Department of Computer Science Technical Reports

Department of Computer Science

1994

The Analysis of Iterative Elliptic PDE Solvers Based on The Cubic Hermite Collocation Discretization

Yu-Ling Lai

Apostolos Hadjidimos

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
94-036

THE ANALYSIS OF ITERATIVE ELLIPTIC PDE SOLVERS BASED ON THE CUBIC HERMITE COLLOCATION DISCRETIZATION

Yu-Ling Lai
Apostolos Hadjidimos
Elias N. Houstis
John R. Rice

CSD TR-94-036
May 1994

THE ANALYSIS OF ITERATIVE ELLIPTIC PDE SOLVERS BASED ON THE CUBIC HERMITE COLLOCATION DISCRETIZATION[*]

YU-LING LAI[†], APOSTOLOS HADJIDIMOS[‡], ELIAS N. HOUSTIS[‡], AND JOHN R. RICE[‡]

Abstract. Collocation methods based on bicubic Hermite piecewise polynomials have been proven effective techniques for solving second-order linear elliptic PDEs with mixed boundary conditions. The corresponding linear system is in general non-symmetric and non-diagonally dominant. Iterative methods for their solution are not known and they are currently solved using Gauss elimination with scaling and partial pivoting. Point iterative methods do not converge even for the collocation equations obtained from model PDE problems. The development of efficient iterative solvers for these equations is necessary for three-dimensional problems and their parallel solution, since direct solvers tend to be space bound and their parallelization is difficult. In this thesis, we develop block iterative methods for the collocation equations of elliptic PDEs defined on a rectangle and subject to uncoupled mixed boundary conditions. For model problems of this type, we derive analytic expressions for the eigenvalues of the block Jacobi iteration matrix and determine the optimal parameter for the block SOR method. For the case of general domains, the iterative solution of the collocation equations is still an open problem. We address this open problem by generalizing *interior collocation* method for PDEs defined on rectilinear regions, study the structure of these equations under different ordering schemes, and apply AOR and CG type iterative solvers to them. Another objective of this thesis is to study the applicability and effectiveness of geometry splitting methods coupled with collocation discretization schemes. Specifically, we consider the Generalized Schwarz Splitting (GSS) method, which is an extension of the Schwarz Alternating Method, for solving elliptic PDE problems with generalized interface conditions. The main focus is the iterative solution of the corresponding enhanced GSS linear system for a model problem. For this we carry out the spectral analysis of the enhanced block Jacobi iteration matrix. In the case of one-dimensional problems, we determine the convergence interval of one-parameter GSS and find a subinterval of it where the optimal parameter lies; moreover, we obtain sets of optimal parameters for the multi-parameter GSS case. We aslo analyze the convergence properties of the one-parameter GSS for a two-dimensional model problem.

Key words. elliptic partial differential equations, collocation methods, SOR iterative method

Introduction

An *open problem* is to find a method for the iterative solution of the discrete equations obtained from applying the collocation method based on bicubic Hermite piecewise polynomials to discretize a general second-order linear elliptic partial differential equation of the form

$$Lu \equiv au_{xx} + cu_{yy} + du_x + eu_y + fu = g, \quad (x, y) \in R, \tag{1.1}$$

subject to the boundary conditions

$$Bu \equiv \alpha u + \beta \frac{\partial u}{\partial n} = \delta, \quad (x, y) \in \partial R, \tag{1.2}$$

[*] This work was supported by AFOSR 91-F49620 and NSF grant CCR 86-19817

[†] Department of Mathematics, Purdue University, West Lafayette, IN 47907

[‡] Department of Computer Sciences, Purdue University, West Lafayette, IN 47907.

where R is a general domain while all the coefficients and the right hand sides in (1.1) and (1.2) may depend on x and y.

One of the objectives of this thesis is to analyze theoretically and experimentally iterative methods for the solution of Hermite collocation equations associated with the PDE equation (1.1) defined on a rectangular domain with Dirichlet or Neumann conditions on parts of the boundary. A "natural" ordering of the collocation equations and unknowns [22] leads to a banded coefficient matrix which is in general non-symmetric and non-diagonally dominant whose diagonal elements are almost all zero. Thus a straightforward application of the classical point iterative methods to solve these equations is not possible. These systems are currently solved by Gauss elimination with scaling and partial pivoting [9]. Some "customized" direct and iterative solvers have been developed for solving the Hermite collocation equations for special elliptic PDE operators and boundary conditions on the unit square [6], [2]. The iterative solution of the Hermite collocation equations was first addressed in [24] and [30] for the case of *interior* Hermite collocation applied on the Poisson PDE problem with Dirichlet boundary conditions defined on the unit square. The application of the iterative methods was based on a special reordering of the equations and the unknowns, which resulted to a block tridiagonal coefficient matrix. In this thesis we extend the iterative approaches proposed in [24] and [30] for a class of "general" Hermite collocation equations. These extensions are based on a new partitioning of the corresponding "interior" collocation matrix which allows us to derive analytically the eigenvalues of the corresponding block Jacobi iteration matrix and determine the optimal overrelaxation factor of the Successive Overrelaxation (SOR) iterative method [39], [41]. In addition, we improve several of Papatheodorou's theoretical results for the "interior" collocation equations [30]. In the case of a model elliptic PDE problem with uncoupled mixed boundary conditions, we derive analytic expressions for the eigenvalues of the block Jacobi iteration matrix based on a new partitioning of the interior collocation matrix, and determine the optimal overrelaxation factor for the block SOR iterative method. We present numerical results which support the theoretical analysis of the block SOR method and compare its convergence behavior with those of the block Jacobi, Gauss-Seidel and AOR used in [30]. Furthermore, we compare the time and memory complexity of the block SOR, LINPACK BAND GE, and GMRES mathematical software for solving the Hermite collocation equations obtained from the discretization of several PDE problems. The numerical results indicate that the block SOR method is the most efficient method for solving these equations.

For the case of general domains, finding methods for the iterative solution of the corresponding discrete collocation equations is still an *open problem*. In a series of papers [22, 20, 21], Houstis, Mitchell and Rice proposed three algorithms for the numerical solution of the second-order linear elliptic PDEs on general two-dimensional domains using the cubic Hermite collocation discretization method. Their software is available in the collected algorithms of the ACM. The most general of these algorithms, called GENCOL, implements the general exterior cubic Hermite collocation approach where the boundary collocation equations are coupled with the interior ones. A simplified version of the GENCOL algorithm, called INTCOL, implements the interior cubic Hermite collocation method when the boundary collocation equations are uncoupled from the interior collocation equations. The applicability of the INTCOL algorithm is limited to PDEs defined on rectangular domains. In order to address the iterative solution of collocation equations we extend the INTCOL algorithm for general rectilinear domains (by "rectilinear" we mean the boundaries are parallel to one of the axes). Throughout, we refer to it by the acronym GINCOL. Moreover, because the ordering of the unknowns and equations in the collocation discretization methods plays a vital role for the numerical solution of the linear system produced, we develop two indexing modules for the GINCOL algorithm. One is based on the finite-element ordering [43] and the other is based on the tensor-product ordering [30]. The collocation coefficient matrix based on a finite-element ordering for the GINCOL algorithm is in general non-symmetric and is not diagonally dominant; many of its diagonal entries are zero. Using the tensor-product ordering, the linear system derived by the GINCOL algorithm generates the same block structure that is produced by INTCOL. We explore the applicability and the convergence properties of the block iterative methods for GINCOL applied to model problems defined on L-shaped domains as well as on a few more general rectilinear domains. Furthermore, the tensor-product ordering was successfully applied to the discrete equations produced by GENCOL together with the SOR and CG iterative solvers. A number of experiments were carried out to study the computational behavior of these iterative schemes and to estimate the various parameters involved.

Another objective of this thesis is to study the mathematical and computational behavior of geometry splitting methods coupled with Hermite collocation discretization schemes. A well known geometry splitting methodology is the Schwarz Alternating Method (SAM). It was originally introduced in [35] over a hundred years ago to solve the Dirichlet problem for Laplace's equation on a plane domain by iterating over a sequence of Dirichlet subproblems defined on two overlapping subregions of the original domain. The coupling of these subproblems is enforced through the so called *interface conditions* defined on the subdomain boundaries in the interior of the whole domain (interfaces). The original formulation of SAM assumed Dirichlet interface conditions that depended on the solution of the neighbor subproblem(s). Its convergence properties are studied in detail in [5] and [25]. One of the early numerical formulations of SAM for elliptic boundary value problems can be found in [29]. The numerical SAM approach has recently become very popular in connection with the parallel solution of elliptic PDEs. This is primarily due to its inherent coarse grain parallel structure. In this thesis, we consider the SAM method with generalized interface conditions which are the linear combination of the solution and its normal derivative on the subdomain interfaces. Each of these conditions depends on a parameter associated with each overlapping region. This extension of SAM is called Generalized Schwarz Splitting

(GSS) [38]. The Schwarz Alternating Method has been coupled with either finite difference or finite element discretization schemes to solve elliptic boundary value problems in complex geometries by many researchers. In some special cases, the convergence properties of SAM have been investigated at a functional level. Since its introduction, the convergence properties of the GSS with finite difference discretization have appeared in many studies including [38] and [26]. To our knowledge, there are only a few researchers who have considered either SAM or GSS coupled with collocation discretization schemes. In [3] the authors apply SAM based on Legendre collocation discretization and spectral methods to solve elliptic problems and demonstrate its convergence for model problems. In [40] the formulation of SAM was considered for the Poisson equations with Dirichlet boundary conditions on an L-shaped region. Only experimental results are reported in [40]. The work in [3, 40] and our recent work in [26] and [27] has motivated us to study the convergence properties of GSS associated with the cubic Hermite collocation discretization technique [22]. The SAM approach can be formulated either on the continuous geometric and functional components of the PDE problem (referred to as the functional level formulation) or on the corresponding discrete geometric and algebraic data structures associated with the numerical method selected (referred to as the matrix equation level formulation). In this thesis, we consider the matrix formulation of SAM and GSS for elliptic PDE problems based on the Hermite collocation discretization procedure. Specifically, we derive the associated enhanced Hermite collocation matrix equation problem [38] for GSS and study its iterative solution.

This thesis consists of four chapters.

Chapter 1 presents an overview of the cubic Hermite collocation method for the second-order elliptic PDE problems. First, we briefly describe the formulations of the GENCOL, INTCOL and HERMCOL algorithms. Then we review the various proposed ordering schemes for these algorithms and the structure of the resulting systems of algebraic equations.

Chapter 2 presents the analysis of block iterative methods for the INTCOL and HERMCOL equations derived from the discretization of second-order elliptic PDEs defined on 2-D rectangular domains. First, we define two partitionings for INTCOL equations. Then, we carry out the spectral analysis of the Jacobi iteration matrix corresponding to the two partitionings individually. These results are applicable for Dirichlet model problems on the unit square. Using these results we analyze the convergence property of the block SOR method. Finally, we study the numerical behavior of several block iteration methods including optimal and adaptive SOR, Jacobi and Gauss-Seidel and verify some of the theoretical results. In addition, we compare the block optimal SOR solution, three preconditioning conjugate gradient methods based on GMRES software and the LINPACK BAND GE solver with respect to their estimated time and memory complexity for some model PDE problems.

Chapter 3 presents the extension of INTCOL method to elliptic PDEs defined on rectilinear domains. We formulate the GINCOL algorithm. Then we develop two different indexing modules based on finite-element ordering and tensor-product ordering, respectively. Finally, we apply the GINCOL algorithm to discretize some PDEs and study the computational behavior of some iterative linear solvers using tensor-product ordering.

