tensor products, we must define the data structures used to represent them. We naturally represent the individual matrix factors of a tensor product as separate matrices stored in Fortran fashion as two-dimensional arrays. However when working with tensor products it is computationally convenient to represent the vectors as k-dimensional arrays. So we have the following definition.

Definition 2.2 *Let* x *be a vector of order* $\prod_{i=1}^{k} N_i$ *where* $N_i \in \mathbb{N}, i = 1, \ldots, k$. *We define the* $N_1 N_2 \ldots N_k$ *k–multidimensional matrix representation* $\mathrm{X} = \{\mathrm{X}_{i_1,i_2,\ldots,i_k}\}$ *of* x *by*

$$\mathrm{X}_{i_1,i_2,\ldots,i_k} = \mathrm{x}_{i_1+N_1(i_2-1+N_2(i_3-1+\ldots+N_{k-1}(i_k-1)\ldots))}$$

where $i_j = 1, \ldots, N_j, j = 1, \ldots, k$.

We also need the following definition for the transpose of a k-multidimensional matrix.

Definition 2.3 *Let* X *be a k-multidimensional matrix of order* $\prod_{i=1}^{k} N_i$. *Then* X^T *is a k-multidimensional matrix of order* $\prod_{i=2}^{k} N_i \times N_1$ *such that*

$$\mathrm{X}^T_{l_2,l_3,\ldots,l_k,l_1} = \mathrm{X}_{l_1,l_2,\ldots,l_k}$$

where $l_i = 1, \ldots, N_i, i = 1, \ldots, k$.

As easily seen from the ADI equations (2.5), the two basic computational blocks needed for the implementation of the ADI schemes are

matrix vector product and matrix inversion (or better solution of linear system).

Thus, we need, an efficient procedure for computing matrix times vector operations of the tensor product form $(\prod_{i=1}^{k} \otimes A_i)$x and the ability to solve efficiently systems of linear equations given in following tensor product form

$$\left(\prod_{i=k}^{1} \otimes A_i\right) \mathrm{X} = \mathrm{B}. \tag{2.6}$$

Using the above definitions the following Lemma and Theorem (their proofs are simple generalizations of the proofs found in [6] for the $k = 2$ case) give us such efficient procedures which only involve the matrices A_i, X and B.

Lemma 2.2 *Let A_i be matrices of order $M_i \times N_i, i = 1, \ldots, k$,* x *be a vector of order $\prod_{i=1}^{k} N_i$ and* X *be its k-multidimensional matrix representation of order $N_1 \ldots N_{k-1} N_k$. The $M_1 \ldots M_{k-1} M_k$ matrix $\left(\prod_{i=k}^{1} \otimes A_i\right)$* x *is given by*

$$\left(\prod_{i=k}^{1} \otimes A_i\right) \mathrm{x} = \left(A_k \left(A_{k-1} \left(\cdots A_2 \left(A_1 \mathrm{X}\right)^T \ldots\right)^T\right)^T\right)^T. \tag{2.7}$$

In the sequel we will use the k-multidimensional representation of each vector. Using the previous Lemma we have the following Theorem, which gives us an efficient way for solving linear systems of equations of the form (2.6).

Theorem 2.2 *Let A_i be matrices of order $N_i \times N_i, i = 1, \ldots, k$. Let* X *and* B *be k-multidimensional matrices of order $N_1 N_2 \ldots N_k$ and consider the linear system $\left(\prod_{i=k}^{1} \otimes A_i\right)$ X = B. If A_i^{-1} exists for $i = 1, \ldots, k$ and if*

$$A_1 \mathrm{Y}_1 = \mathrm{B},$$

$$A_2 \mathrm{Y}_2 = \mathrm{Y}_1^T,$$

$$\vdots$$

$$A_i \mathrm{Y}_i = \mathrm{Y}_{i-1}^T,$$

$$\vdots$$

$$A_k \mathrm{Y}_k = \mathrm{Y}_{k-1}^T,$$

then $\mathrm{X} = \mathrm{Y}_k^T$.

From the same Lemma we also obtain the following Corollary.

Corollary 2.1 *Let* $A_i, i = 1, \ldots, k$, *and* X *be matrices as in previous Theorem. Then*

$$\left(\left(\prod_{i=k}^{1} \otimes A_i \right) \mathrm{X} \right)^T = \left(A_1 \otimes \left(\prod_{i=k}^{2} \otimes A_i \right) \right) \mathrm{X}^T. \tag{2.8}$$

It is important to point out that this simple observation gives us the ability to switch the order of the terms in a tensor product. This allows us instead of solving linear systems of the form

$$\left(\left(\prod_{i=k}^{m+1} \otimes I \right) \otimes A_m \otimes \left(\prod_{i=m-1}^{1} \otimes I \right) \right) \mathrm{X} = \mathrm{B} \ , \tag{2.9}$$

to use existing numerical software to compute the solution of the equivalent, in view of Corollary 2.1, linear system

$$\left(\left(\prod_{i=m-1}^{1} \otimes I \right) \otimes \left(\prod_{i=k}^{m+1} \otimes I \right) \otimes A_m \right) \mathrm{X}^{\overbrace{T \ldots T}^{m-1 \quad times}} = \mathrm{B}^{\overbrace{T \ldots T}^{m-1 \quad times}} \ . \tag{2.10}$$

In order to compare the efficiency of numerical algorithms it is customary to measure their computational complexity by estimating the computer work in terms of basic operations. We do this by computing the so–called operation counts in the traditional manner by counting only floating–point operations.

We give below a series of Lemmata which will help us to estimate, in Chapter 4, the computer work required for our ADI schemes.

Lemma 2.3 *Let* $A_i, i = 1, \ldots, k$, *be matrices of order* $N_i \times N_i$ *and let* X *be a* k *multi-dimensional matrix of order* $N_1 N_2 \ldots N_k$. *Then the computer work required to compute* $(A_k \otimes A_{k-1} \otimes \ldots \otimes A_1)\mathrm{X}$ *is* $\mathcal{K} \sum_{i=1}^{k} (2N_i - 1)$. *If* A_i *are band matrices with respective bandwidths* M_i *then this work reduces to* $4\mathcal{K} \sum_{i=1}^{k} M_i$.

Table 2.1: Number of operations to compute $(A_k \otimes A_{k-1} \otimes \ldots \otimes A_1)\mathrm{X}$.

Procedure	Work			
	Full		Band	
	+,−	*	+,−	*
$A_1\mathrm{X}$	$(N_1-1)\mathcal{K}$	$N_1\mathcal{K}$	$2M_1\mathcal{K}$	$2M_1\mathcal{K}$
$A_2(A_1\mathrm{X})^T$	$(N_2-1)\mathcal{K}$	$N_2\mathcal{K}$	$2M_2\mathcal{K}$	$2M_2\mathcal{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_i(A_{i-1}(\ldots(A_1\mathrm{X})^T\ldots)^T)^T$	$(N_i-1)\mathcal{K}$	$N_i\mathcal{K}$	$2M_i\mathcal{K}$	$2M_i\mathcal{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_k(\ldots(A_1\mathrm{X})^T\ldots)^T$	$(N_k-1)\mathcal{K}$	$N_k\mathcal{K}$	$2M_k\mathcal{K}$	$2M_k\mathcal{K}$
Total	$\mathcal{K}\sum_{i=1}^{k}(N_i-1)$	$\mathcal{K}\sum_{i=1}^{k}N_i$	$2\mathcal{K}\sum_{i=1}^{k}M_i$	$2\mathcal{K}\sum_{i=1}^{k}M_i$

Proof: Since $A_1 \in \mathbb{R}^{N_1 \times N_1}$ and $\mathrm{X} \in \mathbb{R}^{N_1 \times (N_2 N_3 \ldots N_k)}$ it is easy to see that the number of additions/subtractions required to calculate $A_1\mathrm{X}$ is $(N_1-1)\mathcal{K}$ and the number of multiplications is $N_1\mathcal{K}$ if A_1 is full. In the case that A_1 is band with bandwidth M_1 the required work reduces to $2M_1\mathcal{K}$ for both the additions/subtractions and the multiplications. Using Definition 2.3 we continue in the same way for the rest of the matrices and finally we add to get the total operation count. The above are briefly presented in Table 2.1. □

Lemma 2.4 *Let $A_i, i = 1, \ldots, k$, be matrices of order $N_i \times N_i$ and let* X, B *be k multi-dimensional matrices of order $N_1 N_2 \ldots N_k$. Then the computer work required to solve the linear system (2.6) using Gauss elimination with partial pivoting is $2(\mathcal{K}\sum_{i=1}^{k} N_i + \sum_{i=1}^{k} N_i^3/3)$. If A_i are band matrices with respective bandwidths M_i then the computer work required is $2(3\mathcal{K}\sum_{i=1}^{k} M_i + \sum_{i=1}^{k} M_i^2 N_i)$.*

Proof: Recall that the number of additions/subtractions or multiplications for factorizing the matrix $A \in \mathbb{R}^{N \times N}$ is $O(N^3/3)$ or $O(M^2 N)$ if A is band with bandwidth M. While the cost to forward/backward solve $LU\mathrm{Y} = \mathrm{B}$ where $A = LU$, $\mathrm{Y}, \mathrm{B} \in \mathbb{R}^{N \times (\mathcal{K}/N)}$ is $N_i\mathcal{K}$ or $3M\mathcal{K}$ if A is banded with bandwidth M, we now easily understand the entries of Table 2.2. □

Table 2.2: Number of operations to solve $\left(\prod_{i=k}^{1} \otimes A_i\right) \mathrm{X} = \mathrm{B}$ using Gauss elimination.

Procedure	Work			
	Full		Band	
	$+,-$	$*$	$+,-$	$*$
Factor $A_1 = L_1U_1$	$N_1^3/3$	$N_1^3/3$	$M_1^2N_1$	$M_1^2N_1$
Solve $L_1U_1\mathrm{Y}_1 = \mathrm{B}$	$N_1\mathcal{K}$	$N_1\mathcal{K}$	$3M_1\mathcal{K}$	$3M_1\mathcal{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Factor $A_i = L_iU_i$	$N_i^3/3$	$N_i^3/3$	$M_i^2N_i$	$M_i^2N_i$
Solve $L_iU_i\mathrm{Y}_i = \mathrm{Y}_{i-1}^T$	$N_i\mathcal{K}$	$N_i\mathcal{K}$	$3M_i\mathcal{K}$	$3M_i\mathcal{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Factor $A_k = L_kU_k$	$N_k^3/3$	$N_k^3/3$	$M_k^2N_k$	$M_k^2N_k$
Solve $L_kU_k\mathrm{Y}_k = \mathrm{Y}_{k-1}^T$	$N_k\mathcal{K}$	$N_k\mathcal{K}$	$3M_k\mathcal{K}$	$3M_k\mathcal{K}$
Total	$\mathcal{K}\sum_{i=1}^{k} N_i +$ $\sum_{i=1}^{k} N_i^3/3$	$\mathcal{K}\sum_{i=1}^{k} N_i +$ $\sum_{i=1}^{k} N_i^3/3$	$3\mathcal{K}\sum_{i=1}^{k} M_i +$ $\sum_{i=1}^{k} M_i^2N_i$	$3\mathcal{K}\sum_{i=1}^{k} M_i +$ $\sum_{i=1}^{k} M_i^2N_i$

Table 2.3: Number of operations to solve $\left(\prod_{i=k}^{1} \otimes A_i\right) \mathrm{X} = \mathrm{B}$.