Finally, *Chapter 4 presents the formulation and analysis of the Generalized Schwarz Splitting method based on cubic Hermite collocation approach.* We give a brief description of the GSS on a rectangle at functional and matrix levels. Then, we derive the block Jacobi iteration matrix corresponding to applying the GSS coupled with bicubic Hermite collocation discretization for the solution of the Poisson equation with Dirichlet boundary conditions on a rectangular domain split into overlapping stripes. We carry out a spectral analysis of the enhanced block Jacobi iteration matrix for one- and two-dimensional model problems. For one-dimensional problem, we determine the domain of convergence and find a subinterval of it in which the optimal parameter for the one-parameter GSS case lies; moreover, we obtain sets of optimal parameters for the multi-parameter GSS case. Finally, we present a number of numerical examples in the one- and two-dimensional spaces that verify the theoretical results. In addition, we compare the convergence rates of the SAM and GSS methods with minimum and maximum overlap and draw several conclusions.

1. OVERVIEW OF THE CUBIC HERMITE COLLOCATION METHOD

In a series of papers Houstis *et al* [18, 17, 19, 23, 7] have studied the mathematical and computational behavior of the collocation method based on C^1 piecewise polynomials for the numerical solution of the general second-order linear elliptic PDEs. The results indicate that these type of finite element techniques are efficient numerical solvers for such mathematical models. Moreover, in [22, 20, 21] Houstis, Mitchell, and Rice proposed three algorithms for the numerical solution of the second-order linear elliptic PDEs on general two-dimensional domains using the cubic Hermite collocation discretization method. Their software is available in the collected algorithms of the ACM. The most general of the algorithms above, called GENCOL, implements the general exterior cubic Hermite collocation approach where the boundary collocation equations are coupled with the interior ones. In the case of rectangular domains, GENCOL can be considerably simplified. This implementation is referred to throughout as HERMCOL which can be simplified further by eliminating *a priori* some of the boundary degrees of freedom (dofs). This approach is called *interior collocation* and it has been implemented by the INTCOL algorithm. The purpose of this chapter is to present the general formulation of the cubic Hermite collocation discretization approach and a brief description of the three algorithms based on the material in [22, 20, 21]. Moreover, because the ordering of the unknowns and equations in the collocation discretization methods plays a vital role for the numerical solution of the linear algebraic equations produced, we also describe the various proposed indexing schemes for these systems and discuss the sparse structure of them together with the various parameters involved. In this presentation we introduce most of the notations that are used in the subsequent chapters.

This chapter is organized as follows. In Section 1.1 we describe the idea of the cubic Hermite collocation discretization procedure. Section 1.2 presents the various formulations of this collocation method for general and rectangular PDE domains and different type boundary conditions. In Section 1.3 we review the various proposed ordering schemes of the cubic Hermite collocation discretization equations and the structure of the resulting systems of algebraic equations.

1.1 The Cubic Hermite Collocation Method Suppose we are given the second-order linear elliptic PDE

$$\begin{array}{ll} Lu \equiv au_{xx} + cu_{yy} + du_x + eu_y + fu = g & \text{in } \Omega \\ Bu \equiv \alpha u + \beta \frac{\partial u}{\partial n} = \delta & \text{on } \partial\Omega \end{array}$$

where Ω is a bounded region in the k-dimensional space and $\partial\Omega$ is the boundary of Ω. The method of collocation consists of finding a function u_h in a finite dimensional approximate solution subspace of the space of square integrable functions on Ω. The function u_h is chosen so that $L(u_h) = g$ and $B(u_h) = \delta$ are satisfied exactly at certain interior and boundary points, respectively. These points are called collocation points. There are many ways to select the approximate solution subspace and the collocation points. Throughout this thesis we use the subspace of cubic Hermite piecewise polynomials which defines the the cubic Hermite collocation method. This method has been shown to be highly accurate for some second-order elliptic PDE problems (see [31] and [32]). For brevity in the sequel, when we refer to the collocation method without any further explanation, we mean the cubic Hermite collocation method.

The finite-element mesh Ω_h is a set of intervals, rectangles and rectangular parallelepiped regions for 1-D, 2-D and 3-D problems, respectively. The exact definition of Ω_h is given in the next section. The approximate solution u_h is defined on each mesh element in terms of one-dimensional local basis functions ϕ_1, ϕ_2, ϕ_3 and ϕ_4 defined on the interval (t_0, t_1) as follows:

$$\begin{array}{l}\phi_1(t) := (1 - \frac{t-t_0}{t_1-t_0})^2(1 + 2\frac{t-t_0}{t_1-t_0}),\ \phi_2(t) := (t - t_0)(1 - \frac{t-t_0}{t_1-t_0})^2, \\ \phi_3(t) := (1 + \frac{t-t_1}{t_1-t_0})^2(1 - 2\frac{t-t_1}{t_1-t_0}),\ \phi_4(t) := (t - t_1)(1 + \frac{t-t_1}{t_1-t_0})^2.\end{array}$$

The corresponding expressions for u_h are

$$\begin{array}{rcll} u_h(x) & = & \sum_{i=1}^4 \rho_i\phi_i(x), & for\ \text{1-D elements}, \\ u_h(x,y) & = & \sum_{i,j=1}^4 \rho_{ij}\phi_i(x)\phi_j(y), & for\ \text{2-D elements}, \\ u_h(x,y,z) & = & \sum_{i,j,z=1}^4 \rho_{ijk}\phi_i(x)\phi_j(y)\phi_k(z), & for\ \text{3-D elements}. \end{array}$$

From the definition of the basis functions it is clear that there are 2, 4 and 8 unknowns associated with each node for the 1-D, 2-D and 3-D cases, respectively. Furthermore, one can easily show that the values of the unknown ρ's coincide with the values of the approximate solution and its derivatives at the nodes. For example, let $(\rho_1, \rho_2, \rho_3, \rho_4)$ be the four unknowns associated with a node q on a 2-D domain, then

$$\rho_1 = u_h(q),\ \rho_2 = \frac{\partial u_h}{\partial y}(q),\ \rho_3 = \frac{\partial u_h}{\partial x}(q),\ \rho_4 = \frac{\partial^2 u_h}{\partial x \partial y}(q).$$

From the definition of the basis functions, we can easily see that the second derivative of u_h is not continuous at the element boundaries. On the other hand, using Gaussian quadrature theory [31], higher accuracy is obtained if the interior collocation points are located at the Gaussian points of the mesh element rather than at the grid nodes. As for the placement of the boundary collocation points, we follow the scheme suggested in [22]. One of the restrictions is that the number of these points must be equal to the difference of the dimension of the approximate solution subspace and the number of interior collocation points.

1.2 Formulation of Hermite Collocation Methods

1.2.1 GENCOL: Collocation Method for General 2-D Domains

The procedure of solving a PDE problem by the general collocation method can be roughly broken into the five steps indicated below (see [22]):

(1) define the PDE problem,
(2) place a rectangular grid over the domain of definition,
(3) generate the finite-element mesh,
(4) locate the collocation points and form the linear system,
(5) solve the linear system.

Steps (3) and (4) are the ones that constitute the core of the general collocation method. A detailed description of these two steps follows.

First we overlay the domain Ω by a rectangular grid G and identify the rectangular elements of G that are interior or exterior to $\partial\Omega$ or that intersect $\partial\Omega$. The latter ones are called boundary elements. It might happen that the intersection of certain boundary elements with Ω is very small. Their inclusion as element of the finite-element mesh Ω_h will not only enlarge the linear system to be solved but may, in some extreme cases, also cause numerical instability in its solution. It is thus natural to discard those boundary elements which may cause trouble. We define the finite-element mesh Ω_h as the union of the interior elements and those boundary elements e_b for which the ratio of the area of $e_b \cap \Omega$ over the area of e_b is greater than a certain amount called $DSCARE$. The portions of $\partial\Omega$ in the discarded elements are either allocated to neighboring elements or ignored. This is controlled by a logical variable called $GIVOPT$ ($GIVOPT = .TRUE.$ means allocate to neighboring elements). Note that by using this "discarding" procedure some elements may change from boundary to exterior or from interior to boundary. To assure the implementation of this procedure, some assumptions must be satisfied (see [22]):

- The boundary $\partial\Omega$ of Ω, consisting of at least two pieces, is given in a parameterized form in a clockwise manner.
- A boundary element does not contain a whole boundary piece of Ω, and there are at most two boundary pieces in it.
- The sides of a boundary element which are treated as pieces of the boundary of Ω_h must be adjacent and the number of them is at most three.
- If a boundary element is discarded, then no more than two of its neighboring elements can be interior elements.
- The boundary does not enter an element more than once, except when it leaves the element and reenters it along the same element edge. Further the neighboring element to this edge is discarded.

The above assumptions are usually satisfied for a reasonably fine mesh. Below we present a code outline for the above procedure ([22]). For this a rectangular element of G is identified by the indices (IX, JY) of its lower left corner grid point, where $1 \leq IX \leq$ *number of x-grid lines* and $1 \leq JY \leq$ *number of y-grid lines.*

```
LOOP: FOR EACH BOUNDARY POINT $B_l$ DO :
   IF THE BOUNDARY LEAVES AN ELEMENT AND ENTERS
      A NEW ELEMENT $(IX, JY)$ AT THIS POINT
      THEN SAVE THE BOUNDARY POINT INDICES FOR
        THE NEW ELEMENT AS
        $ELTYPE(IX, JY) = IENTER + 1000 \times IEXIT$
        WHERE $IENTER$ AND $IEXIT$ ARE THE INDICES OF
        THE BOUNDARY POINTS WHERE THE BOUNDARY
        ENTERS AND EXITS THE ELEMENT $(IX, JY)$
   ENDIF
ENDLOOP ;

LOOP: FOR EACH ELEMENT $(IX, JY)$ OF $G$ DO :
CASE TYPE OF ELEMENT $(IX, JY)$
   EXTERIOR: ELTYPE$(IX, JY) := -1$ /* do not use element */
   INTERIOR: ELTYPE$(IX, JY) := 0$ /* use element */
   BOUNDARY:
      IF $\frac{AREA\ OF\ ELEMENT\ INTERSECTION}{AREA\ OF\ ELEMENT} < DSCARE$
         THEN ELTYPE$(IX, JY) := -$ELTYPE$(IX, JY)$
              /* do not use element */
         ELSE ELTYPE$(IX, JY) := (IENTER + 1000 * IEXIT)$
              /* the element is used with $ELTYPE$ unchanged */
      ENDIF
ENDCASE;
ENDLOOP;

LOOP: FOR EACH BOUNDARY SEGMENT DO :
   /* if segment is in element $(IX, JY)$ and $ELTYPE(IX, Y) < -1$
   then the boundary segment in the discarded element is assigned to
   a neighboring element */
      IF ANY NEIGHBORING ELEMENTS HAVE NO
        ASSOCIATED BOUNDARY SEGMENT
        THEN THE BOUNDARY SEGMENT IS SPLIT AMONG
        THEM UP TO TWO PIECES
      ELSEIF GIVOPT = .TRUE.
        THEN THE BOUNDARY SEGMENT IS SPLIT BETWEEN
        THE TWO ELEMENTS WHOSE ASSOCIATED
        BOUNDARY SEGMENTS ARE CONNECTED TO IT
      ENDIF
ENDLOOP
/* note : if $GIVOPT = .FALSE.$ then the piece of the boundary in the
discarded element is not used */
```

Now, we can determine the interior collocation points on $\Omega_h \cap \Omega$. We split the points into two groups. One group consists of all the sets of the four Gaussian points on the corresponding interior mesh elements. Since the four Gaussian points in a boundary mesh element e_b might not be in Ω, a mapping from e_b onto $e_b \cap \Omega$ is necessary. Thus, the other group of elements is composed of the images of the four Gaussian points of each boundary element under this mapping. The map depends on several aspects of the geometry and is too complicated to give a detailed description here (see [22]). However, the main idea is the following: First, the boundary $\partial(e_b \cap \Omega)$ is partitioned into four parts and each side of e_b is mapped by a one-to-one mapping onto one of those parts. Then, the map from e_b to $e_b \cap \Omega$ is determined by linearly blending those four maps of the boundary.