Procedure	Work			
	Full		Band	
	$+,-$	$*$	$+,-$	$*$
$(A_k \otimes \ldots \otimes A_1)B$: Expanding $\prod_{i=k}^{1} \otimes A_i$ Using Lemma 2.2	$N^{2k} - N^k$ $k(N^{k+1} - N^k)$	N^{2k} kN^{k+1}	$2MN^k$ $2kMN^k$	$2MN^k$ $2kMN^k$
Solve $(\prod_{i=k}^{1} \otimes A_i)X = B$: Expanding $\prod_{i=k}^{1} \otimes A_i$ Using Lemma 2.3	$N^{3k}/3 + N^{k+1}$ $k(N^{k+1} + N^3/3)$	$N^{3k}/3 + N^{k+1}$ $k(N^{k+1} + N^3/3)$	$(M^2 + 3M)N^k$ $k(3MN^k + M^2N)$	$(M^2 + 3M)N^k$ $k(3MN^k + M^2N)$

For matrices $A_i, \mathrm{X}, \mathrm{B}$ given as above, where now $N_i = N$ and $M_i = M$ for $i = 1, \ldots, k$, the work required for the above considered procedures but with the tensor product $\prod_{i=k}^{1} \otimes A_i$ expanded is compared with the results of the above given Lemmata in Table 2.3 where the significant difference in the efficiency is clearly observed.

The procedures described above are not only time efficient but also memory efficient. Using simple calculations we can obtain Table 2.4 which shows the amount of memory required to store the data by using the data structures described at the beginning of this section (second row) and by storing $\prod_{i=k}^{1} \otimes A_i$ expanded (third row). As we see, the

Table 2.4: Memory requirements for storing $\prod_{i=1}^{k} \otimes A_i$.

Storage format	Memory	
	Full	Band
$(A_k \otimes \cdots \otimes A_1)$	$\sum_{i=1}^{k} N_i^2$	$2\sum_{i=1}^{k} K_i N_i$
expand $\prod_{i=k}^{1} \otimes A_i$	$\prod_{i=1}^{k} N_i^2$	$2K_1 N_1 \prod_{i=2}^{k} N_i^2$

usage of tensor products can significantly reduce the computer memory required to store $\prod_{i=k}^{1} \otimes A_i$. This is particularly important for solving three (or more) dimensional problems for which memory can easily be exhausted even on modern computers.

Finally it is worth to point out that the ADI methods considered in this paper have, due to the uncoupling of the directions achieved, an increased degree of inherent parallelism which has been exploited lately both experimentally [28], [3] and theoretically [23]. Several parallel implementations of ADI methods have been proposed recently. Specifically in [23] a complexity analysis of such implementations on distributed memory parallel machines with ring, 2-dimensional grid and hypercube connectivity shows that the ADI method can be made highly efficient achieving almost linear speedup. A parallel implementation of ADI methods on shared memory machines is presented in [3]. Experimental results obtained on a Sequent Balance machine exhibit the increased parallelization of the implementation.

Chapter 3

Cubic Spline Collocation Methods

The main objective of this Chapter is to derive the linear system of algebraic equations associated with the Cubic Spline Collocation (CSC) discretization method which we briefly describe here. We also introduce some notation to be used later. For a detailed formulation and analysis of the Cubic Spline Collocation method in two dimensions the reader is referred to [20].

We start by discretizing uniformly, with step size h_i, each interval $[a_i, b_i], i = 1, \ldots, k$. We also introduce two extra points beyond the ends of the intervals to get

$$\Delta_i \equiv \left\{ \tau_\ell^i = a_i + \ell h_i \ ; \ \ell = -1, \ldots, N_i + 1 \text{ with } h_i = \frac{b_i - a_i}{N_i} \right\}$$

a uniform partition of $[a_i, b_i]$. Then $\Delta \equiv \prod_{i=1}^{k} \otimes \Delta_i$ is the induced uniform partition of the domain Ω. We denote by $S_{3,\Delta_i} \equiv \mathrm{P}_{3,\Delta_i} \cap C^2([a_i, b_i])$ the space of the one–dimensional splines defined by the partition Δ_i of $[a_i, b_i]$. The basis elements of the space of the k–dimensional splines $S_{3,\Delta}$ are obtained by taking the tensor product of the basis of the one–dimensional splines S_{3,Δ_i} which can be chosen so that

$$B_\ell^i(\tau_{\ell\pm1}^i) = \frac{1}{6}, B_\ell^i(\tau_\ell^i) = \frac{2}{3}, \left.\frac{d^2 B_\ell^i(\tau)}{d\tau^2}\right|_{\tau_{\ell\pm1}^i} = \frac{1}{h_i^2}, \left.\frac{d^2 B_\ell^i(\tau)}{d\tau^2}\right|_{\tau_\ell^i} = \frac{-2}{h_i^2}, \quad (3.1)$$

while B_ℓ^i and $\frac{d^2 B_\ell^i(\tau)}{d\tau^2}$ vanish at all other points in Δ_i.

The cubic spline Collocation approximate $u_\Delta \in S_{3,\Delta}$ can then be represented as:

$$u_\Delta(\mathrm{x}) = \sum_{\ell_1=-1}^{N_1+1} \sum_{\ell_2=-1}^{N_2+1} \dots \sum_{\ell_k=-1}^{N_k+1} U_{\ell_1 \ell_2 \dots \ell_k} B_{\ell_1}^1(x_1) B_{\ell_2}^2(x_2) \dots B_{\ell_k}^k(x_k) \quad (3.2)$$

where $\mathrm{x} = (x_1, x_2, \dots, x_k)$ is a point in Δ and where $U_{\ell_1 \ell_2 \dots \ell_k}$ (with $\ell_i = -1, \dots, N_i + 1, i = 1, \dots, k$) are the unknown spline Collocation coefficients. In order to determine these $\prod_{i=1}^k (N_i + 3)$ unknowns we require u_Δ to satisfy the PDE (1) at all points in Δ and the boundary conditions (2) at all boundary points $\Delta \cap \partial\Omega$. Using well known results from the spline-interpolation theory ([25]) it can be easily seen that the solution u of the PDE problem satisfies the Collocation equations within an error of order $O(h^2)$. In order to obtain an optimal ($O(h^4)$) spline approximation u_Δ of u we force it to satisfy the perturbed PDE $L'u = f$ where the operator L' is a perturbation of the operator L in (1) and can be obtained by replacing, for $i = 1, \dots, k$,

$$\left. \frac{\partial^2 u}{\partial x_i^2} \right|_{\mathrm{x}}$$

by

$$\frac{1}{12} \left[\left. \frac{\partial^2 u}{\partial x_i^2} \right|_{\mathrm{x}^-} + 10 \left. \frac{\partial^2 u}{\partial x_i^2} \right|_{\mathrm{x}} + \left. \frac{\partial^2 u}{\partial x_i^2} \right|_{\mathrm{x}^+} \right]$$

where x^-, x and x^+ are grid points in Δ consecutive in the i–direction.

It has been observed ([20]) that, in both $O(h^2)$ and $O(h^4)$ Collocation schemes for the PDE problem defined by (1a), the Collocation equations obtained from the boundary conditions and from the differential equation at the end points of each line can be uncoupled. For the 3–dimensional case we can achieve such an uncoupling in the following way. We denote by $U_x^{(0)}, U_x^{(1)}$ and $U_x^{(-1)}$ the vectors of order $(N_2+3)(N_3+3)$ which contain the unknowns at the levels $x_1 = \tau_0^1, x_1 = \tau_1^1$ and $x_1 = \tau_{-1}^1$, respectively. We also denote by $F_x^{(0)}$ the vector of the same order which contains the values of the right–hand side function f at $x_1 = \tau_0^1$. Since $u = 0$ at $x_1 = \tau_0^1$, we have that

$$\left. \frac{\partial^2 u}{\partial x_2^2} \right|_{\mathrm{x}=(\tau_0^1, x_2, x_3)} = \left. \frac{\partial^2 u}{\partial x_3^2} \right|_{\mathrm{x}=(\tau_0^1, x_2, x_3)} = 0$$

and therefore the PDE (1) becomes

$$a_1 \frac{\partial^2 u}{\partial x_1^2}(\mathrm{x}) = f(\mathrm{x}) \quad , \quad \mathrm{x} = (\tau_0^1, x_2, x_3)$$

where without loss of generality we assume that a_1 is constant. Requiring the spline Collocation approximation u_Δ (3.2) to satisfy the above two differential equations we easily obtain, using (3.1), the following systems of algebraic equations at level $x_1 = \tau_0^1$:

$$\frac{a_1}{h^2}\left[(T_4 \otimes T_4)\, U_x^{(-1)} - 2\,(T_4 \otimes T_4)\, U_x^{(0)} + (T_4 \otimes T_4)\, U_x^{(1)}\right] = F_x^{(0)},$$

$$\frac{1}{6}\left[(T_4 \otimes T_4)\, U_x^{(-1)} + 4\,(T_4 \otimes T_4)\, U_x^{(0)} + (T_4 \otimes T_4)\, U_x^{(1)}\right] = 0.$$

From these systems we finally obtain the uncoupled linear systems

$$(T_4 \otimes T_4)\, U_x^{(0)} = -\frac{h_1^2}{6a_1} F_x^{(0)}$$

and

$$(T_4 \otimes T_4)\, U_x^{(-1)} = -\,(T_4 \otimes T_4)\, U_x^{(1)} + \frac{2h_1^2}{3a_1} F_x^{(0)}.$$

Working in the same way we obtain analogous equations for the other 5 levels of the boundary to obtain similar to the above given expressions for the rest of the "boundary" unknowns.

The Collocation equations *away* ($2 \le \ell \le N_i - 2$) from the boundary can be presented, for the three-dimensional Poisson equation, in the form of stencils. The stencils associated to the $O(h^2)$ scheme and to the $O(h^4)$ scheme are given in Figures 3.1 and 3.2 respectively. The value of each entry in these stencils is the coefficient of the corresponding unknown. All entries in the $O(h^2)$ stencil have been multiplied by a factor of $-\frac{1}{12h^2}$ and all entries in the $O(h^4)$ one by $-\frac{1}{432h^2}$. For lines next to the boundary, the equations have similar form with appropriately modified right sides (see [16]).