To locate the boundary collocation points, one has to compute the number of boundary points such that the total number of collocation points is equal to the number of the unknowns. Let N_v and N_e be the numbers of nodes and mesh elements, respectively, on the finite-element mesh Ω_h. Since there are four unknowns associated with each node and we set four interior collocation points on each mesh element, it follows that there are $4N_v - 4N_e$ boundary collocation points that need to be determined. On the other hand, it can be shown using the Euler-Poincaré characteristic of the regular region of a surface ([4]) that $N_e - N_s + N_v = 1 - N_h$, where N_s is the number of element sides of Ω_h and N_h is the number of holes of Ω_h. Furthermore, it is easy to find that $N_s = B_s + I_s$ and $4N_e = B_s + 2I_s$, where B_s and I_s are the numbers of element sides on ∂E and in the interior of Ω_h, respectively. A little manipulation using these relations shows that

$$4N_v - 4N_e = 2B_s + 4(1 - N_h).$$

The procedure of determining the boundary collocation points consists of two passes. The first pass is to place the collocation points on the boundary of Ω_h. The second pass is to map the boundary sides of a boundary element of Ω_h onto the boundary segment of Ω associated with this element. Then the images of the collocation points placed by the first pass are the boundary collocation points sought to generate the boundary collocation equations. A more detailed description of these two passes in code form is presented below (see [22]).

```
PASS 1: /* associate boundary collocation points (BCPS) with boundary of
finite element mesh */
```

```
  PLACE TWO BCPS ON EACH BOUNDARY SIDE OF $\Omega_h$ IN
  THE SAME CONFIGURATION AS PARAMETERS BCP1 AND
  BCP2 ARE PLACED IN THE INTERVAL (0,1)
  PLACE ONE BCP AT EACH CORNER OF $\partial\Omega \cap E$
  IF THE END OF THE LAST BOUNDARY SIDE IS A CONCAVE
    CORNER OF THE FINITE ELEMENT MESH
    THEN REPLACE THE TWO BCPS OF THE LAST
    BOUNDARY SIDE WITH ONE BCP AT THE MIDPOINT OF
    THE SIDE
  ENDIF
  IF THE BEGINNING OF THE FIRST BOUNDARY SIDE
    IS A CONCAVE CORNER OF THE FINITE ELEMENT MESH
    THEN MOVE THE TWO BCPS OF THE FIRST SIDE SO
    THAT THE FIRST BCP IS AT THE BEGINNING OF THE
    FIRST SIDE AND THE SECOND BCP IS AT THE MIDPOINT
    OF THE FIRST SIDE
  ENDIF
/* this placement is represented by values in (0,1) with 1/2 corresponding to
the corner if there are two boundary sides and 1/3 and 2/3 corresponding to the
corners if there are three boundary sides */

PASS 2 : /* mapping the BCPS from $\partial\Omega_h$ to $\partial\Omega$ */
/* this is a mapping from (0,1) to the segment of $\partial\Omega$ associated with an element
of $\Omega_h$ */
  IF THE SEGMENT OF $\partial\Omega$ IS CONTAINED IN ONE PIECE OF
    THE BOUNDARY
    THEN LINEARLY MAP (0,1) TO $(PENTER, PEXIT)$
    DETERMINE THE BCPS FROM THE PASS 1
    VALUES AND THE DEFINITION OF $\partial\Omega$
  ELSEIF THE SEGMENT OF $\partial\Omega$ IS CONTAINED IN TWO
    PIECES OF THE BOUNDARY
    THEN LINEARLY MAP (0,1/2) TO $(PENTER, B_{2,I})$ AND
    (1/2,1) TO $(B_{1,I+1}, PEXIT)$, WHERE $I$ IS THE NUMBER
    OF THE FIRST PIECE AND $B_{2,I}$, $B_{1,I+1}$ ARE FROM
    THE PARAMETRIZED FORM OF BOUNDARY PIECE.
    DETERMINE THE BCPS FROM THE PASS 1 VALUES AND
    THE DEFINITION OF $\partial\Omega$
  ELSE ERROR /* allow no more than two boundary pieces
    in a element */
  ENDIF
```

It is easy to see that the procedure above does give $2B_s+4(1-N_h)$ boundary collocation points. The user is allowed to adjust the placement of the boundary collocation points in a boundary edge by changing the two parameters $BCP1$ and $BCP2$. The default case ($BCP1 = BCP2 = 0$) selects two Gaussian points in a boundary edge.

Once the collocation points are determined, to generate the collocation equations is a simple task. The collocation equations are represented by the following arrays :

$COEF(n,l)$ = lth coefficient value of equation n
$IDCO(n,l)$ = index of the unknown associated with $COEF(n,l)$
$BBBB(n)$ = right hand side value of equation n

1.2.2 HERMCOL and INTCOL: Collocation Methods for Rectangular Domains Throughtout this subsection, the domain Ω is assumed to be rectangular and is denoted by R. In this case, the domain discretization process can be simply defined by the vectors GRIDX and GRIDY which contain values of x-grid and y-grid lines, respectively. Thus, the finite-element mesh generator process is not needed. Then, the steps of generating the collocation equations are considerably simplified. It is developed as another algorithm in [22] and is called *Hermite Collocation* (HERMCOL) in [33]. A code skeleton is :

```
LOOP OVER ELEMENTS $E$ OF $R$ DO:
   LOOP OVER INTERIOR COLLOCATION POINTS DO:
     FOR $N = NROW + 1, NROW + 4$ DO:
       GENERATE $COEF(N,*), IDCO(N,*)$ and $BBBB(N)$
   ENDLOOP
   $NROW = NROW + 4$
   IF ELEMENT IS A BOUNDARY ELEMENT
   THEN LOOP OVER $K$ BOUNDARY COLLOCATION POINTS DO:
          FOR $N = NROW + 1, NROW + K$ DO :
            GENERATE $COEF(N,*), IDCO(N,*)$ and $BBBB(N)$
          ENDLOOP
          $NROW = NROW + K$
   ENDIF
ENDLOOP
```

If the problem has *uncoupled boundary conditions*, that is, at no point are the boundary conditions mixed, i.e,

$$u \equiv \delta \;\; on \;\; \partial R_1 \subset \partial R,$$
$$\frac{\partial u}{\partial n} \equiv \delta \;\; on \; \partial R_2 = \partial R - \partial R_1 \subset \partial R,$$

then the boundary collocation equations can be solved explicitly during the discretization of the boundary conditions. Thus, the HERMCOL can be simplified and the simplified version is called *Interior Collocation* (INTCOL) [33]. It consists of two consecutive steps. The first step is implemented by two parallel asynchronous processes based on the assumption that *the boundary conditions only change type on the boundary nodes.* A code skeleton for these two processes is:

```
      /* OPERATOR DISCRETIZATION */
  LOOP OVER ALL ELEMENTS OF R DO:
    LOOP OVER INTERIOR COLLOCATION POINTS DO:
      FOR $N = NROW + 1, NROW + 4$ DO:
        GENERATE $COEF(N,*)$, $IDCO(N,*)$ and $BBBB(N)$
    ENDLOOP
  ENDLOOP
      /* BOUNDARY DISCRETIZATION */
  LOOP OVER EACH BOUNDARY PIECE:
    LOOP1 OVER EACH NODE $T_i$ OF THE BOUNDARY PIECE:
      DETERMINE THE LEFT OR RIGHT HALF-INTERVAL
      ($[T_{i-1/2}, T_i]$ OR $[T_i, T_{i+1/2}]$) WHERE THE BOUNDARY
      CONDITION IS OF THE SAME TYPE AS AT $T_i$.
      /* denote the interval by $\Delta$ and its two Gauss points by $\tau_1, \tau_2$ */
      $S = \{\tau_1, \tau_2$ AND END POINTS of $\Delta\}$;
      CASE BOUNDARY CONDITION TYPE IS:
        DIRICHLET ($U = \delta$): DETERMINE $U_x$ (OR $U_y$) AT $T_i$
          BY INTERPOLATING $\delta$ BY A CUBIC POLYNOMIAL AT
          THE POINTS S; IDENTIFY THE ACTIVE UNKNOWNS;
        NEUMANN ($\partial U/\partial N = \delta$): DETERMINE $U_{xy}$ ($= U_{yx}$) AT $T_i$
          BY INTERPOLATING $\delta$ BY A CUBIC POLYNOMIAL AT
          THE POINTS S; IDENTIFY THE ACTIVE UNKNOWNS;
      ENDCASE;
    ENDLOOP1;
  ENDLOOP;
```

Finally, the nonactive unknowns predetermined in the boundary discretization process are eliminated from equations generated in the operator discretization process, i.e., $IDCO$ and $BBBB$ are modified at this stage.

1.3 Ordering and Solution of Collocation Equations The properties of the coefficient matrix of the linear system arising from the discretization of a PDE problem by the collocation method strongly depends on the ordering of the unknowns and equations. A specific ordering may produce a linear system suitable for an iterative solver while the same iterative solver might not be applicable to the linear system obtained by another ordering. Conclusively, there are three basic approaches to the ordering of the unknowns and the equations for the collocation method. Before giving a detailed description of these three orderings, we depict the numbering of the unknowns and equations on an L-shaped domain and a rectangular domain with Dirichlet boundary conditions in Figures 1.1, 1.2 and 1.3. Collocation points are shown in Times-Bold font and their numbering indicates the ordering of the equations. The unknowns are associated with nodal points and are numbered in Times-Roman font. Those unknowns eliminated symbolically during the discretization of boundary conditions are denoted by x.

The first ordering is obtained by a natural extension of the finite-element ordering in [43] to the general domains. We call it the *finite-element ordering*. It is available for GENCOL, INTCOL and HERMCOL. More specifically, once the finite-element mesh is defined, the mesh nodes and the mesh elements are numbered in a natural way from south to north, west to east. Note that there are four unknowns associated with a mesh node in the algorithms GENCOL and HERMCOL. Thus, the unknowns are numbered in groups of four (or fewer than four for INTCOL because some unknowns are eliminated during the discretization of the boundary equations) in the order of the corresponding mesh node. The four unknowns associated with a mesh node are locally ordered so they respectively represent the values of u, u_y, u_x ans u_{xy} at the mesh node. In this ordering the collocation points are numbered element by element following the element numbering in the mesh. In the case of boundary elements, for GENCOL the interior collocation points are numbered counter-clockwise first followed by the colckwise numbering of boundary collocation points, for HERMCOL the boundary collocation points are numbered first followed by the counter-clockwise numbering of the interior collocation points. Figure 1.1 display this ordering for a finite element mesh of an L-shaped region for GENCOL and rectangular regions for INTCOL and HERMCOL.

The second ordering is called the *tensor-product ordering*. This scheme was originally defined in [30] for INTCOL and is extended to be used for HERMCOL in [27]. First, the HERMCOL unknowns are split into two sets $\{u, u_y\}$ and $\{u_x, u_{xy}\}$. Then, on each x-grid line we number the unknowns $\{u, u_y\}$ node by node (south to north) followed by the numbering of $\{u_x, u_{xy}\}$ unknowns corresponding to the nodal points of the same grid line. The HERMCOL collocation points are ordered from south to north along left edge of R, x-Gauss grid lines and right edge of R from west to east. In the case of INTCOL, we have only interior collocation points, thus they are ordered from south to north along x-Gauss grid lines corresponding to x-coordinadtes of the Gauss points. Then the numbering of the active unknowns is determined by the indices of the interior collocation points as follows. At each nodal point, the active unknowns use the same index as the nearest interior collocation points. Figure 1.2 illustrates this ordering scheme for a rectangular region.

The third one is called the *collorder ordering*, which is defined for INTCOL and HERMCOL in [8]. The idea is that the unknowns are numerbered in the same way as the *finite-element ordering*; for the numbering of collocation points, the collocation points are associated with the nearest grid point and are numbered in groups of four (or two for INTCOL collocation points on the edges of R) in the order of their corresponding grid point. The collocation points may be locally ordered in any way and some collocation points are reordered depending on the boundary conditions (the detailed description in [8]). Figure 1.3 illustrates this ordering of collocation points in a rectangular region.

The finite-element ordering of the unknowns and equations of the collocation equations usually gives a banded linear system with a large number of zero diagonal elements (see Figure 1.4(a)). If the domain Ω is rectangular, then the bandwidth is $4 \times NGRIDY + 7$ for HERMCOL and $2 \times NGRIDY + 3$ for INTCOL, respectively, where $NGRIDY$ is the number of y-grid lines. As for the general domain, the linear system becomes less regular in pattern and very little can be said about its bandwidth because it depends on both $NGRIDY$ and the shape of Ω. Sometimes, the linear system can be made with efficiency of bandedness by a widely used frontal method [43]. On the other hand, the presence of many zero diagonal elements prevents most iterative method from being applied. Thus, the most reliable and preferable way to solve the linear system corresponding to the collocation equation using finite-element ordering is Gauss elimination with scaling and partial pivoting [9].

The tensor-product ordering yields the coefficient matrix of the INTCOL or HERMCOL equations with bandwidth $4 \times NGRIDY - 2$ or $4 \times NGRIDY + 2$ individually and with a nice block structure shown in Figure 1.5. Furthermore, the coefficient matrix has non-zero diagonal elements for INTCOL (see Figure 1.4 (b)) and might have some zero-diagonal elements corresponding to uncoupled boundary conditions for HERMCOL. Thus, both direct solvers and iterative solvers can be applied for the solutions of INTCOL equations or HERMCOL equations with mixed boundary conditions using this ordering. However, direct solvers tend to require much more memory as well as time and their parallelization is difficult. It is very desirable to have a suitable iterative solvers for INTCOL and HERMCOL equations. A detailed description of the application of iterative solvers for the INTCOL and HERMCOL equations using the tensor-product ordering and a study of their convergence behavior is presented in the next chapter.

The collorder ordering produces a coefficient matrix of INTCOL (HERMCOL) equations with bandwidth $4 \times NGRIDY$ $(4 \times NGRIDY + 7)$. However, the matrix still has some zero diagonal elements corresponding to boundary conditions for the HERMCOL equations. They can be removed from the diagonal easily by a mild reordering of the unknowns associated with that boundary grid point. Thus, the usual iterative method is applicable using this ordering. Unfortunately, it diverges rapidly when directly applied. Experiments indicate that Gauss elimination without pivoting is safe for the solution of INTCOL or HERMCOL equations using this ordering.

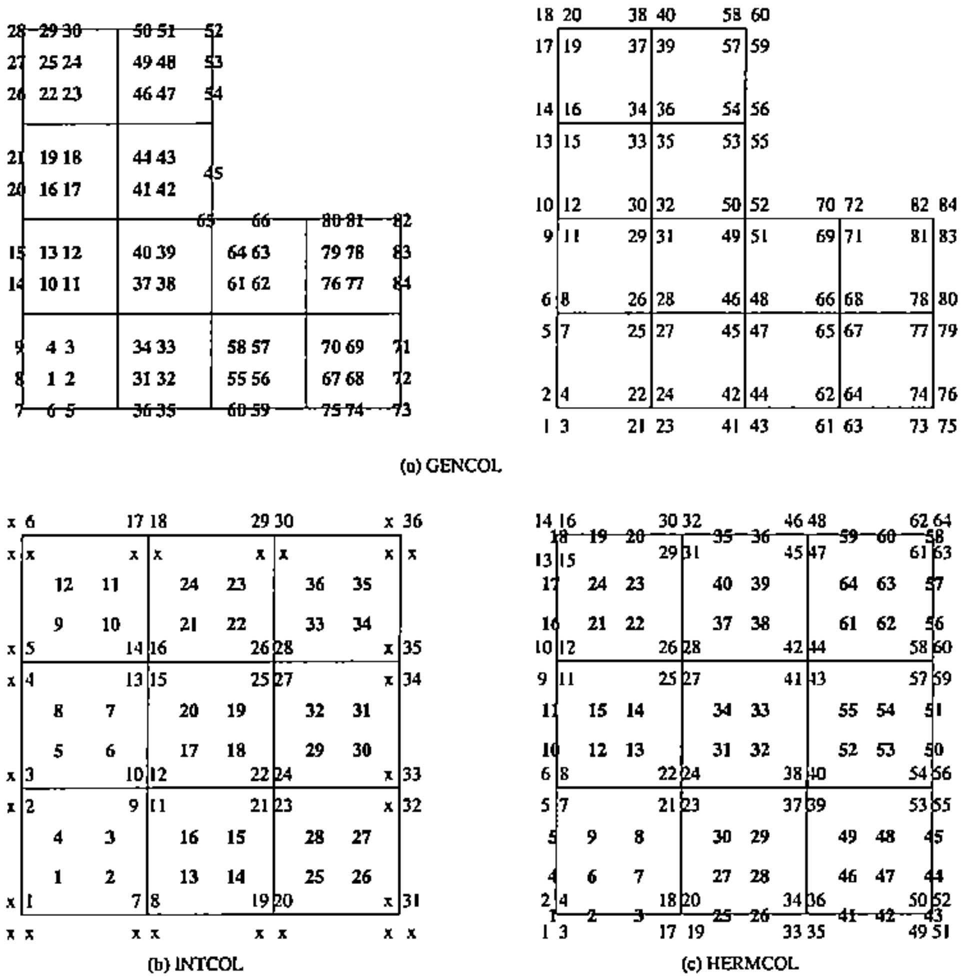


FIG. 1.1. *Finite-element orderings of the collocation points and unknowns associated with GENCOL, INTCOL and HERMCOL.*

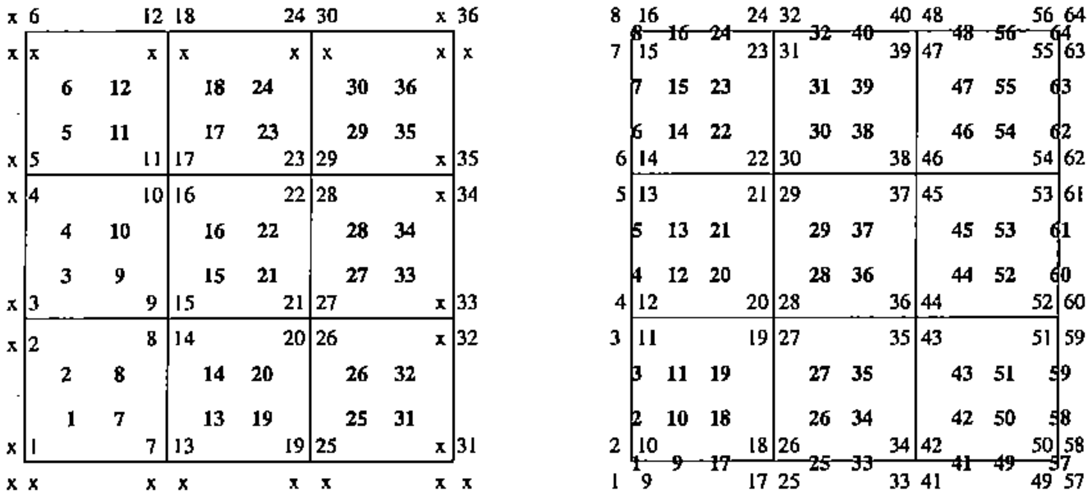


FIG. 1.2. *Tensor-product ordering of the collocation points and unknowns associated with INTCOL and HERMCOL.*

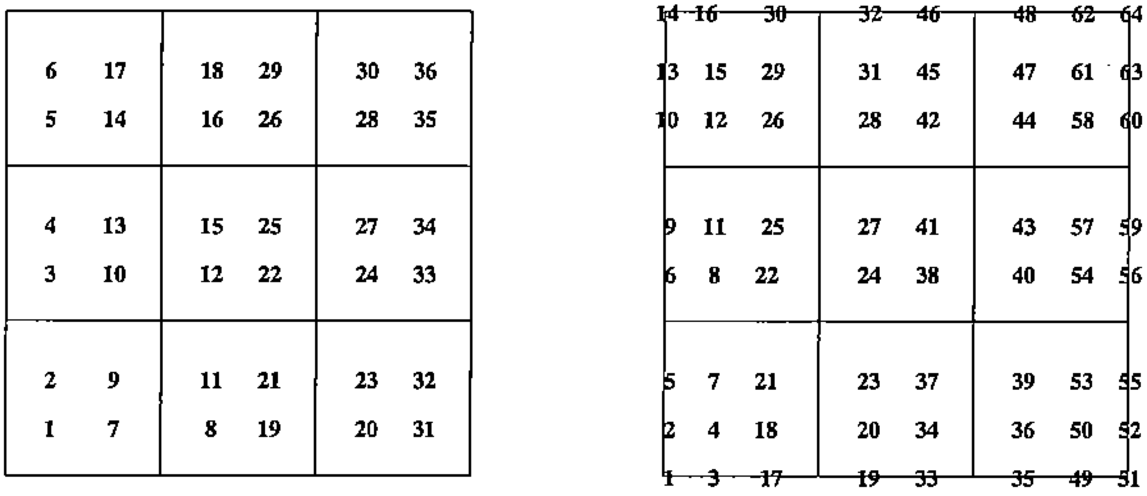


FIG. 1.3. *Collorder ordering of the collocation points associated with INTCOL and HERMCOL, respectively*

```
dxx...xxxxxx........................
xdx...xxxxxx........................
xxd...xxxxxx........................
xxx0..xxxxxx........................
.xxxd...xxxxxxxx....................
.xxxx0..xxxxxxxx....................
.xxxx.0.xxxxxxxx....................
.xxxx..0xxxxxxxx....................
...xxx..0...xxxxxx..................
...xxx...0..xxxxxx..................
...xxx....0.xxxxxx..................
...xxx.....0xxxxxx..................
......xxxxxx0.....xxxxxx............
......xxxxxx.0....xxxxxx............
......xxxxxx..0...xxxxxx............
......xxxxxx...0..xxxxxx............
........xxxxxxxx0...xxxxxxxx........
........xxxxxxxx.0..xxxxxxxx........
........xxxxxxxx..0.xxxxxxxx........
........xxxxxxxx...0xxxxxxxx........
............xxxxxx..0...xxxxxx......
............xxxxxx...0..xxxxxx......
............xxxxxx....0.xxxxxx......
............xxxxxx.....0xxxxxx......
..................xxxxxx0.....xxx...
..................xxxxxx.0....xxx...
..................xxxxxx..0...xxx...
..................xxxxxx...0..xxx...
....................xxxxxxxx0..xxxx.
....................xxxxxxxx.0.xxxx.
....................xxxxxxxx..0xxxx.
....................xxxxxxxx...dxxx.
........................xxxxxx..0xxx
........................xxxxxx...dxx
........................xxxxxx...xdx
........................xxxxxx...xxd
                (a)
```

```
dxx...xxx...xxx.....................
xdx...xxx...xxx.....................
.xdxx..xxxx..xxxx...................
.xxdx..xxxx..xxxx...................
...xdx...xxx...xxx..................
...xxd...xxx...xxx..................
xxx...dxx...xxx.....................
xxx...xdx...xxx.....................
.xxxx..xdxx..xxxx...................
.xxxx..xxdx..xxxx...................
...xxx...xdx...xxx..................
...xxx...xxd...xxx..................
......xxx...dxx...xxx...xxx.........
......xxx...xdx...xxx...xxx.........
.......xxxx..xdxx..xxxx..xxxx.......
.......xxxx..xxdx..xxxx..xxxx.......
.........xxx...xdx...xxx...xxx......
.........xxx...xxd...xxx...xxx......
......xxx...xxx...dxx...xxx.........
......xxx...xxx...xdx...xxx.........
.......xxxx..xxxx..xdxx..xxxx.......
.......xxxx..xxxx..xxdx..xxxx.......
.........xxx...xxx...xdx...xxx......
.........xxx...xxx...xxd...xxx......
..................xxx...dxx...xxx...
..................xxx...xdx...xxx...
...................xxxx..xdxx..xxxx.
...................xxxx..xxdx..xxxx.
.....................xxx...xdx...xxx
.....................xxx...xxd...xxx
..................xxx...xxx...dxx...
..................xxx...xxx...xdx...
...................xxxx..xxxx..xdxx.
...................xxxx..xxxx..xxdx.
.....................xxx...xxx...xdx
.....................xxx...xxx...xxd
                (b)
```