From the above discussion, the assumed representation (3.2) of u_Δ and the nature of the B-spline basis functions (3.1) we easily conclude

(see also [20]) that the *interior* Collocation equations can be written in the form

$$\sum_{i=1}^{k} A_i \mathrm{U} = \mathrm{F}, \quad A_i \in \mathbb{R}^{\mathcal{K}\times\mathcal{K}}, \tag{3.3}$$

where $\mathcal{K} = \prod_{i=1}^{k}(N_i - 1)$ and where (for the Poisson equation)

$$A_i \equiv \left(\prod_{j=k}^{i+1} \otimes T_4^j\right) \otimes \frac{1}{6^{k-1}h_i^2}\mathcal{E}^i \otimes \left(\prod_{j=i-1}^{1} \otimes T_4^j\right) \ , \ i = 1, \ldots, k, \tag{3.4}$$

with

$$\mathcal{E}^i = \begin{cases} T_{-2}^i & \text{second order scheme} \\ \frac{1}{12} T_{10}^i T_{-2}^i & \text{fourth order scheme} \end{cases}$$

where

$$T_\alpha^i = tridiag(1, \alpha, 1), \quad T_\alpha^i \in R^{N_i - 1}.$$

The right–hand side vector F holds the associated values of the right–hand side function f of the PDE and contains the effect of the elimination of the "boundary" unknowns.

Figure 3.1: The coefficient stencil of the $O(h^2)$ Collocation equations at points away from the boundary.

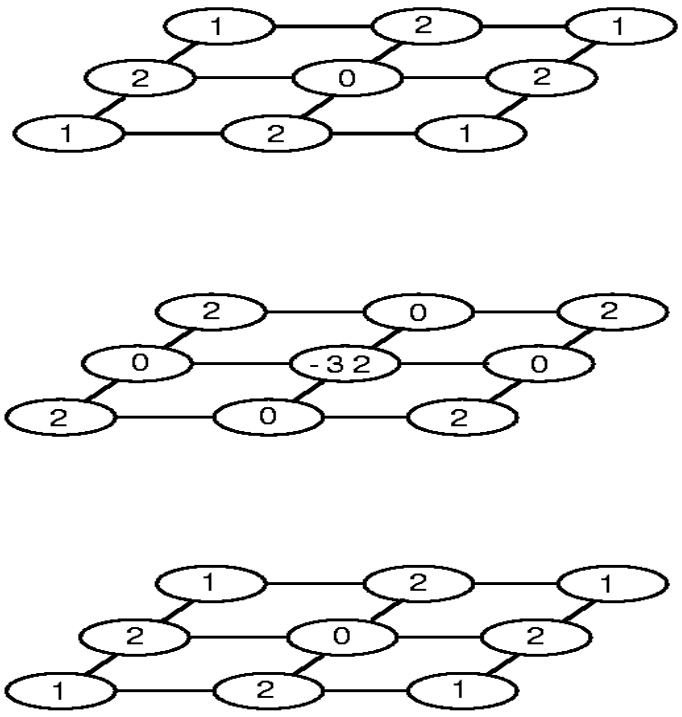
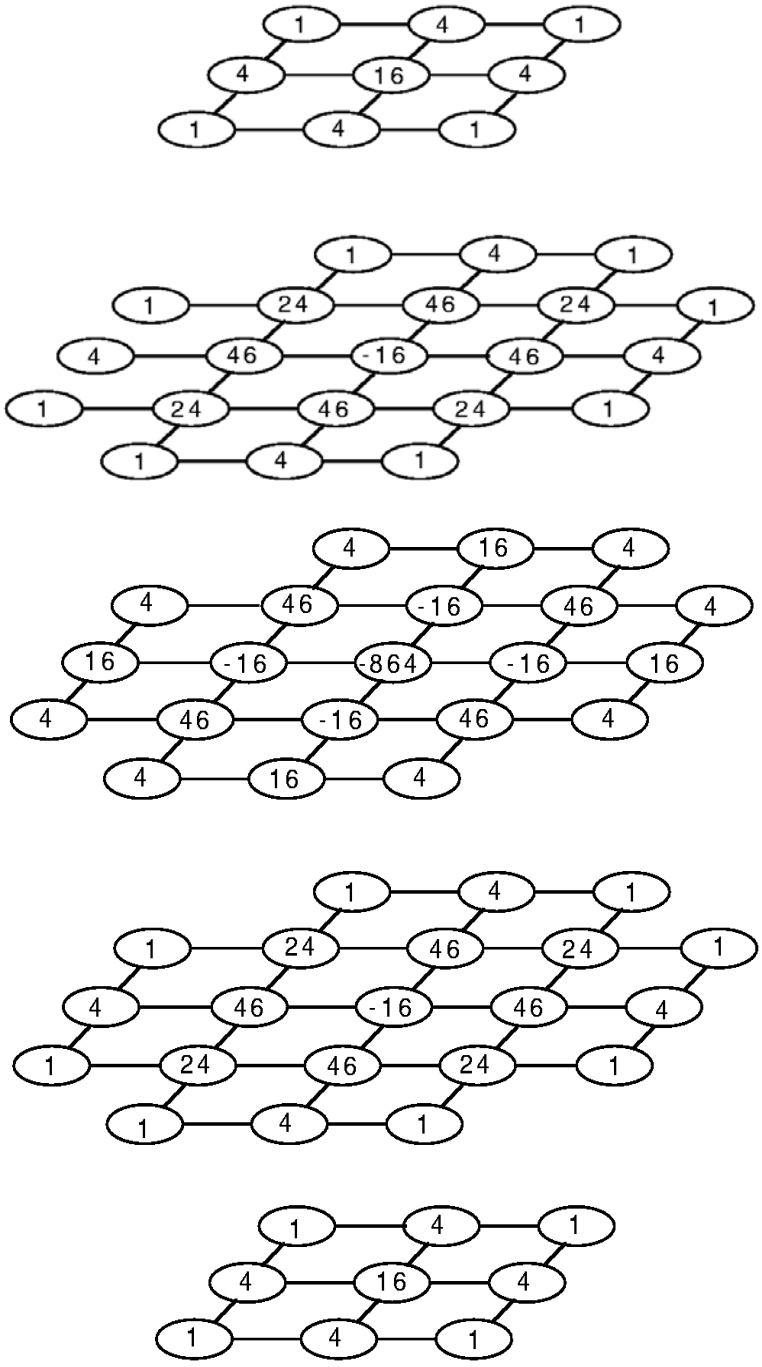


Figure 3.2: The coefficient stencil of the $O(h^4)$ Collocation equations at points away from the boundary.



Chapter 4

ADISC methods

In this Chapter we formulate and analyze ADI methods for solving the cubic spline Collocation equations derived in the previous chapter. A detailed complexity analysis of the ADISC methods is also presented.

4.1 Derivation

The Collocation linear system of coefficients (3.3) can be solved by several direct and iterative solvers (see [20], and [19] for the two dimensional case, and [16] for the k–dimensional one). Our first ADISC method, associated with the linear system (3.3), can be obtained by generalizing the approach found in [9] and can be described by the following recurrence relation:

Given a guess $\mathrm{U}^{(0)}$, iterate for $s = 0, 1, \ldots,$

$$(A_1 + r_{s+1} D)\, \mathrm{U}^{(s+1/k)} = \left[(A_1 + r_{s+1} D) - \omega \sum_{i=1}^{k} A_i \right] \mathrm{U}^{(s)} + \omega \mathrm{F} \qquad (4.1)$$

$$(A_j + r_{s+1} D)\, \mathrm{U}^{(s+j/k)} = A_j \mathrm{U}^{(s+(j-1)/k)} + r_{s+1} D \mathrm{U}^{(s)},\ j = 2, \ldots, k \qquad (4.2)$$

where $D \equiv \frac{1}{6^k} \prod_{j=k}^{1} \otimes T_4^j$, $r_{s+1}, s = 0, 1, \ldots$, are positive parameters to accelerate the convergence and ω is a relaxation parameter. It should be pointed out that for $\omega = 1$ our scheme reduces to the so called Douglas-Rachford scheme ([10]) and that for $\omega = 2$ it reduces to the Douglas scheme ([9]).

The above ADISC scheme can be written in the following matrix form

$$\mathrm{U}^{(s+1)} = T_{r_{s+1},\omega}\mathrm{U}^{(s)} + r_{s+1}^{k-1}\omega \prod_{i=k}^{1}\left(\frac{1}{r_{s+1}}D^{-1}A_i + I\right)^{-1}\mathrm{F} \tag{4.3}$$

where the iteration matrix $T_{r_{s+1},\omega}$ is given by

$$T_{r_{s+1},\omega} = I - \omega\frac{1}{r_{s+1}}\prod_{i=1}^{k}\left(\frac{1}{r_{s+1}}D^{-1}A_i + I\right)^{-1}\left(\sum_{i=1}^{k}D^{-1}A_i\right). \tag{4.4}$$

In an effort to increase the per iteration efficiency of the above described ADISC scheme we express all the tridiagonal matrices T_{α}^{i} involved, in terms of the associated identity matrix I and the tridiagonal matrix T_{-2}^{i}. Substituting these expressions in (3.3), the $O(h^2)$ interior Collocation equations are now given (for the Poisson equation) in the form

$$\left[\sum_{i=1}^{k}X_i + \sum_{i=2}^{k}\sum_{j=1}^{\binom{k}{i}}H_{ij}\right]\mathrm{U} = \mathrm{F} \tag{4.5}$$

with

$$X_i = -\frac{1}{h^2}\left(\prod_{j=k}^{i+1}\otimes I\right)\otimes T_{-2}^{i}\otimes\left(\prod_{j=i-1}^{1}\otimes I\right)$$

and

$$H_{ij} = -\frac{i}{6^{i-1}h^2}C_{ij}$$

where C_{ij} is a tensor product of k matrices where i of them are T_{-2}^{i} and the rest $k-i$ are identity matrices.