FIG. 1.4. *(a) and (b) display the structure of the coefficient matrix of the INTCOL linear system for the 3 × 3 mesh using finite-element ordering and tensor-product ordering, respectively, where d denotes a nonzero diagonal element.*

$$\begin{bmatrix}
\mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & & \\
\mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & & \\
 & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & \\
 & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & \\
 & & & & \ddots & & & & \\
 & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \\
 & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \\
 & & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} \\
 & & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X}
\end{bmatrix}
\qquad
\begin{bmatrix}
\mathrm{X} & \mathrm{X} & & & & & & & & & \\
\mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & & & \\
\mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & & & \\
 & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & \\
 & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & \\
 & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} & & \\
 & & & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\
 & & & & & & & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\
 & & & & & & & & & \mathrm{X} & \mathrm{X}
\end{bmatrix}$$

(*a*) INTCOL (*b*) HERMCOL

Fig. 1.5. *The structure of INTCOL and HERMCOL equations assuming tensor-product ordering.*

2. BLOCK ITERATIVE METHODS FOR CUBIC HERMITE COLLOCATION EQUATIONS

Collocation methods based on bicubic Hermite piecewise polynomials have been proven effective techniques for solving general second order linear elliptic PDEs with mixed boundary conditions [22]. From Chapter 1, we know that using finite-element ordering the corresponding system of discrete collocation equations is in general non-symmetric and non-diagonally dominant. Their iterative solution is not known and they are currently solved using Gauss elimination with scaling and partial pivoting. Using collorder ordering the Point iterative methods like those in ITPACK [36] do not converge even for the collocation equations obtained from the discretization of model PDE problems. In this chapter we develop and analyze block iterative methods for the INTCOL and HERMCOL equations using tensor-product ordering. Papatheodorou was first to determine the exact parameters of AOR type iterative methods for the case of INTCOL equations associated with a model problem in [30]. We generalize the results of Papatheodorou for the INTCOL equations and extend them for a specific class of HERMCOL equations. A number of numerical results are presented to verify the theoretical ones.

The organization of this chapter is as follows. In Section 2.1, we define two partitionings for INTCOL equations and introduce a notation for defining the various block partitionings of collocation coefficient matrices used in the spectral analysis of the Jacobi iteration matrix. In Sections 2.2 and 2.3, we carry out the spectral analysis of the Jacobi iteration matrix corresponding to the partitionings P_I and P_{II} respectively. These results are applicable for Dirichlet model problems on the unit square. In Section 2.4, we use the results in Sections 2.2 and 2.3 to study the convergence analysis of the block SOR method.Moreover, we make some comparisons concerning the two block Jacobi iteration matrices and develop the corresponding optimal block SOR iterative method. Finally, in Section 2.5 we study the numerical behavior of block iterative methods including optimal and adaptive SOR, Jacobi and Gauss-Seidel and verify some of the theoretical results obtained in this chapter. In addition, we compare the block optimal SOR solution, three preconditioning conjugate gradient methods based on GMRES software and the LINPACK BAND GE solver with respect to their estimated time and memory complexity for two model PDE problems with several types of boundary conditions and a general PDE problem. The numerical results indicate that the block SOR method developed is an efficient alternative for solving the Hermite collocation equations obtained from the dicretization of general elliptic PDEs defined on rectangularregions and subject to uncoupled mixed boundary conditions.

2.1 Preliminaries Throughout this chapter, the domain R is a rectangle and discretized by $n+1$ x-grid lines and $m+1$ y-grid lines. We will focus on iterative methods for the INTCOL and HERMCOL equations using tensor-product ordering. Under the assumptions above, the x in Figure 1.5(a) denotes a $2m \times 2m$ matrix while the coefficient matrix of the INTCOL equations is a $4mn \times 4mn$ matrix. For this matrix, we consider two different partitionings for it

$$P_I = \left[\begin{array}{ccccccccc}
\text{x} & \text{x}\,| & \text{x} & & & & & & \\
\text{x} & \text{x}\,| & \text{x} & & & & & & \\
\hline
 & \text{x}\,| & \text{x} & \text{x}\,| & \text{x} & & & & \\
 & \text{x}\,| & \text{x} & \text{x}\,| & \text{x} & & & & \\
 & & \ddots & & \ddots & & \ddots & & \\
 & & & & \text{x}\,| & \text{x} & \text{x}\,| & \text{x} & \\
 & & & & \text{x}\,| & \text{x} & \text{x}\,| & \text{x} & \\
\hline
 & & & & & & \text{x}\,| & \text{x} & \text{x} \\
 & & & & & & \text{x}\,| & \text{x} & \text{x}
\end{array}\right], P_{II} = \left[\begin{array}{ccccccccc}
\text{x}\,| & \text{x} & \text{x} & & & & & & \\
\hline
\text{x}\,| & \text{x} & \text{x} & & & & & & \\
 & |\,\text{x} & \text{x}\,| & \text{x} & \text{x} & & & & \\
\hline
 & \text{x} & \text{x}\,| & \text{x} & \text{x}\,| & & & & \\
 & & \ddots & & \ddots & & \ddots & & \\
 & & & & |\,\text{x} & \text{x}\,| & \text{x} & \text{x} & \\
\hline
 & & & & \text{x} & \text{x}\,| & \text{x} & \text{x}\,| & \\
 & & & & & & |\,\text{x} & \text{x}\,| & \text{x} \\
\hline
 & & & & & & \text{x} & \text{x}\,| & \text{x}
\end{array}\right].$$

There is no surprise that we consider the partitioning P_{II} here, since applying P_I to the coefficient matrix of HERMCOL equations and using the fact that the INTCOL coefficient matrix is a principle submatrix of the HERMCOL coefficient matrix we end up with the partitioning P_{II} for the INTCOL coefficient matrix. Apparently, both partitionings make the coefficient matrix be a block 2-cyclic consistently ordered matrix [39]. This property motivates us to explore the use of block SOR iterative methods to solve the corresponding linear system.

Before we proceed, some notations for partitioning matrices are introduced. First, we introduce the block form

$$[A|B] = \left[\begin{array}{cc|cc} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & a_{22} & b_{21} & b_{22} \end{array}\right]$$

which we subsequently use to construct the following $(2n) \times (2n)$ matrix

$$[A|B]_{\otimes(2n)} = \begin{bmatrix} a & B & & & \\ & A & B & & \\ & & \ddots & & \\ & & & A & B \\ & & & & A \;\; b \end{bmatrix}, \quad a = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_{12} \\ b_{22} \end{bmatrix}.$$

Note that if all a_{ij} and b_{ij} are $2m \times 2m$ matrices, then A and B are matrices of 2×2 block form and of order $4m$. So the matrix $[A|B]_{\otimes(2n)}$ is of order $4mn$.

2.2 Spectral Analysis of the Jacobi Matrix Corresponding to P_I We consider the INTCOL coefficient matrix for the case of a Poisson equation on a rectangle with Dirichlet boundary conditions and a uniform grid. In this case the collocation coefficient matrix is of the form

$$A = \left[\begin{array}{cc|cc} A_1 & A_2 & A_3 & -A_4 \\ A_3 & A_4 & A_1 & -A_2 \end{array}\right]_{\otimes(2n)} \tag{2.1}$$

with each A_i being of order $2m$. Note that the partitioning P_I allows us to write A as

$$A = \begin{bmatrix} D_1 & -U_1 & & & \\ -L_1 & D_1 & -U_1 & & \\ & \ddots & \ddots & \ddots & \\ & & -L_1 & D_1 & -U_1 \\ & & & -L_1 & \bar{D}_1 \end{bmatrix} \tag{2.2}$$

where

$$D_1 = \begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix}, \ \bar{D}_1 = \begin{bmatrix} A_2 & -A_4 \\ A_4 & -A_2 \end{bmatrix}, \ -L_1 = \begin{bmatrix} 0 & A_1 \\ 0 & A_3 \end{bmatrix}, \ -U_1 = \begin{bmatrix} -A_4 & 0 \\ -A_2 & 0 \end{bmatrix}. \tag{2.3}$$

In the subsequent analysis we assume that D_1, $\bar{D}_1$ are nonsingular. Furthermore, we introduce the matrices

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix}^{-1} \begin{bmatrix} -A_4 & A_1 \\ -A_2 & A_3 \end{bmatrix}, \tag{2.4}$$

$$\begin{bmatrix} R_{31} \\ R_{32} \end{bmatrix} = \begin{bmatrix} A_2 & -A_4 \\ A_4 & -A_2 \end{bmatrix}^{-1} \begin{bmatrix} A_1 \\ A_3 \end{bmatrix}, \tag{2.5}$$

and note that

$$\begin{bmatrix} -A_4 & A_1 \\ -A_2 & A_3 \end{bmatrix} = \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix} \begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}. \tag{2.6}$$

From the relations (2.4) and (2.6) we obtain

$$R = \begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix} \begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

and

$$R^{-1} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} R \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

Consequently, we have

$$R^{-1} = \begin{bmatrix} R_{11} & -R_{12} \\ -R_{21} & R_{22} \end{bmatrix}. \tag{2.7}$$

As we have observed in some applications R_{21} is invertible. Then it follows from (2.7) that R_{11} $(= R_{21}R_{22}R_{21}^{-1})$ is similar to R_{22} and we prove the following lemma.

LEMMA 2.1. *If R_{21} is nonsingular, then $R_{31} = -R_{21}^{-1}$.*

Proof. First we observe that equation (2.4) implies

$$\begin{bmatrix} A_2 & A_3 \\ A_4 & A_1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} -A_4 & A_1 \\ -A_2 & A_3 \end{bmatrix}.$$

From equation (2.5) we have that $A_2R_{31} - A_4R_{32} = A_1$ and $A_4R_{31} - A_2R_{32} = A_3$. If we use the expressions for A_1 and A_3 in the equation above, we obtain

$$\begin{bmatrix} A_2 & -A_4 \\ A_4 & -A_2 \end{bmatrix} \begin{bmatrix} I & -R_{32} \\ 0 & -R_{31} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} A_2 & -A_4 \\ A_4 & -A_2 \end{bmatrix} \begin{bmatrix} 0 & R_{31} \\ I & R_{32} \end{bmatrix}.$$

Since $\bar{D}_1$ is invertible, the equation above can be simplified as follows

$$\begin{bmatrix} I & -R_{32} \\ 0 & -R_{31} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} 0 & R_{31} \\ I & R_{32} \end{bmatrix}.$$

Comparing both sides, we readily obtain that $-R_{31}R_{21} = I$. So, our assertion is established. □

The block partitioning of A in (2.2) corresponds to the splitting $A = D - L - U$, where $D = \mathrm{diag}(D_1, \ldots, D_1, \bar{D}_1)$ and where L and U are strictly lower and upper triangular matrices, respectively. Let $J = D^{-1}(L+U)$ be the block Jacobi iteration matrix associated with this partitioning. An easy calculation using equations (2.3), (2.4) and (2.5) shows that

$$J = \left[\begin{array}{cc|cc|cc|c|cc|cc|cc}
0 & 0 & R_{11} & 0 & & & & & & & & & \\
0 & 0 & R_{21} & 0 & & & & & & & & & \\
\hline
0 & R_{12} & 0 & 0 & R_{11} & 0 & & & & & & & \\
0 & R_{22} & 0 & 0 & R_{21} & 0 & & & & & & & \\
\hline
 & & & \ddots & & \ddots & & \ddots & & & & & \\
\hline
 & & & & & & & 0 & R_{12} & 0 & 0 & R_{11} & 0 \\
 & & & & & & & 0 & R_{22} & 0 & 0 & R_{21} & 0 \\
\hline
 & & & & & & & & & 0 & R_{31} & 0 & 0 \\
 & & & & & & & & & 0 & R_{32} & 0 & 0
\end{array}\right]. \qquad (2.8)$$

Due to the presence of the zeros in the first and last block columns of J and Lemma 2.1, it is easy to show that the spectrum $\sigma(J)$ of J satisfies $\sigma(J) = \sigma(J_1) \cup \{0\}$, where J_1 is given by

$$J_1 = \begin{bmatrix}
0 & R_{21} & 0 & & & & & & \\
R_{12} & 0 & 0 & R_{11} & & & & & \\
R_{22} & 0 & 0 & R_{21} & 0 & & & & \\
 & 0 & R_{12} & 0 & 0 & R_{11} & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & & 0 & R_{12} & 0 & 0 & R_{11} \\
 & & & & & R_{22} & 0 & 0 & R_{21} \\
 & & & & & & 0 & -R_{21}^{-1} & 0
\end{bmatrix} \qquad (2.9)$$

Note that J_1 has only $(n-1)$ diagonal blocks. Using (2.7) we obtain that

$$J_1^{-1} = \begin{bmatrix}
0 & -R_{21} & R_{22} & & & & & & \\
R_{21}^{-1} & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & -R_{21} & R_{22} & & & & \\
 & R_{11} & -R_{21} & 0 & 0 & 0 & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & & R_{11} & -R_{21} & 0 & 0 & 0 \\
 & & & & & 0 & 0 & 0 & -R_{21} \\
 & & & & & & R_{11} & -R_{21} & 0
\end{bmatrix}. \qquad (2.10)$$

Then, from (2.9) and (2.10), we have that

$$J_1 + J_1^{-1} = \begin{bmatrix} 0 & 0 & R_{22} & & & & & & \\ R^{*} & 0 & 0 & R_{11} & & & & & \\ R_{22} & 0 & 0 & 0 & R_{22} & & & & \\ & R_{11} & 0 & 0 & 0 & R_{11} & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & R_{11} & 0 & 0 & 0 & R_{11} \\ & & & & & R_{22} & 0 & 0 & 0 \\ & & & & & & R_{11} & -R^{*} & 0 \end{bmatrix} \tag{2.11}$$

where $R^{*} = R_{12} + R_{21}^{-1}$. From the directed graph (in Figure 2.1) associated with $J_1 + J_1^{-1}$,

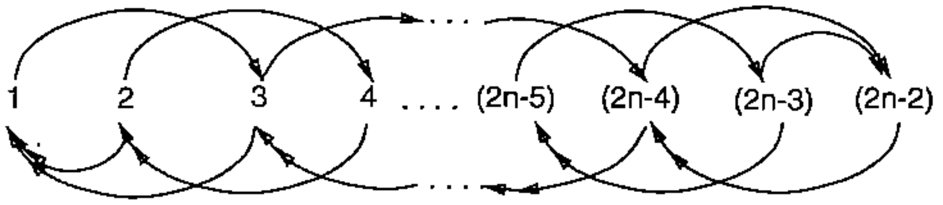


FIG. 2.1. *The directed graph corresponds to the matrix $J_1 + J_1^{-1}$*

it is readily seen that through a similarity permutation transformation that $J_1 + J_1^{-1}$ is transformed to

$$\bar{J} = \left[\begin{array}{ccccc|ccccc} 0 & R_{22} & & & & & & & & \\ R_{22} & 0 & R_{22} & & & & & & & \\ & \ddots & \ddots & \ddots & & & & 0 & & \\ & & R_{22} & 0 & R_{22} & & & & & \\ & & & R_{22} & 0 & & & & & \\ \hline R^{*} & & & & & 0 & R_{11} & & & \\ & 0 & & & & R_{11} & 0 & R_{11} & & \\ & & \ddots & & & & \ddots & \ddots & \ddots & \\ & & & 0 & & & & R_{11} & 0 & R_{11} \\ & & & & -R^{*} & & & & R_{11} & 0 \end{array}\right]. \tag{2.12}$$

Let

$$K = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{bmatrix}$$

be a square matrix of order $(n-1)$. Note that from (2.7) R_{22} is similar to R_{11}. So we have that $\sigma(\bar{J}) = \sigma(G)$ where $G = K \otimes R_{22}$. The symbol $\otimes$ denotes Kronecker product (cf. [16] and also [28] where tensor products were used for the first time in connection with discretized PDE problems). Some of its properties used here are

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \text{ and } (A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.$$

(For the first property to hold it is assumed that the matrix products AC and BD are well defined while for the second one that A and B are square nonsingular matrices.) We know that there always exist nonsingular matrices X and Y such that $KX = XD_K$ and $R_{22}Y = YJ_R$, respectively, where $D_K = \text{diag}(2\cos\frac{\pi}{n}, \cdots, 2\cos\frac{(n-1)\pi}{n})$ and J_R is the Jordan canonical form of R_{22}. It then follows that $G(X \otimes Y) = (X \otimes Y)\,(D_K \otimes J_R)$. We see that $D_K \otimes J_R$ is an upper triangular matrix and the nonsingularity of X and Y implies that $X \otimes Y$ is nonsingular. So we conclude that

$$\sigma(G) = \cup_{k=1}^{n-1}\{2\rho\cos\frac{k\pi}{n}|\rho \in \sigma(R_{22})\}.$$

Our discussion is summarized in the following theorem.

THEOREM 2.1. *Let J be the block Jacobi iteration matrix corresponding to (2.1) based on the partitioning P_I and assume the relations (2.3), (2.4) and (2.5) hold. Then the spectrum of J is given by the following relation*

$$\sigma(J) = \{0\} \cup_{k=1}^{n-1} \{\mu|\mu + \frac{1}{\mu} = 2\rho\cos\frac{k\pi}{n}, \quad \rho \in \sigma(R_{22})\}. \tag{2.13}$$

As a direct consequence of this theorem we can make the following observations:

Remark 1: Zero is an eigenvalue of J of multiplicity $4m$.

Remark 2: The corresponding result in [30] can be obtained as a special case of the theorem above.

For the justification of Remark 2 we denote by $\bar{R}$ the corresponding matrix R in [30] and assume that 2^l is the order of J. Then the corresponding result in [30] can be stated as follows : For every $\mu \in \sigma(J)$, if $\mu \neq 0$ then $\mu + \frac{1}{\mu} = \frac{2}{\rho}\cos\theta$, where $\rho \in \sigma(\bar{R}_{11})$ and $\theta = \frac{(2m-1)\pi}{2^k}$, $m = 1, 2, \ldots, 2^k$, $k = 1, 2, \ldots, l$. If we set $n = 2^l$ in Theorem 2.1 then we can easily show that $\bar{R}_{11}R_{22} = -I$ and $\{\frac{k\pi}{n}|k = 1, 2, \ldots, (n-1)\} = \{\frac{(2m-1)\pi}{2^k}|m = 1, 2, \ldots, 2^k, k = 1, 2, \ldots, l\}$. This implies that the corresponding result in [30] is a special case of the theorem above.

2.3 Spectral Analysis of the Jacobi Matrix Corresponding to P_{II} First, we apply the block partitioning P_{II} to the interior collocation matrix (2.1) and consider the corresponding splitting $A = D - L - U$. If we assume that A_1 and A_2 of (2.1) are nonsingular then D is invertible and the Jacobi matrix associated with the above splitting is $J = D^{-1}(L + U)$. Further, we consider the matrix $J' = (L + U)D^{-1}$. It is clear that the spectra of J and J' are the same, that is $\sigma(J) = \sigma(J')$. Since J' is much easier to study, we turn our attention to $\sigma(J')$. The block partitioning and the definition of J' imply that

$$J' = \left[\begin{array}{cc|cc|cc|cc}
0 & P & Q & & & & & \\
P-Q & 0 & 0 & & & & & \\ \hline
 & 0 & 0 & P & Q & & & \\
 & Q & P & 0 & 0 & & & \\ \hline
 & & & & \ddots & & & \\ \hline
 & & & 0 & 0 & P & Q & \\
 & & & Q & P & 0 & 0 & \\ \hline
 & & & & & 0 & 0 & P-Q \\
 & & & & & Q & P & 0
\end{array}\right] \tag{2.14}$$

where $P = -\frac{1}{2}(A_3A_1^{-1} + A_4A_2^{-1})$, $Q = -\frac{1}{2}(A_3A_1^{-1} - A_4A_2^{-1})$. Since P and Q are $2m \times 2m$ matrices, it is not an easy task to find $\sigma(J')$ directly. Instead, we determine $\sigma(J')$ when P and Q are real scalars and use this result to find $\sigma(J')$ in the general case.

LEMMA 2.2. *If P and Q are real scalars, then the eigenvalues μ of J' in (2.14) are either $\mu = \pm(P - Q)$ or they satisfy the equation $\mu^2 - 2Q\mu\cos\theta + Q^2 - P^2 = 0$, where $\theta = \frac{k\pi}{n}$, $k = 1, 2, \ldots, (n-1)$.*

Proof. This proof is based on the analysis in [10, pp. 218-230] which has been successfully used in [37] and [26]. For this reason we keep the notation established in [10]. For the sake of convenience, we assume that $PQ(P \pm Q) \neq 0$. However, our analysis does essentially carry over to the more general case. The problem of determining the eigenvalues and eigenvectors of J' is equivalent to solving the boundary value problem of the matrix difference equation

$$\begin{cases} B_0Z_{k-1} + (B_1 - \mu I)Z_k + B_2Z_{k+1} = 0, \quad k = 1, 2, \ldots, n \\ z_{2,0} = -z_{1,1}, \quad z_{1,n+1} = -z_{2,n} \end{cases}$$
$$B_0 = \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}, \; Z_k = \begin{bmatrix} z_{1,k} \\ z_{2,k} \end{bmatrix}, \tag{2.15}$$

where μ is an eigenvalue of J'. This can be solved by the nonmonic matrix polynomial theory. The nonmonic matrix polynomial which corresponds to (2.15) is given by

$$L(\lambda) := B_2\lambda^2 + (B_1 - \mu I)\lambda + B_0 = \begin{bmatrix} Q\lambda^2 - \mu\lambda & P\lambda \\ P\lambda & Q - \mu\lambda \end{bmatrix}. \tag{2.16}$$

From Theorem 8.3 in [10] we know that the general solution of (2.15) is given by

$$Z_k = X_F J_F^k g, \quad k = 0, 1, 2, \ldots \tag{2.17}$$

where (X_F, J_F) (cf. [10, Chs 1, 7]) is a Jordan pair of the matrix polynomial $\mathrm{L}(\lambda)$, $g \in \mathbb{C}^n$, and n is the degree of $\det(\mathrm{L}(\lambda))$. From (2.16) it is readily obtained that

$$\det(L(\lambda)) = -\lambda(Q\mu\lambda^2 - (\mu^2 + Q^2 - P^2)\lambda + Q\mu). \tag{2.18}$$

We distinguish two cases according to whether μ is zero or not.