The above form of the linear system naturally leads us to our second ADI scheme given by the following recurrence relation:

$$\left(r_{s+1}I + X_1\right)\mathrm{U}^{(s+1/k)} = \tag{4.6}$$
$$\left(r_{s+1}I + X_1\right)\mathrm{U}^{(s)} - \omega\left(\sum_{i=1}^{k}X_i + \sum_{i=2}^{k}\sum_{j=1}^{\binom{k}{i}}H_{ij}\right)\mathrm{U}^{(s)} + \omega\mathrm{F}$$

$$(r_{s+1}I + X_i)\,\mathrm{U}^{(s+i/k)} = r_{s+1}\mathrm{U}^{(s+(i-1)/k)} + X_i\mathrm{U}^{(s)}\ ,\ i = 2,\ldots,k \quad (4.7)$$

which in matrix form is given by

$$\mathrm{U}^{(s+1)} = S_{r_{s+1},\omega}\mathrm{U}^{(s)} + r_{s+1}^{k-1}\omega\left(\prod_{i=k}^{1}(r_{s+1}I + X_i)^{-1}\right)\mathrm{F} \quad (4.8)$$

and where the associated iteration matrix is

$$S_{r_{s+1},\omega} = \quad (4.9)$$
$$I - \omega\left(\prod_{i=k}^{1}\left(I + \frac{1}{r_{s+1}}X_i\right)^{-1}\right)\left(\sum_{i=1}^{k}\frac{1}{r_{s+1}}X_i + \sum_{i=2}^{k}\sum_{j=1}^{\binom{k}{i}}\frac{1}{r_{s+1}}H_{ij}\right).$$

As we easily see the new scheme is closer to the ADI scheme (2.5) associated with the 5 point star discretization method as far as matrix inversions are concerned. It has been observed ([27]) that ADI formulations like (4.6) - (4.7) may lead to an increased accuracy. The analysis of this phenomenon is beyond the scope of this study and will be presented elsewhere. Therefore, in this study we will only consider the $O(h^2)$ version of (4.6) - (4.7).

In the sequel we will call the ADI method defined by the relations (4.1) -(4.2) ADISC1 and the one defined by the relations (4.6) - (4.7) ADISC2.

4.2 Complexity analysis

Based on the complexity results obtained in Chapter 2 and specifically Lemmata 2.3 and 2.4, it is easy to calculate the, per iteration, work involved in the two $O(h^2)$ ADISC schemes derived above. This way, a detailed arithmetic count for the sweep of our schemes in the first direction (relations (4.1) and (4.6)) is presented in Tables 4.1 and 4.3, respectively, while the arithmetic count in the other directions (associated with relations (4.2) and (4.7)) are presented in Tables 4.2 and 4.4. Multiplying the last rows in Tables 4.2 and 4.4 by $k-1$ and adding them to the last rows in Tables 4.1 and 4.3 respectively we easily see

Table 4.1: Work to sweep the first direction of ADISC1.

Procedure	Work	
	$+,-$	$*$
$W_1 := D + r_{s+1}A_1$	$3(N_1-1)$	$3(N_1-1)$
$U_i^{(s)} := A_i U^{(s)}$	$2k\mathcal{K}$	$2k\mathcal{K}$
$Y := \sum_{i=1}^{k} U_i^{(s)}$	$(2k^2+k-1)\mathcal{K}$	$2k^2\mathcal{K}$
R $:= W_1 U^{(s)} - \omega Y + \omega F$	$2(k+1)\mathcal{K}$	$2(k+1)\mathcal{K}$
Solve $W_1 U^{(s+1/k)} =$ R	$3k\mathcal{K} + \sum_{i=1}^{k} N_i$	$3k\mathcal{K} + \sum_{i=1}^{k} N_i$
Total	$(2k^2+6k+1)\mathcal{K}+$ $3(N_1-1)+\sum_{i=1}^{k} N_i$	$(2k^2+5k+2)\mathcal{K}+$ $3(N_1-1)+\sum_{i=1}^{k} N_i$

that the total number of additions (O_1^A) and multiplications (O_1^M) is given for our ADISC1 scheme by

$$O_1^A = 9k^2\mathcal{K} + (3+k)\sum_{i=1}^{k} N_i - 3k \quad , \tag{4.10}$$

and

$$O_1^M = (9k^2-k-1)\mathcal{K} + (3+k)\sum_{i=1}^{k} N_i - 3k \quad . \tag{4.11}$$

Similarly for the ADISC2 scheme we have that

$$O_2^A = (2^k(k+1)+7k)\mathcal{K} + 4\sum_{i=1}^{k} N_i - 3k \quad ,$$

and

$$O_2^M = (2^k+8k+1)\mathcal{K} + 4\sum_{i=1}^{k} N_i - 3k \quad .$$

To compare the per iteration complexity of our two schemes we need to compare the overall total work required. Assuming that the time to perform (in the CPU) a multiplication is twice the time to perform an addition, we calculate the quantities $O_i = O_i^A + 2O_i^M$, $i = 1, 2$, which correspond to the CPU time needed for the ADISC1 and ADISC2 schemes to perform an iteration, respectively. In Figure 4.1 we plot $100\frac{O_1-O_2}{O_2}$ versus the dimension k of the PDE domain Ω. As depicted

Table 4.2: Work to sweep the last $k-1$ directions of ADISC1.

Procedure	Work	
	+,-	*
$W_i := D + r_{s+1}A_i$	$3(N_i - 1)$	$3(N_i - 1)$
$Y_1 := DU^{(s+(i-1)/k)}$	$2k\mathcal{K}$	$2k\mathcal{K}$
$Y_2 := A_iU^{(s)}$	$2k\mathcal{K}$	$2k\mathcal{K}$
$Y := Y_1 + r_{s+1}Y_2$	$\mathcal{K}$	$\mathcal{K}$
Solve $W_iU^{(s+i/k)} = \mathrm{Y}$	$3k\mathcal{K} + \sum_{j=1}^{k} N_j$	$3k\mathcal{K} + \sum_{j=1}^{k} N_j$
Total	$(7k+1)\mathcal{K}+$ $3(N_i - 1) + \sum_{j=1}^{k} N_j$	$(7k+1)\mathcal{K}+$ $3(N_i - 1) + \sum_{j=1}^{k} N_j$

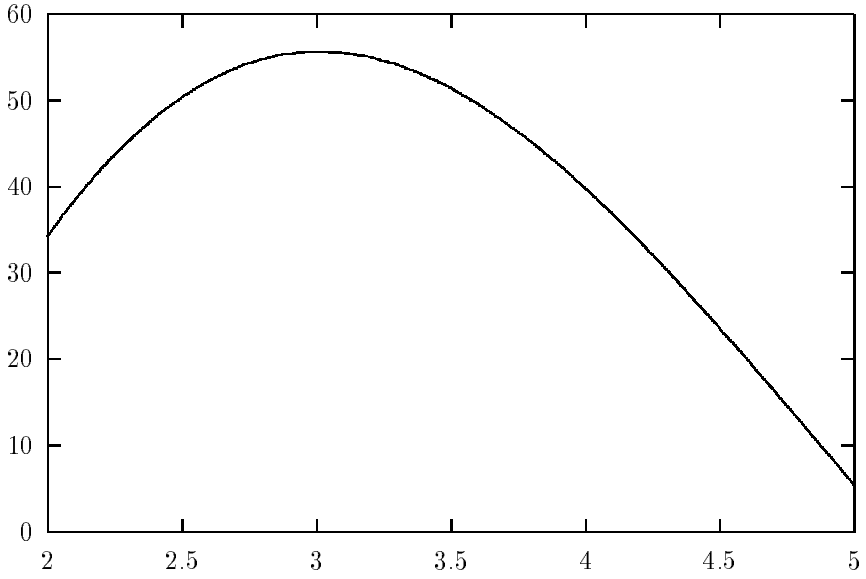
Table 4.3: Work to sweep the first direction of ADISC2.

Procedure	Work	
	+,-	*
$Z_1 := r_{s+1}I + X_1$	$3(N_1 - 1)$	$3(N_1 - 1)$
$U_i^{(s)} := X_iU^{(s)}$	$2\mathcal{K}$	$2\mathcal{K}$
$V_{ij}^{(s)} := H_j^iU(s)$	$2i\mathcal{K}$	$2i\mathcal{K}$
$\sum_{i=1}^{k} U_i^{(s)} + \sum_{i=2}^{k}\sum_{j=1}^{(i)} V_{ij}^{(s)} = V$	$(2^k(k+1) + k - 1)\mathcal{K}$	$(2^k + 2)k\mathcal{K}$
$\mathrm{R} := Z_1U^{(s)} - \omega V + \omega F$	$4\mathcal{K}$	$4\mathcal{K}$
Solve $Z_1U^{(s+1/k)} = \mathrm{R}$	$3\mathcal{K} + N_1$	$3\mathcal{K} + N_1$
Total	$(2^k(k+1) + k + 6)\mathcal{K}$ $+4N_1 - 3$	$((2^k + 2)k + 7)\mathcal{K}$ $+4N_1 - 3$

Table 4.4: Work to sweep the last $k-1$ directions of ADISC2.

Procedure	Work	
	+,-	*
$Z_i := r_{s+1}I + X_i$	$3(N_i - 1)$	$3(N_i - 1)$
$Y_1 := X_iU^{(s)}$	$2\mathcal{K}$	$2\mathcal{K}$
$Y := r_{s+1}U^{(s+(i-1)/k)} + Y_1$	$\mathcal{K}$	$\mathcal{K}$
Solve $Z_iU^{(s+i/k)} = Y$	$3\mathcal{K} + N_i$	$3\mathcal{K} + N_i$
Total	$6\mathcal{K} + 4N_i - 3$	$6\mathcal{K} + 4N_i - 3$

Figure 4.1: Efficiency improvement (per cent).



the ADISC2 iterative method is, per iteration, more efficient than the ADISC1 one when Ω is a k–dimensional domain for $k = 2, 3, 4, 5$ achieving its maximum relative efficiency, of approximately 55%, for three-dimensional problems. In more than 5–dimensions ADISC1 seems to win, increasing its relative efficiency as k increases.

4.3 Convergence analysis

In this Section we present our theoretical results concerning the convergence analysis of the ADISC methods proposed earlier in this Chapter. Although the analysis can be carried out for the generalized Helmholtz PDE problem (i.e. $a_i = 1, i = 1, \ldots, k$) $(1b) - (2b)$, for simplicity in the presentation, we consider the Poisson PDE equation (i.e. $\gamma = 0$). Recall that our two ADI methods for solving the interior Collocation equation can be written in the form

$$\mathrm{U}^{(s+1)} = M_{r_{s+1},\omega}\mathrm{U}^{(s)} + \mathrm{G} \tag{4.12}$$

where the iteration matrix $M_{r_{s+1},\omega}$ is given by relations (4.4) and (4.9)

for the ADISC1 and ADISC2 methods, respectively. We start our analysis by giving below analytic expressions of the eigenvalues of these iteration matrices.