Case 1 : $\mu = 0$. Then 0 is a double eigenvalue of $\mathrm{L}(\lambda)$ and $x_1 = [1, 0]^T$ is the corresponding eigenvector of $\mathrm{L}(0)$. It follows that the Jordan chain associated with the 0 eigenvalue is of length 2 . For the other vector x_2 in the Jordan chain we have $L'(0)x_1 + L(0)x_2 = 0$, where $L'(0)$ is the matrix of the derivatives of entries of L at $\lambda = 0$. Consequently, we have

$$X_F = \begin{bmatrix} 1 & 0 \\ 0 & -P/Q \end{bmatrix}, \; J_k = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; g = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix}.$$

Applying the boundary conditions, it follows that $g_1 = 0$. So $Z_k = 0$, $k = 1, 2, \ldots, n$, which implies that $0 \notin \sigma(J')$.

Case 2 : $\mu \neq 0$. The eigenvalues of $L(\lambda)$ are given by the expressions

$$\lambda_0 = 0, \lambda_1 = \frac{\mu^2 + Q^2 - P^2 + \sqrt{(\mu^2 + Q^2 - P^2)^2 - 4Q^2\mu^2}}{2Q\mu},$$
$$\lambda_2 = \frac{\mu^2 + Q^2 - P^2 - \sqrt{(\mu^2 + Q^2 - P^2)^2 - 4Q^2\mu^2}}{2Q\mu}.$$

It is clear from (2.18) that $\lambda_1\lambda_2 = 1$ and $(\lambda_1+\lambda_2)Q\mu = \mu^2 + Q^2 - P^2$.

If $\lambda_1 \neq \lambda_2$, the eigenvectors of $L(\lambda)$ associated with λ_i , i=0,1,2, are

$$x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x_1 = \begin{bmatrix} w_1 \\ 1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} w_2 \\ 1 \end{bmatrix}$$

where

$$w_1 = \frac{\mu\lambda_1 - Q}{P\lambda_1}, \quad w_2 = \frac{\mu\lambda_2 - Q}{P\lambda_2}.$$

Since all the eigenvalues of $L(\lambda)$ have only one eigenvector each, the finite Jordan pair is given by

$$X_F = \begin{bmatrix} 1 & w_1 & w_2 \\ 0 & 1 & 1 \end{bmatrix}, \quad J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}, \quad g = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix}.$$

It is easy to check that the vectors Z_k defined by (2.17) satisfy the matrix difference equation (2.15).

Now, we determine the vector g to satisfy the boundary conditions in (2.15). The first condition implies

$$(1 + w_1\lambda_1)g_1 + (1 + w_2\lambda_2)g_2 = 0,$$

and the second one implies

$$(\lambda_1^n + w_1\lambda_1^{n+1})g_1 + (\lambda_2^n + w_2\lambda_2^{n+1})g_2 = 0.$$

Combining them, we have the following 2×2 homogeneous linear system to solve

$$\begin{bmatrix} 1 + w_1\lambda_1 & 1 + w_2\lambda_2 \\ \lambda_1^n + w_1\lambda_1^{n+1} & \lambda_2^n + w_2\lambda_2^{n+1} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.19}$$

If $[g_1, g_2] = [0, 0]$, then $Z_k = 0$, for every $k = 0, 1, 2 \ldots$. So there must exist a nonzero solution to (2.19), hence the determinant of the matrix coefficient of (2.19) must equal zero. From this we obtain

$$(1 + w_1\lambda_1)(1 + w_2\lambda_2)(\lambda_2^n - \lambda_1^n) = 0. \tag{2.20}$$

If we assume $1 + w_i\lambda_i = 0$ then we get $\lambda_i = \frac{Q-P}{\mu}$. Moreover, solving $Q\mu\lambda_i^2 - (\mu^2 + Q^2 - P^2)$ $\lambda_i + Q\mu = 0$ with respect to μ, we obtain $\mu = \pm(Q-P)$ for $P \neq 0$. This implies $\lambda_1 = \lambda_2 = \pm 1$ which contradicts the assumption $\lambda_1 \neq \lambda_2$. Hence from (2.20) we conclude $\lambda_1^n - \lambda_2^n = 0$ and determine that $\lambda_1 = e^{i\theta}$, $\lambda_2 = e^{-i\theta}$, $\theta = \frac{k\pi}{n}$, $k = 1, 2, \ldots, n-1$ since $\lambda_1\lambda_2 = 1$. It is worth noticing that for each pair of λ's there are two μ's obtained from equation $\mu^2 - 2Q\mu\cos\theta + Q^2 - P^2 = 0$.

For the case $\lambda_1 = \lambda_2$, following the same analysis as above, we end up with the following solutions

$$\lambda_1 = \lambda_2 = 1 \begin{cases} \mu = Q + P, \ X_F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & \frac{-Q}{P} \end{bmatrix}, \quad J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \ g = \begin{bmatrix} g_0 \\ 0 \\ 0 \end{bmatrix} \\ \mu = Q - P, \ X_F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & \frac{-Q}{P} \end{bmatrix}, \ J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \ g = \begin{bmatrix} g_0 \\ 1 \\ 0 \end{bmatrix} \end{cases}$$

$$\lambda_1 = \lambda_2 = -1 \begin{cases} \mu = -Q - P, \; X_F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & \frac{Q}{P} \end{bmatrix}, \; J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \; g = \begin{bmatrix} g_0 \\ 0 \\ 0 \end{bmatrix} \\ \mu = P - Q, \quad X_F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & \frac{Q}{P} \end{bmatrix}, \quad J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \; g = \begin{bmatrix} g_0 \\ 1 \\ 0 \end{bmatrix}. \end{cases}$$

By considering the associated g's, we find that $\mu = \pm(Q - P) \in \sigma(J')$ which concludes the proof of this lemma. □

It is difficult to determine $\det(L(\lambda))$ explicitly when P and Q are real matrices. This is due to the fact that (2.16) is not a 2×2 matrix. Thus, applying the analysis above to obtain $\sigma(J')$ is not an easy task. Instead, we determine μ from each known λ from the scalar case. Specifically, we can show that for $\lambda = e^{\frac{k\pi}{n}i}$, the equation $\det(L(\lambda)) = 0$ can be simplified into

$$\det\left(\begin{bmatrix} Qe^{\frac{k\pi}{n}i} - \mu I & P \\ P & Qe^{-\frac{k\pi}{n}i} - \mu I \end{bmatrix}\right) = 0 \qquad (2.21)$$

which is equivalent to determining the eigenvalues of the matrix

$$S_k = \begin{bmatrix} Qe^{\frac{k\pi}{n}i} & P \\ P & Qe^{-\frac{k\pi}{n}i} \end{bmatrix}.$$

To eliminate the complex numbers involved, we perform the similarity transformation $R_k S_k R_k^{-1}$, where

$$R_k = \begin{bmatrix} I & -e^{\frac{k\pi}{n}i} I \\ iI & ie^{\frac{k\pi}{n}i} I \end{bmatrix}.$$

Then the problem at hand is transformed into the problem of determining the spectrum $\sigma(T_k)$ of

$$T_k = \begin{bmatrix} (Q - P)\cos\frac{k\pi}{n}, & (Q - P)\sin\frac{k\pi}{n} \\ -(Q + P)\sin\frac{k\pi}{n}, & (Q + P)\cos\frac{k\pi}{n} \end{bmatrix}.$$

Lemma 2.2 gives the basic idea as to how to tackle the matrix problem case. The following lemma is the corresponding result.

LEMMA 2.3. *Let J' be the matrix in (2.14) with P and Q being real matrices. Then its spectrum is given by*

$$\sigma(J') = \cup_{k=1}^{n-1} \sigma(T_k) \cup \sigma(P - Q) \cup \sigma(Q - P)$$

To prove it, first we state and prove another lemma.

LEMMA 2.4. *Define*

$$Y = \begin{bmatrix} \cos\frac{\pi}{n} & \sin\frac{\pi}{n} & \cdots & \cos\frac{(n-1)\pi}{n} & \sin\frac{(n-1)\pi}{n} & 1 & -1 \\ -\cos\frac{0\pi}{n} & \sin\frac{0\pi}{n} & \cdots & -\cos\frac{(n-1)0\pi}{n} & \sin\frac{(n-1)0\pi}{n} & -1 & -1 \\ \cos\frac{2\pi}{n} & \sin\frac{2\pi}{n} & \cdots & \cos\frac{(n-1)2\pi}{n} & \sin\frac{(n-1)2\pi}{n} & 1 & 1 \\ -\cos\frac{\pi}{n} & \sin\frac{\pi}{n} & \cdots & -\cos\frac{(n-1)\pi}{n} & \sin\frac{(n-1)\pi}{n} & -1 & 1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ \cos\frac{n\pi}{n} & \sin\frac{n\pi}{n} & \cdots & \cos\frac{(n-1)n\pi}{n} & \sin\frac{(n-1)n\pi}{n} & 1 & (-1)^n \\ -\cos\frac{(n-1)\pi}{n} & \sin\frac{(n-1)\pi}{n} & \cdots & -\cos\frac{(n-1)^2\pi}{n} & \sin\frac{(n-1)^2\pi}{n} & -1 & (-1)^n \end{bmatrix}, \qquad (2.22)$$

then the matrix Y is invertible.

Proof. An obvious permutation of rows and columns transforms the matrix Y to the matrix Y'

$$Y' = \begin{bmatrix} \cos\frac{\pi}{n} & \cdots & \cos\frac{(n-1)\pi}{n} & 1 & \sin\frac{\pi}{n} & \cdots & \sin\frac{(n-1)\pi}{n} & -1 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ \cos\frac{n\pi}{n} & \cdots & \cos\frac{(n-1)n\pi}{n} & 1 & \sin\frac{n\pi}{n} & \cdots & \sin\frac{(n-1)n\pi}{n} & (-1)^n \\ -\cos\frac{0\pi}{n} & \cdots & -\cos\frac{(n-1)0\pi}{n} & -1 & \sin\frac{0\pi}{n} & \cdots & \sin\frac{(n-1)0\pi}{n} & -1 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ -\cos\frac{(n-1)\pi}{n} & \cdots & -\cos\frac{(n-1)^2\pi}{n} & -1 & \sin\frac{(n-1)\pi}{n} & \cdots & \sin\frac{(n-1)^2\pi}{n} & (-1)^n \end{bmatrix} .$$

Apply then a sequence of elementary row and column operations on Y' as follows. First, add the ith row to the $(i+n+1)st$ one, for every $i = 1, 2, \ldots, (n-1)$, next, divide the ith row by 2, for every $i = (n+1), \ldots, 2n$, and then subtract the $(i+n+1)st$ row from the ith one, for every $i = 1, 2, \ldots, (n-1)$. After this series of operations takes place we permute some of the rows and columns of the resulting matrix and finally we end up with a matrix $C = \mathrm{diag}(A, B)$, where A is an $(n+1) \times (n+1)$ matrix with entries $a_{ij} = \cos\frac{(i-1)(j-1)\pi}{n}$ and B is an $(n-1) \times (n-1)$ matrix with entries $b_{ij} = \sin\frac{ij\pi}{n}$. Now let

$$K = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{bmatrix}_{(n-1)\times(n-1)} , \quad L = \begin{bmatrix} 0 & 2 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 2 & 0 \end{bmatrix}_{(n+1)\times(n+1)} .$$