Lemma 4.1 *The eigenvalues* $\underline{\nu}$ *of the iteration matrix* $M_{r_{s+1},\omega}$ *defined by the relation (4.4) for the ADISC1 scheme and by (4.9) for the ADISC2 scheme are given by*

$$\underline{\nu} = \underline{\nu}^{(\underline{\ell})} = 1 - \omega \left[\frac{\sum_{i=1}^{k} \frac{\alpha_i^{(\ell_i)}}{r_{s+1}} + C}{\prod_{i=1}^{k} \left(\frac{\alpha_i^{(\ell_i)}}{r_{s+1}} + 1 \right)} \right] \tag{4.13}$$

with $\underline{\ell} = (\ell_1, \ldots, \ell_k)$, $\ell_i = 1, \ldots, N_i - 1$, *where*

$$\alpha_i^{(\ell_i)} = \begin{cases} \frac{-6N_i^2 \lambda_i^{(\ell_i)}}{6+\lambda_i^{(\ell_i)}}, & \textit{for ADISC1} \quad O(h^2) \\ \frac{-N_i^2 \lambda_i^{(\ell_i)} \left(\lambda_i^{(\ell_i)} + 12 \right)}{2\left(\lambda_i^{(\ell_i)} + 6 \right)}, & \textit{for ADISC1} \quad O(h^4) \\ -N_i^2 \lambda_i^{(\ell_i)}, & \textit{for ADISC2} \quad O(h^2) \end{cases} , \tag{4.14}$$

$$\lambda_i^{(\ell_i)} = -4 sin^2 \left(\frac{\ell_i \pi}{2N_i} \right), \tag{4.15}$$

and $C = 0$ *for the ADISC1 scheme and*

$$C = \sum_{i=2}^{k} \frac{i}{r_{s+1}} \sum_{j=1}^{\binom{k}{i}} \frac{N_j^2}{6^{i-1}} \prod_{m=1}^{k} p_m^{(ij)} \lambda_m^{(\ell_m)}$$

for the ADISC2 scheme with

$$p_m^{(ij)} = \begin{cases} 1 & \textit{if there is } T_{-2} \textit{ at the position } m \textit{ of the matrix } H_{ij} \\ \frac{1}{\lambda_m^{(\ell_m)}} & \textit{if there is } I \textit{ at the position } m \textit{ of the matrix } H_{ij} \end{cases}$$

Proof: It is known ([16]) that the eigenvalues $\lambda_i^{(\ell_i)}$ of the matrix $T_{-2}^i \in \mathbb{R}^{N_i \times N_i}$ are given by the relation (4.15). It is easy to see that all matrices $T_a^i \in \mathbb{R}^{N \times N}$, $T_a^i \equiv tridiag(1, a, 1)$, $a \in \mathbb{R}$, have a common set of linearly independent eigenvectors and that the eigenvalues of the matrix T_a^i are given by $(a+2)+\lambda_i^{(\ell_i)}$. Relations (4.13) and (4.14) can be

now easily obtained by applying Lemma 2.1 to the iteration matrices given by (4.4) and (4.9). □

Denote now the error at the s^{th} iteration by $\mathrm{E}^{(s)} = \mathrm{U} - \mathrm{U}^{(s)}$ where U and $\mathrm{U}^{(s)}$ are the coefficients of u_Δ and $u_\Delta^{(s)}$ in (3.2), respectively. Then from (4.12) we have

$$\mathrm{E}^{(s+1)} = M_{r_{s+1},\omega}\mathrm{E}^{(s)}.$$

We can expand the error $\mathrm{E}^{(s)}$ in terms of the eigenvectors $\mathrm{p}_{i_j}, j = 1,\ldots,k$, of $D^{-1}A_i$ or of X_i (see Lemma 2.1) to get

$$\mathrm{E}^{(s)} = \sum_{i_1=1}^{N_1-1} \cdots \sum_{i_k=1}^{N_k-1} E^s_{i_1,\ldots,i_k} \mathrm{p}_{i_1} \otimes \ldots \otimes \mathrm{p}_{i_k}.$$

Combining the above two relations we have

$$\mathrm{E}^{(s+1)} = \sum_{i_1=1}^{N_1-1} \cdots \sum_{i_k=1}^{N_k-1} \nu_{i_1,\ldots,i_k}(r_{s+1}) E^s_{i_1,\ldots,i_k} \mathrm{p}_{i_1} \otimes \ldots \otimes \mathrm{p}_{i_k}$$

where $\nu_{i_1,\ldots,i_k}(r_{s+1})$ are the eigenvalues of the iteration matrix given by Lemma 4.1. Thus we have

$$\mathrm{E}^{(s)} = \sum_{i_1=1}^{N_1-1} \cdots \sum_{i_k=1}^{N_k-1} \left[\prod_{j=0}^{s} \nu_{i_1,\ldots,i_k}(r_j)\right] E^0_{i_1,\ldots,i_k} \mathrm{p}_{i_1} \otimes \ldots \otimes \mathrm{p}_{i_k}. \qquad (4.16)$$

For the ADISC1 scheme (the ADISC2 scheme can be treated similarly) we use relation (4.16), Lemma 2.1 and the fact that the function

$$g(x_1,\ldots,x_k) := \frac{\sum_{i=1}^k x_i}{\prod_{i=1}^k (x_i+1)} - 1,$$

is always negative, to obtain our first convergence result.

Theorem 4.1 *For any given set of positive acceleration parameters $r_{s+1}, s = 0,1,\ldots,$ and $0 < \omega \leq 2$ the proposed iterative methods ADISC1 and ADISC2 converge from any initial guess.*

For the rest of this section we restrict ourselves in 3 dimensions since most of our results can not be easily extended to more dimensions. Without loss of generality, and for simplicity in the presentation only,

we will assume that we have a uniform discretization grid of equal spacing in all dimensions (i.e. $N_i = N, i = 1, 2, 3$).

It is worth to note that the ADISC1 iterative method can be exact (except for round–off) in a number of iterations equal to the number of unknowns. This can be easily seen by observing that $\nu_{i_1,i_2,i_3}(r_s)$ is a fraction whose numerator is a cubic polynomial in r_s which has a real root $r_{i_1 i_2 i_3}$ for which the denominator does not vanish. Therefore, $\mathrm{E}^{(s)}$ can be made zero in $(N-1)^3$ iterations by setting $r_s = r_{i_1 i_2 i_3}$, $s = 1, \ldots, (N-1)^3$ and $i_j = 1, \ldots, (N-1), j = 1, 2, 3$.

If the number s of iterations required is known in advance, one can determine the optimum values for the sequence of iteration parameters r_s by solving a minimax problem. This minimax problem becomes a whole sequence of such problems since in practice we very rearly know s in advance. Our objective in this study is to choose a sequence of "good" acceleration parameters r_s that will reduce the number of iterations required to produce a satisfactory approximation to the solution. We will do that for the Douglas scheme, i.e. we set $\omega = 2$.

We start by letting for $j = 1, 2, 3$

$$\begin{aligned} \zeta_\ell &= \tfrac{12}{r_\ell h^2}, \ \ell = 1, 2, \ldots, \\ \xi_{i_j} &= \tfrac{\sin^2 \frac{i_j \pi}{2N}}{3 - 2\sin^2 \frac{i_j \pi}{2N}}, \quad i_j = 1, \ldots, N-1 \ . \end{aligned} \tag{4.17}$$

Then we have that

$$\nu_{i_1,i_2,i_3}(r_\ell) = 1 - 2\frac{\zeta_\ell \xi_{i_1} + \zeta_\ell \xi_{i_2} + \zeta_\ell \xi_{i_3}}{(1 + \zeta_\ell \xi_{i_1})(1 + \zeta_\ell \xi_{i_2})(1 + \zeta_\ell \xi_{i_3})}. \tag{4.18}$$

We would like to find a sequence $\{\zeta^{(\ell)}\}, \ell = 1, \ldots, P$, such that

$$\mu \le \zeta^{(\ell)} \xi_{i_j} \le \nu$$

for every ℓ for at least one j. μ and ν are parameters that will be determined later on. Since $\xi_1 = \frac{\sin^2 \frac{\pi}{2N}}{3 - 2\sin^2 \frac{\pi}{2N}}$ and $\xi_{N-1} \approx 1$ we define $\xi^{(1)} = \xi_1$ and the sequence $\{\zeta^{(\ell)}, \xi^{(\ell)}\}$ such that

$$\zeta^{(\ell)} \xi^{(\ell)} = \mu \ \text{ and } \ \zeta^{(\ell)} \xi^{(\ell+1)} = \nu$$

from which we have for $\ell = 1, \ldots, P-1$ that

$$\zeta^{(\ell)} = \mu \left(\frac{\mu}{\nu}\right)^{\ell-1} \frac{3 - 2\sin^2 \frac{\pi}{2N}}{\sin^2 \frac{\pi}{2N}}, \tag{4.19}$$

$$\xi^{(\ell)} = \left(\frac{\nu}{\mu}\right)^{\ell-1} \frac{\sin^2 \frac{\pi}{2N}}{3 - 2\sin^2 \frac{\pi}{2N}}, \tag{4.20}$$

for which $\mu \leq \zeta^{\ell} \xi \leq \nu$ for every ξ such that $\xi^{(\ell)} \leq \xi \leq \xi^{(\ell+1)}$. We stop generating terms when we cross 1, i.e. when $\xi^{(P+1)} \approx 1 \approx \xi_{N-1}$, and thus we have that

$$P = \log^{-1}\left(\frac{\nu}{\mu}\right) \log\left(\frac{3 - 2\sin^2 \frac{\pi}{2N}}{\sin^2 \frac{\pi}{2N}}\right). \tag{4.21}$$

Using the expressions for ζ_ℓ in (4.17) and (4.19) we obtain the following expression for the acceleration parameters

$$r_s = \frac{12N^2}{\mu}\left(\frac{\nu}{\mu}\right)^{s-1} \frac{\sin^2 \frac{\pi}{2N}}{3 - 2\sin^2 \frac{\pi}{2N}} \quad s = 1, \ldots, P. \tag{4.22}$$

We proceed by stating next two Lemmata found in [9].

Lemma 4.2 *Let*

$$\rho \equiv \rho(a, b, c) = 1 - \frac{2(a + b + c)}{(1 + a)(1 + b)(1 + c)} \tag{4.23}$$

and

$$\begin{aligned}\hat{\rho}(\mu, \nu) \equiv \max\{|\rho(a, b, c)| : [\mu \leq a \leq \nu; 0 \leq b, c \leq \nu] \ \text{ or} \\ [\mu \leq b \leq \nu; 0 \leq a, c \leq \nu] \ \text{ or } \ [\mu \leq c \leq \nu; 0 \leq a, b \leq \nu]\}.\end{aligned}$$

Then if $\mu < 1 < \nu$

$$\hat{\rho}(\mu, \nu) = \max\left[1 - \frac{6\nu}{(1 + \nu)^3}, 1 - \frac{2\mu}{1 + \mu}\right]. \tag{4.24}$$

Lemma 4.3 *Let* $\nu \geq 1$ *and*

$$\mu = \frac{3\nu(1+\nu)^{-3}}{1-3\nu(1+\nu)^{-3}} = \frac{3\nu}{1-3\nu^2+\nu^3}; \tag{4.25}$$

then

$$\hat{\rho}(\mu,\nu) = 1 - \frac{6\nu}{(1+\nu)^3} = 1 - \frac{2\mu}{1+\mu}. \tag{4.26}$$

If we denote by R_s the operator which maps $\mathrm{E}^{(0)}$ to $\mathrm{E}^{(s+1)}$, then we easily see from (4.16) that its L_2 norm is

$$||R_s|| = \max_{i_1,\ldots,i_k} \left[\prod_{j=0}^{s} \nu_{i_1,\ldots,i_k}(r_j) \right]. \tag{4.27}$$

From Lemmata 4.2 and 4.3 we have that if we iterate P times using the parameters given in (4.22) we have

$$||R_P|| \leq \hat{\rho}(\mu,\nu)$$

and if we iterate mP times using the parameters in (4.22) cyclicly (i.e. $r_{mP+\ell} = r_\ell$, $m = 1, 2, \ldots$) we have that

$$||R_{mP}|| \leq \hat{\rho}^m(\mu,\nu).$$

If we want now to have $||R_{mP}|| \approx \epsilon$ we get

$$m \approx \frac{\log \epsilon}{\log \hat{\rho}(\mu,\nu)}$$

and therefore

$$mP \approx \frac{-\log \epsilon \log \left(\frac{3-2\sin^2 \frac{\pi}{2N}}{\sin^2 \frac{\pi}{2N}} \right)}{\log \left(\hat{\rho}(\mu,\nu)\right)^{-1} \log \left(\frac{\nu}{\mu}\right)}. \tag{4.28}$$

Assuming (for our convenience) that μ satisfies (4.25) and evaluating the denominator of relation (4.28) for $\nu = 1(.01)2$ we find that the number of iterations mP is minimized for $\nu = 1.78$ and $\mu = 0.33$. We can summarize the above discussion (the $O(h^4)$ case can be treated in a similar way) in the following Theorem.

Theorem 4.2 *If the acceleration parameters r_s are selected as*

$$r_s = \begin{cases} \frac{12N^2}{\mu}\left(\frac{\nu}{\mu}\right)^{s-1} \frac{\sin^2\frac{\pi}{2N}}{3-2\sin^2\frac{\pi}{2N}} & \text{in the } O(h^2) \text{ case} \\ \frac{8N^2}{\mu}\left(\frac{\nu}{\mu}\right)^{s-1} \frac{\sin^2\frac{\pi}{2N}\left(3-\sin^2\frac{\pi}{2N}\right)}{6-4\sin^2\frac{\pi}{2N}} & \text{in the } O(h^4) \text{ case} \end{cases}, s = 1, \ldots, P \tag{4.29}$$

and used cyclicly, i.e. $r_{iP+s} = r_s$, $i = 1, 2, \ldots$, $s = 0, \ldots, P-1$, the ADISC1 iterative methods with $\omega = 2$ will reduce the initial error $\mathrm{E}^{(0)}$ by a preassigned factor of ϵ in mP iterations where $m \approx \frac{\log\epsilon}{\log.5}$

$$P \approx \begin{cases} 0.59\log\left(\frac{3-2\sin^2\frac{\pi}{2N}}{\sin^2\frac{\pi}{2N}}\right) & \text{in the } O(h^2) \text{ case} \\ 0.59\log\left(\frac{6-4\sin^2\frac{\pi}{2N}}{\sin^2\frac{\pi}{2N}\left(3-\sin^2\frac{\pi}{2N}\right)}\right) & \text{in the } O(h^4) \text{ case} \end{cases}. \tag{4.30}$$

In order to further increase the efficiency of our ADISC1 scheme we would like to find a value of ω for which either the asymptotic rate of convergence is a maximum or the number of calculations needed to reduce the norm of the matrix operator that maps the error vector $\mathrm{E}^{(0)}$ to the error vector $\mathrm{E}^{(s)}$, $||R_s||$, below a preassigned tolerance is a minimum. We will follow the latter approach and the techniques found in [14]. Recall that the eigenvalues of the ADISC1 iteration matrix are given by

$$\rho = \rho(a,b,c) \equiv \underline{\nu} = 1 - \omega\frac{a+b+c}{(1+a)(1+b)(1+c)} \tag{4.31}$$

where $a = \zeta_\ell\xi_{i_1}$, $b = \zeta_\ell\xi_{i_2}$ and $c = \zeta_\ell\xi_{i_3}$ with ζ_ℓ and ξ_{i_j} given by (4.17). We decide to use as acceleration parameters r_s the ones obtained above for $\omega = 2$. Thus we replace, for $\ell = 1, \ldots, n_0$, $(n_0 = P)$ the ζ_ℓ by $\zeta^{(\ell)}$ that are given in (4.19). So if for any triple (a,b,c) such that $a \geq b \geq c$ we iterate n_0 times then there exists a $n^* \in \{1, \ldots, n_0\}$ such that one of the following is satisfied

$$\mu \leq a_{n^*} \leq \nu, \quad t\mu \leq b_{n^*}, c_{n^*} \leq \nu \tag{4.32}$$

$$\mu \leq b_{n^*} \leq \nu, \quad t\mu \leq a_{n^*}, c_{n^*} \leq \nu \tag{4.33}$$

$$\mu \leq c_{n^*} \leq \nu, \quad t\mu \leq a_{n^*}, b_{n^*} \leq \nu \tag{4.34}$$

with

$$t = \frac{6\tan^2\frac{\pi}{2N} - 4\sin^2\frac{\pi}{2N}}{6 - 4\sin^2\frac{\pi}{2N}}. \tag{4.35}$$

For $n \neq n^*$ the considered triple does not satisfy one of the (4.32), (4.33) and (4.34) anymore. Instead, we have,

$$t\mu \leq a_n, b_n, c_n \leq t^{-1}\nu. \tag{4.36}$$

By simply taking derivatives and considering several cases we can easily prove the following Lemma.

Lemma 4.4 *Let t be given by relation (4.35), $f = f(a,b,c) = \frac{a+b+c}{(1+a)(1+b)(1+c)}$,*

$$f^*_{max} = \max\{f(a,b,c), \text{for} \quad a,b,c \text{ that satisfy (4.36)}\},$$

$$f_{max} = \max\{f(a,b,c), \text{for} \quad a,b,c \text{ that satisfy one of (4.32), (4.33) and (4.34)}\}$$

and

$$f_{min} = \min\{f(a,b,c), \text{for} \quad a,b,c \text{ that satisfy one of (4.32), (4.33)and(4.34)}\}.$$

Then we have

$$f^*_{max} = \frac{2t\mu + t^{-1}\nu}{(1+t\mu)^2(1+t^{-1}\nu)}, \tag{4.37}$$

$$f_{max} = \frac{2t\mu + \nu}{(1+t\mu)^2(1+\nu)}, \tag{4.38}$$

and

$$f_{min} = \min\left\{\frac{\mu + 2t\mu}{(1+t\mu)^2(1+\mu)}, \frac{3\nu}{(1+\nu)^3}\right\}. \tag{4.39}$$

We are now in position to determine the optimum value of ω as function of N, μ and ν as follows. From (4.31) we observe that the possible range of ω is limited by the fact that we must have $|\rho| < 1$. So for triples a, b, c that satisfy the inequality (4.36) we have $-1 < 1 - \omega f < 1$ from which we have that

$$0 < \omega \leq \frac{2}{f^*_{max}}. \tag{4.40}$$

With ω in the above range we can easily verify that $|\rho| < 1$ for any triple (a, b, c) that satisfies one of the inequalities (4.32), (4.33) and (4.34). For convenience we take

$$\frac{\mu + 2t\mu}{(1 + t\mu)^2(1 + \mu)} = \frac{3\nu}{(1 + \nu)^3}. \tag{4.41}$$

We can now easily see that for optimality we must have $1 - \omega f_{min} = -(1 - \omega f_{max})$, that is

$$\omega = \frac{2}{f_{min} + f_{max}}. \tag{4.42}$$

To conclude, we consider the following two cases for ω_{opt}:

If $\frac{2}{f_{min}+f_{max}} \leq \frac{2}{f^*_{max}}$ and if we choose ω_{opt} as in relation (4.42) which also satisfies (4.40) then $|\rho| \leq 1 - \omega_{opt} f_{min} \equiv \rho(\mu, \nu)$

If $\frac{2}{f_{min}+f_{max}} > \frac{2}{f^*_{max}}$ and for ω_{opt} in the range defined by (4.40) we have that $1 - \omega_{opt} f_{min} < -(1 - \omega_{opt} f_{max})$ and so we have again $|\rho| \leq \rho(\mu, \nu)$.

So by choosing

$$\omega_{opt} = min\left\{\frac{2}{f_{min} + f_{max}}, \frac{2}{f^*_{max}}\right\} \tag{4.43}$$

we have $|\rho| < 1 - \omega_{opt} f_{min}$.

Going back to (4.41), we can see that function $y(\nu) = \frac{3\nu}{(1+\nu)^3}$ decreases for $\nu \geq 1$ and so we have $y(\nu) \leq y(1) = 3/8$ for $\nu \geq 1$. Thus to find the optimum pair of (μ, ν), which minimizes the total number of iterations, we search among the pairs (μ, ν) that satisfy (4.39), $0 < \mu \leq 1$ and

$$0 < \frac{\mu + 2t\mu}{(1 + t\mu)^2 (1 + \mu)} \leq \frac{3}{8}$$

and maximize the function $\log \rho(\mu, \nu)^{-1} \log(\mu/\nu)$.

Since $0 < f_{min} < 3/8$ and we want $0 < 1 - \omega f_{min} \leq 1$, the desired inequality is equivalent to $0 < 1 - \frac{3}{8}\omega \leq 1$ or

$$0 \leq \omega < \frac{8}{3}.$$

Chapter 5

Numerical Experiments

In this Chapter we present the results of our numerical experiments obtained for the $O(h^2)$ and the $O(h^4)$ ADISC1 iterative methods.

Our computer implementation of these ADISC1 methods is based on software components available to us through BLAS and TENPACK. BLAS (Basic Linear Algebra Subroutines) ([24], [8]), is a widely used collection of routines that implement basic linear algebra operations like inner product of vectors, matrix vector product etc. and widely available, on most machines in a highly optimized assembly form. TENPACK ([4]) is a software package that implements, on top of LINPACK ([7]), algorithms for the basic tensor product operations proposed in [6]. We had to make some modifications and extensions of certain TENPACK routines for our needs. It should be pointed out that our codes perform only very few floating point operations outside BLAS and TENPACK.

For the discretization of the PDE domain Ω we have used a uniform, equal in all dimensions, discretization with grid spacing h. As initial guess we have used the zero function. We stop the iterations of our ADISC schemes when we achieve three significant digits correct in the max-norm of the relative error or when the max–norm of the difference of two successive iterations ($||\mathrm{U}^{(s+1)} - \mathrm{U}^{(s)}||_\infty$) is less than 10^{-7}.

All experiments presented in this section have been performed in double precision Fortran on a CONVEX C-3420 configured with 128Mb RAM and all CPU times reported are in seconds.

For our experimental study we have consider three PDE model prob-

lems defined by the following PDE equations

PDE 1 The Poisson equation

$$u_{xx} + u_{yy} + u_{zz} = f$$

PDE 2 The generalized Helmholtz equation

$$u_{xx} + u_{yy} + u_{zz} - \gamma u = f$$

with $\gamma(x, y, z) = 100 + \cos(2\pi x) + \sin(3\pi y) + \cos(\pi z)$

PDE 3 The general elliptic equation

$$(1 + x^2)u_{xx} + \exp(y - 1)u_{yy} + (3 + \sin^2(\pi z))u_{zz} + \gamma u = f$$

with $\gamma(x, y, z) = e^{2x}\cos(3\pi x) + y^3 - 2y + \sin(\pi z)\cos(2\pi z)$

on the unit cube with homogeneous Dirichlet boundary conditions. The right–hand side function f was selected so that the true solution of PDE 1 and PDE 3 is

$$u(x, y, z) = 10e^{x+y+z}(x^2 - x)(y^2 - y)(z^2 - z)$$

and the true solution of PDE 2 is

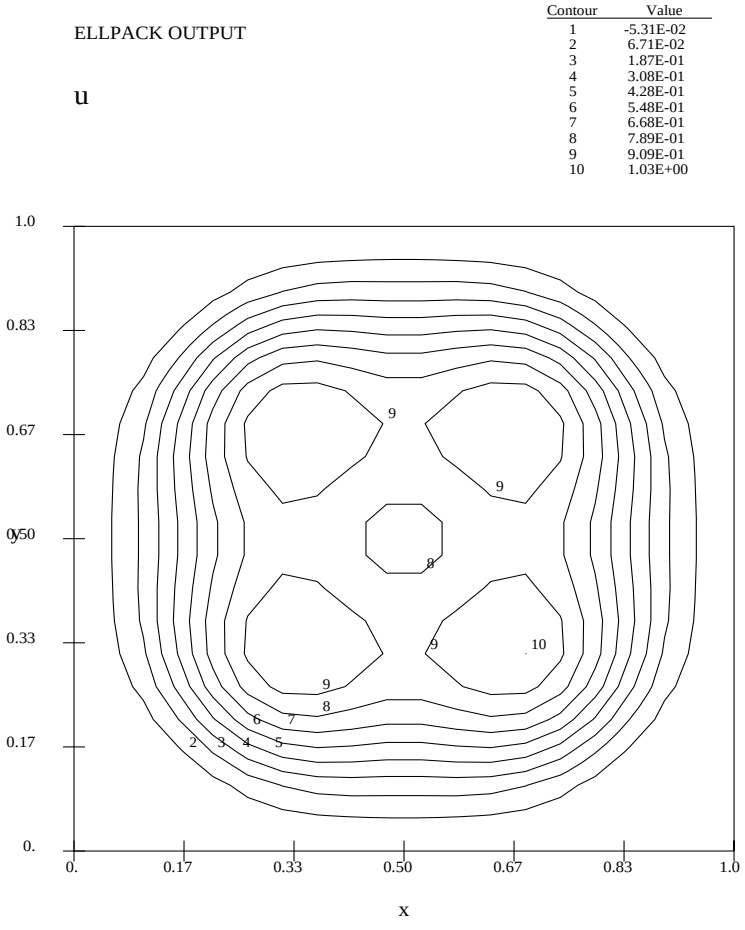
$$u(x, y, z) = -0.31c(x)s(x)(y^2 - y)c(y)s(z)\left(\frac{1}{1+(4((x-0.5)^2+(y-0.5)^2+(z-0.5)^2))^4} - 0.5\right)$$

where $c(t) := 5.4 - \cos(4\pi t)$ and $s(t) := \sin(\pi t)$. As we easily see all solutions are analytic with the one associated with the model problem PDE 2 (a problem from stratospheric physics [30] whose computed solution on the plane $z = .5$ is shown in Figure 5.1) being oscillatory.

In Table 5.1 we present our results for the $O(h^2)$ scheme. Specifically we present, for different number of discretization points N ($N = 5(5)30$) and for the three PDE model problems:

The achieved accuracy: By *error* in the third column of Table 5.1 we denote the maximum norm of the error at the discretization points i.e. $error = ||u - u_\Delta||_\infty$.

Figure 5.1: Contour plot of the computed solution for model problem PDE 2 at $z = 0.5$.



The order of convergence: We compute an estimation of the order of the discretization from the expression

$$-log\frac{(||(u-u_{\Delta_1})||_\infty}{||(u-u_{\Delta_2})||_\infty}/log\left(\frac{h_1}{h_2}\right),$$

where u_{Δ_i} represents the Collocation spline approximate obtained using a uniform grid step h_i in all direction. As we easily see, the order of convergence is $O(h^2)$ for all three model problem as theoretically was expected.

Optimum ω: The optimum value of the relaxation parameter ω was obtained experimentally by systematically searching the value of ω in $[0,4]$ which corresponds to the minimum number of iterations required by the ADISC1 method to satisfy the stopping criteria. These experimental estimations of ω_{opt} agree within a reasonable accuracy with the theoretical ones obtained (when applicable) by relation (4.43). It should be pointed out that the optimal point of ω for the model problem PDE 2 becomes an interval of length up to .5. During this systematic search for ω_{opt} we were able to confirm the theoretically obtained interval of convergence for ω which agrees with remarkable accuracy with the experimental one.

Number of iterations: In the last three columns of Table 5.1 we present the number of iterations required by the Douglas–Rachford, the Douglas and the optimum ADISC1 iteration methods to satisfy the stopping criteria. As it can be observed there is a significant decrease on the number of iterations as we move from the Douglas–Rachford scheme to the Douglas scheme and to the optimum scheme for all the PDE model problems. We can also observe another nice feature of, at least, our optimal scheme in the fact that the associated number of iterations required for convergence for a specific discretization step, remains constant for all model problems.

The same as above observations can be made for the data in Table 5.2 where we present, in the same to the $O(h^2)$ case way, our results

Table 5.1: Error, order of convergence, ω_{opt} and number of iterations for the $O(h^2)$ scheme applied to the three PDE model problems.

					Number of Iterations		
	N	error	order	ω_{opt}	$\omega = 1$	$\omega = 2$	$\omega = \omega_{opt}$
PDE 1	5	1.01e-3		2.3	21	7	5
	10	3.20e-4	1.88	2.3	37	13	7
	15	1.55e-4	1.93	2.3	44	13	9
	20	8.99e-5	2.00	2.3	57	18	13
	25	5.90e-5	1.97	2.3	62	21	14
	30	4.18e-5	1.96	2.3	75	18	13
PDE 2	5	7.87e-2		2.1 – 2.3	17	6	6
	10	3.29e-2	1.43	2.1 – 2.6	20	10	8
	15	1.52e-2	2.05	2.2 – 2.6	22	8	6
	20	9.43e-3	1.76	2.2 – 2.6	42	10	10
	25	6.07e-3	2.06	2.1 – 2.4	34	13	10
	30	4.27e-3	2.00	2.2 – 2.6	37	14	10
PDE 3	5	1.01e-3		2.1 – 2.2	15	3	4
	10	3.31e-4	1.84	2.2	23	10	7
	15	1.53e-4	2.05	2.1	28	7	7
	20	9.16e-5	1.89	2.1	45	13	10
	25	6.00e-5	1.98	2.1	57	17	14
	30	4.21e-5	2.01	2.1	49	14	13

obtained by the $O(h^4)$ ADISC1 scheme applied to the model problem PDE 1. In addition we can point out that the number of iterations required does not increase significantly as we move from the $O(h^2)$ to the $O(h^4)$ scheme.

To check the efficiency of our implementations we give in Table 5.3 the average CPU time required to perform one iteration associated with the experiments presented in Table 5.1. A least squares logarithmic fit of these data shows that the per iteration total time T is given by $T = .0003N^{2.93}$. This experimental estimation confirms our theoretical one obtained in the complexity analysis in Section 4.2 (see relations (4.10) and (4.11)).

In Figures 5.2 and 5.3 we present a detailed history of the con-

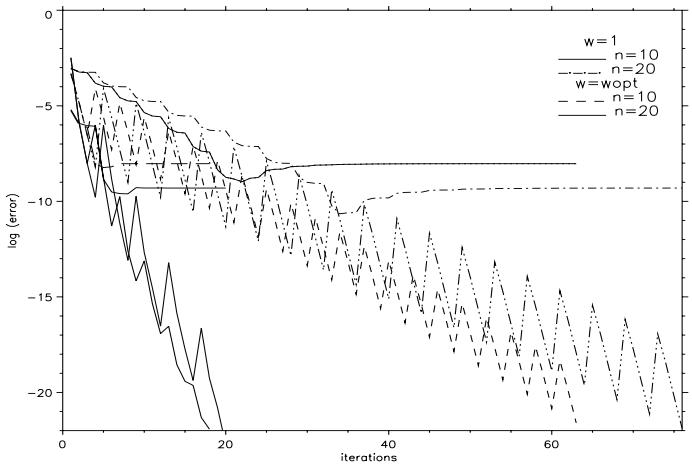
Table 5.2: Error, order of convergence, ω_{opt} and number of iterations for the $O(h^4)$ scheme applied to model problem PDE 1.

				Number of Iterations		
N	error	order	ω_{opt}	$\omega = 1$	$\omega = 2$	$\omega = \omega_{opt}$
5	3.01e-4		2.3	31	11	9
10	3.31e-5	3.91	2.3	51	16	10
15	7.90e-6	3.82	2.3	60	19	12
20	3.16e-6	3.66	2.3	61	19	13
25	1.51e-6	3.36	2.3	73	21	13
30	8.95e-7	3.19	2.3	77	25	17

Table 5.3: Per iteration CPU time for the ADISC schemes.

	$O(h^2)$		
N	PDE 1	PDE 2	PDE 3
5	.04	.04	.06
10	.28	.28	.29
15	.93	.93	.97
20	2.2	2.19	2.23
25	4.24	4.20	5.20
30	7.3	7.19	7.40

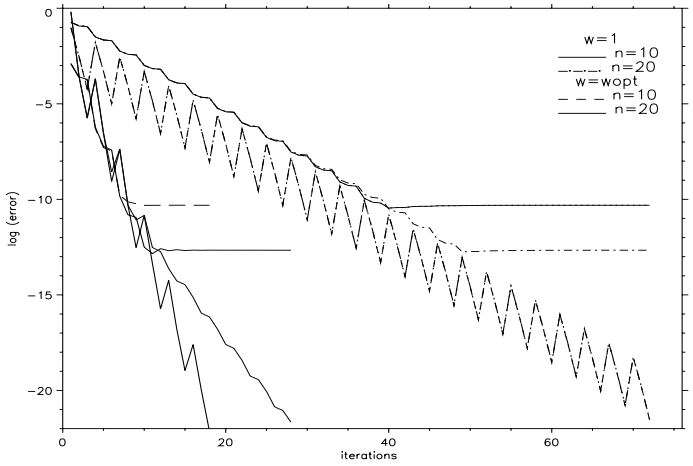
Figure 5.2: History of convergence for $O(h^2)$ and PDE 1.



vergence of the $O(h^2)$ and the $O(h^4)$ ADISC1 methods applied to the model problem PDE 1 with all stopping criteria removed. We can split both Figures in two branches, the upper one corresponding to the Douglas–Rachford scheme and the lower one corresponding to the optimum scheme. Each of these branches can be further splitted into two groups of lines corresponding to *error* being $||u - u_\Delta||_\infty$ (lines '———' and '– . — .' on the upper branch) or being $||\mathrm{U}^{(s+1)} - \mathrm{U}^{(s)}||_\infty$ (lines '- - - - ' and '– ... —' on the upper branch). We easily see the effect of using the acceleration parameters cyclicly which is apparent for the $||\mathrm{U}^{(s+1)} - \mathrm{U}^{(s)}||_\infty$ case and almost invisible in the $||u - u_\Delta||_\infty$ case.

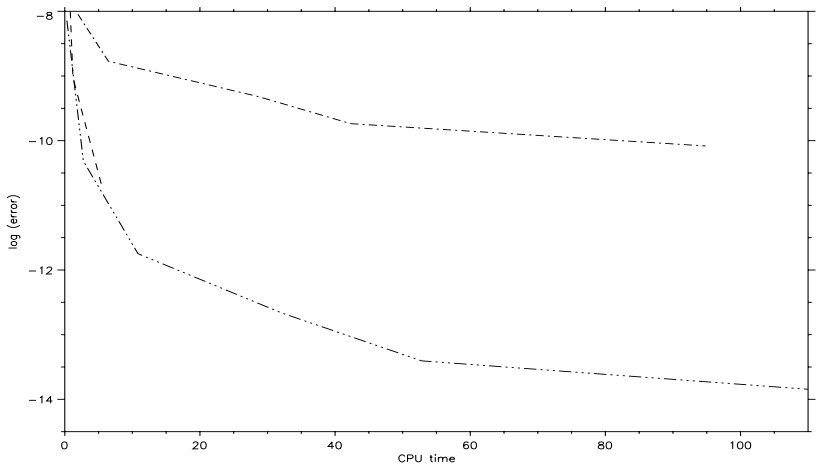
In order to rank the proposed ADISC1 iterative method we compare it with the two 3-dimensional PDE solving methods available in ELLPACK ([30]), namely the standard 7–point star finite difference method which is an $O(h^2)$ method and an $O(h^4)$ 27–point difference method called HODIE. In Figure 5.4 we plot the logarithm of the error $||u - u_\Delta||_\infty$ versus the required CPU time to achieve it. The data for the 7–point star method do not appear in the graph since its efficiency is too low and the associated line is further above the line -8 parallel to the x–axis. Although Figure 5.4 compares the implementation and not the actual methods themselves we can claim that the proposed methods easily outperform the standard 7–point star method

Figure 5.3: History of convergence for $O(h^4)$ and PDE 1.



and perform equally well with the high order HODIE method. Here we should point out that we were unable to obtain more points to extend the HODIE line in the graph further due to memory limitations. Furthermore the applicability of the HODIE method is restricted to generalized Helmholtz problems only and can not be applied to general self adjoint PDEs.

Figure 5.4: Efficiency of $O(h^2)$ ADISC1 (_ _) , $O(h^4)$ ADISC1 (_ . . . _) and $O(h^4)$ HODIE (- - -) methods.



Chapter 6

Synopsis and conclusions

In this study we have formulated, analyzed and implemented efficient ADI iterative methods for the solution of the linear algebraic systems which arise from the discretization of Self–Adjoint Elliptic PDE problems in k dimensions using $O(h^2)$ and $O(h^4)$ cubic Spline Collocation.

Two ADISC schemes have been proposed, namely the ADISC1 and the ADISC2. A detailed per iteration complexity analysis has been presented showing us that the ADISC2 scheme is much more efficient (within an iteration) for $k = 2, 3, 4$ dimensions while ADISC1 takes over for $k > 5$. More specifically we have shown that the total number of operations required to perform one iteration step in the $O(h^2)$ ADISC1 and ADISC2 schemes are $(18k^2 - k - 1)N^k + O(N)$ and $(2^k(k+2) + 15k + 1)N^k + O(N)$, respectively, where N represents the number of discretization points in one direction.

A convergence analysis of the proposed schemes has been carried out for the Poisson PDE and can be easily extended to the generalized Helmholtz PDE. Specifically we first prove the convergence of the $O(h^2)$ and $O(h^4)$ ADISC1 and the $O(h^2)$ ADISC2 schemes in k dimensions for any set of positive acceleration parameters and for $0 < \omega < 2$. The rest of our analysis is restricted to our ADISC1 schemes and to 3 dimensions since most of our results can not be extended to more dimensions. We obtain values for the acceleration parameters r_s which are not optimal but reasonably good. Furthermore we estimate the number of iterations required by our schemes to reduce the initial error

by a preassigned factor ϵ. Finally we obtain analytic expressions for the optimal values and give intervals of convergence of the relaxation parameter ω.

We have implemented out ADISC1 schemes using software components that take full advantage of the tensor product formulation of our iterative methods. Our extensive numerical results confirm the increased efficiency of the methods predicted by our complexity analysis which is verified by our timing results. Furthermore a careful experimental comparison has been carried out which shows that our ADI schemes out–perform well known methods. We have used our implementation to solve three PDE model problems, namely a Poisson, a Helmholtz and a general PDE on a unit cube. The experimental data obtained exhibit a good agreement with the theoretical results obtained and show that these results are valid for more general PDE problems.

Bibliography

[1] D. N. Arnold and J. Saraven, *On the asymptotic convergence of spline - collocation methods for partial differential equations*, SIAM J. Numer. Anal., 21 (1984), pp. 459–472.

[2] B. Bialecki, *An alternating direction implicit method for orthogonal spline collocation linear systems*, Numer. Math., 59 (1991), pp. 413–429.

[3] J. Bonomo, P. Buis, and W. R. Dyksen, *Parallel adi methods on shared memory machines*, Submited, (1992).

[4] P. Buis and W. Dysken, *Tenpack*, Tech. Report CSD-TR-90-048, Purdue University, W. Lafayette. IN, 1990.

[5] K. Cooper and P. Prenter, *Alternating direction collocation method for separable elliptic parial differential equations*, SIAM J. Numer. Anal., 28 (1991), pp. 711–727.

[6] C. de Boor, *Efficient computer manipulation of tensor products*, ACM Trans. Math. Softw., 5 (1979), pp. 173–182.

[7] J. Dongarra, J. Bunch, C. Moler, and G. Stewart, *LINPACK User's Guide*, SIAM, Philadelphia, PA, 1979.

[8] J. J. Dongarra, J. Du Croz, I. Duff, and S. Hammarling, *A set of level 3 basic linear algebra subprograms*, ACM Trans. Math. Softw., 16 (1990), pp. 1–17.

[9] J. Douglas, *Alternating direction methods for three space variables*, Numer. Math., 4 (1962), pp. 41–53.

[10] J. Douglas and H. Rachford, *On the numerical solution of heat conduction problems in two or three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–438.

[11] W. R. Dyksen, *A tensor product generalized ADI method for elliptic problems on cylindrical domains with holes*, J. Comp. Appl. Maths, 16 (1986), pp. 43–58.

[12] ——, *Tensor product generalized ADI methods for elliptic problems*, SIAM J. Numer. Anal., 24 (1987), pp. 59–76.

[13] W. R. Dyksen, *A tensor product generalized ADI methods for the method of planes*, Num. Meth. for PDEs, 4 (1988), pp. 283–300.

[14] A. Hadjidimos, *On a generalized alternating direction implicit method for solving Laplace's equation*, IMA J. Numer. Anal., 10 (1968), pp. 324–328.

[15] A. Hadjidimos, E. Houstis, J. Rice, and E. Vavalis, *On the iterative solution of line spline collocation schemes for elliptic PDEs*, Tech. Report CSD-TR-020, Purdue University, W. Lafayette. IN, 1991.

[16] ——, *Iterative line cubic spline collocation methods for elliptic partial differential equations in several dimensions*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 715–734.

[17] P. Halmos, *Finite Dimensional Vector Spaces*, D. van Nostrand, New York, NY, 1958.

[18] E. Houstis, *Collocation methods for linear elliptic problems*, BIT, 18 (1978), pp. 301–310.

[19] E. Houstis, J. Rice, and E. Vavalis, *Spline collocation methods for elliptic partial differential equations*, in Advances in Computer Methods for Partial Differential Equations, R. Vichnevetsky and R. Stepleman, eds., vol. V, IMACS, 1984, pp. 191–194.

[20] E. Houstis, E. Vavalis, and J. Rice, *Convergence of $O(h^4)$ cubic spline collocation methods for elliptic partial differential equations*, SIAM J. Numer. Anal., 6 (1988), pp. 54–74.

[21] E. N. Houstis, W. F. Mitchell, and J. R. Rice, *GENCOL: Collocation on general domains with bicubic Hermite polynomials*, ACM Trans. Math. Softw., 11 (1985), pp. 413–415.

[22] ——, *INTCOL and HERMCOL: Collocation on rectangular domains with bicubic Hermite polynomials*, ACM Trans. Math. Softw., 11 (1985), pp. 416–418.

[23] S. L. Johnsson, Y. Saad, and M. Schultz, *Alternating direction methods on multiprocessors*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 686–700.

[24] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, *Basic linear algebraic subprogram for Fortran usage*, ACM Trans. Math. Softw., 5 (1979), pp. 324–325.

[25] B. K. Lucas, *Error bounds for interpolating cubic splines under various end conditions*, SIAM J. Numer. Anal., 11 (1974), pp. 569–584.

[26] R. Lynch, J. Rice, and D. Thomas, *Tensor product analysis of alternating direction implicit methods*, Journal of the Society for Industrial and Applied Mathematics, 13 (1965), pp. 995–1006.

[27] A. Mitchell and G. Fairweather, *Improved forms of the alternating direction methods of Douglas, Peaceman, and Rachford for solving parabolic and elliptic equations*, Numer. Math., 6 (1964), pp. 285–292.

[28] J. M. Ortega and R. G. Voigt, *Solution of partial differential equations on vector and parallel computers*, SIAM Review, 27 (1985), pp. 149–240.

[29] P. Percell and M. F. Wheeler, *A c^1 finite element collocation method for elliptic equations*, SIAM J. Numer. Anal., 17 (1980), pp. 605–622.

[30] J. Rice and R. Boisvert, *Solving Elliptic Problems Using ELLPACK*, Springer-Verlag, New York, NY, 1985.

[31] S. Rubin and R. Graves, *Viscous flow solution with a cubic spline approximation*, Comput. & Fluids, 3 (1974), pp. 1–36.

[32] R. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc., New Jersey, 1962.

[33] D. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, NY, 1971.