It is readily checked that

$$LA = A \ \mathrm{diag}(2, 2\cos\frac{\pi}{n}, \ldots, 2\cos\frac{(n-1)\pi}{n}, -2),$$

$$KB = B \ \mathrm{diag}(2\cos\frac{\pi}{n}, 2\cos\frac{2\pi}{n}, \ldots, 2\cos\frac{(n-1)\pi}{n}).$$

From the equations above we see that each column of A is an eigenvector of L and all the eigenvalues of L are distinct, therefore A is invertible. So is B. It follows then that C is invertible too. On the other hand, we know that applying any nonsingular elementary operations on a matrix results in a nonsingular matrix if and only if the original one is nonsingular. This observation implies the invertibility of Y', and therefore that of Y, from the fact that $C = \mathrm{diag}(A, B)$. □

Proof of Lemma 2.3: Let us fix k. Then from $T_k \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}$, we have

$$\cos\frac{k\pi}{n}(Q-P)x + \sin\frac{k\pi}{n}(Q-P)y = \lambda x \ , \tag{2.23}$$

$$-\sin\frac{k\pi}{n}(Q+P)x + \cos\frac{k\pi}{n}(Q+P)y = \lambda y \ . \tag{2.24}$$

Combining appropriate multiples of equations (2.23)–(2.24) and simplifying them by using trigonometric identities, we get

$$\begin{aligned} Q\ (\cos \tfrac{(j+1)k\pi}{n} x + \sin \tfrac{(j+1)k\pi}{n} y)\ &+\ P\ (-\cos \tfrac{(j-1)k\pi}{n} x + \sin \tfrac{(j-1)k\pi}{n} y) \\ &=\ \lambda\ (\cos \tfrac{jk\pi}{n} x + \sin \tfrac{jk\pi}{n} y), \end{aligned} \tag{2.25}$$

$$\begin{aligned} Q\ (-\cos \tfrac{(j-1)k\pi}{n} x + \sin \tfrac{(j-1)k\pi}{n} y)\ &+\ P\ (\cos \tfrac{(j+1)k\pi}{n} x + \sin \tfrac{(j+1)k\pi}{n} y) \\ &=\ \lambda\ (-\cos \tfrac{jk\pi}{n} x + \sin \tfrac{jk\pi}{n} y). \end{aligned} \tag{2.26}$$

Let z be the following vector

$$\begin{aligned} z = \Big[&\cos \frac{k\pi}{n} x^T + \sin \frac{k\pi}{n} y^T, -x^T,\ \cos \frac{2k\pi}{n} x^T + \sin \frac{2k\pi}{n} y^T, -\cos \frac{k\pi}{n} x^T + \sin \frac{k\pi}{n} y^T, \ldots, \\ &\cos \frac{nk\pi}{n} x^T + \sin \frac{nk\pi}{n} y^T, -\cos \frac{(n-1)k\pi}{n} x^T + \sin \frac{(n-1)k\pi}{n} y^T \Big]^T. \end{aligned}$$

Then applying (2.23), (2.24), (2.25) and (2.26), it is easy to check that the vector z satisfies $J'z = \lambda z$. Similarly, if we consider $T_k \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \lambda \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ and construct the vectors z_1 and z_2 as above, corresponding to x_1, y_1 and x_2, y_2, respectively, then $J'z_2 = z_1 + \lambda z_2$. On the other hand we know that if $(P-Q)x_2 = x_1 + \lambda x_2$ and let $v_i = [x_i^T, -x_i^T, x_i^T, -x_i^T, \ldots,]^T$ and $\bar{v}_i = [-x_i^T, -x_i^T, x_i^T, x_i^T, \ldots,]$, $i = 1, 2$, then $J'v_2 = -v_1 - \lambda v_2$ and $J'\bar{v}_2 = \bar{v}_1 + \lambda \bar{v}_2$.

The analysis so far can be summarized as follows. For each T_k we know that there exists a nonsingular matrix X_k such that $T_k X_k = X_k J_k$, $k = 1, 2, \ldots, n-1$, where J_k is the Jordan canonical form of T_k. Similarly we have $(P-Q)X_n = X_n J_n$. Note that each X_k is a $2m \times 2m$ matrix, except for $k = n$ where X_n is of order m. Let

$$V = (Y \otimes I_m)\mathrm{diag}(X_1, X_2, \ldots, X_{n-1}, X_n, X_n),$$

where Y is defined and proved to be invertible in Lemma 2.4. It is clear that V is also nonsingular. Consequently the analysis above shows that

$$J'V = V\mathrm{diag}(J_1, J_2, \ldots, J_{n-1}, -J_n, J_n),$$

and the assertion of the lemma follows. □

Noting that (2.14) gives $Q - P = A_4 A_2^{-1}$ and $Q + P = -A_3 A_1^{-1}$, we conclude this section with the principal result about the eigenvalues of the Jacobi iteration matrix for the HERMCOL equations.

THEOREM 2.2. *Let J be the block Jacobi iteration matrix corresponding to (2.1) with the partition P_{II}. Then its spectrum is given by*

$$\sigma(J) = \cup_{k=1}^{n-1} \sigma(T_k) \cup \sigma(A_4 A_2^{-1}) \cup \sigma(-A_4 A_2^{-1})$$

where

$$T_k = \begin{bmatrix} A_4 A_2^{-1} \cos \frac{k\pi}{n} & A_4 A_2^{-1} \sin \frac{k\pi}{n} \\ A_3 A_1 \sin \frac{k\pi}{n} & -A_3 A_1^{-1} \cos \frac{k\pi}{n} \end{bmatrix}.$$

Remark: Note that the coefficient matrix in (2.1) was obtained from a particular class of HERMCOL equations by eliminating some unknowns symbolically.

2.4 Iterative Methods for the Solution of a Model Problem

In this section we consider the collocation equations obtained by the discretization of the model PDE problem with Dirichlet or Neumann boundary conditions defined on the unit square. Using the analysis of the previous sections; we derive the eigenvalue spectra of the block Jacobi iteration matrices J_1 and J_2 corresponding to the block partitionings P_I and P_{II}, respectively. Then the analysis of the optimal SOR method for the Dirichlet problem is made and optimal results are obtained for the method based on partitionaing P_{II}. For the block SOR method based on partitioning P_I, optimal results are already known [14]. We conclude the present section with the analysis of the optimal SOR method for Neumann boundary conditions.

2.4.1 The Dirichlet Case

We consider the iterative solution of the interior collocation equations associated with the following Dirichlet boundary value problem

$$\begin{gathered} u_{xx} + u_{yy} = f \ \text{ in } \ R = (0,1) \times (0,1)\ , \\ u = g \ \text{ on } \ \partial R. \end{gathered} \tag{2.27}$$

and a uniform mesh ($h_x = 1/n = 1/m = h_y$). After applying Papatheodorou's tensor-product ordering scheme shown in Section 1.3 (see Figure 1.2) and factoring out $(1/9h^2)$, the collocation matrix is the same as the matrix A in (2.1). For this particular problem, the entries of A_i for $i = 1, 2, 3, 4$ are independent of h and have the same structure as A. More specifically, we have

$$A_i = \left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & -a_4 \\ a_3 & a_4 & a_1 & -a_2 \end{array}\right]_{\otimes(2n)} . \tag{2.28}$$

The values of a_j for j=1,2,3,4 corresponding to the A_i's are listed below (see [30]).

	a_1	a_2	a_3	$-a_4$
A_1	$-24 - 18\sqrt{3}$	$-12 - 8\sqrt{3}$	24	$-3 - \sqrt{3}$
A_2	$-12 - 8\sqrt{3}$	$-3 - 2\sqrt{3}$	$3 - \sqrt{3}$	0
A_3	24	$3 - \sqrt{3}$	$-24 + 18\sqrt{3}$	$12 - 8\sqrt{3}$
A_4	$3 + \sqrt{3}$	0	$-12 + 8\sqrt{3}$	$3 - 2\sqrt{3}$

In this case we have the INTCOL coefficient matrix whose entries are explicitly expressed. This motivates us to try and to find analytic expressions for the elements of $\sigma(J)$. For this, some preliminary analysis is needed.

LEMMA 2.5. *Let the matrices A and B be defined as follows*

$$A = \left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & -a_4 \\ a_3 & a_4 & a_1 & -a_2 \end{array}\right]_{\otimes(2n)} , \quad B = \left[\begin{array}{cc|cc} b_1 & b_2 & b_3 & -b_4 \\ b_3 & b_4 & b_1 & -b_2 \end{array}\right]_{\otimes(2n)}$$

and suppose that B is nonsingular and $a_2b_4 \neq a_4b_2$. Then the generalized eigenproblem $A^Tx = \lambda B^Tx$ has eigenvalues λ given by the expressions

(i) $\lambda = \frac{a_2+a_4}{b_2+b_4}$ associated with the eigenvector $x = [1, 1, -1, -1, \ldots]^T$.

(ii) $\lambda = \frac{a_2-a_4}{b_2-b_4}$ associated with the eigenvector $x = [1, -1, 1, -1, \ldots]^T$.

(iii) λ *satisfies the equation* $\frac{f_1(\lambda)f_2(\lambda)-f_3(\lambda)f_4(\lambda)}{f_1(\lambda)f_4(\lambda)-f_2(\lambda)f_3(\lambda)} = \cos\theta$, $\theta = \frac{k\pi}{n}$, $k = 1, 2, \ldots, (n-1)$, *with associated eigenvector* $x = [\rho_1 + \rho_2 g, w_1\rho_1 + w_2\rho_2 g, \ldots, \rho_1^n + \rho_2^n g, w_1\rho_1^n + w_2\rho_2^n g]^T$, *where* $f_i(\lambda) = a_i - \lambda b_i$, $i = 1, 2, 3, 4$, $\rho_1 = e^{i\theta}$, $\rho_2 = e^{-i\theta}$, $w_1 = \frac{\rho_1 f_2(\lambda) - f_4(\lambda)}{f_2(\lambda) - \rho_1 f_4(\lambda)}$, $w_2 = \frac{1}{w_1}$ *and* $g = -\frac{w_1 f_2(\lambda) + f_4(\lambda)}{w_2 f_2(\lambda) + f_4(\lambda)}$.

Proof. To solve the generalized eigenproblem (cf. [11, pp. 251-266]) $A^T x = \lambda B^T x$ is equivalent to solving the matrix difference equation

$$B_0 Z_{k-1} + B_1 Z_k + B_2 Z_{k+1} = 0, \qquad k = 1, 2, \ldots, n,$$

where

$$B_0 = \begin{bmatrix} -f_4(\lambda) & -f_2(\lambda) \\ 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} f_2(\lambda) & f_4(\lambda) \\ f_3(\lambda) & f_1(\lambda) \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ f_1(\lambda) & f_3(\lambda) \end{bmatrix},$$

subject to the boundary conditions

$$B_0 Z_0 = 0\ , \ \begin{bmatrix} 0 & 0 \\ f_1(\lambda) & f_3(\lambda) \end{bmatrix} Z_{n+1} = \begin{bmatrix} 0 & 0 \\ -f_2(\lambda) - f_4(\lambda) & -f_1(\lambda) - f_2(\lambda) \end{bmatrix} Z_n.$$

For simplicity in the following discussion we assume that there is no λ such that $(f_1(\lambda) \pm f_3(\lambda))^2 + (f_2(\lambda) \pm f_4(\lambda))^2 = 0$. Following the same analysis as in the proof of Lemma 2.2 with ρ playing the role of λ, we get

$$L(\rho) = \begin{bmatrix} f_2(\lambda)\rho - f_4(\lambda) & f_4(\lambda)\rho - f_2(\lambda) \\ f_1(\lambda)\rho^2 + f_3(\lambda)\rho & f_3(\lambda)\rho^2 + f_1(\lambda)\rho \end{bmatrix}.$$

Thus, we have

$$\begin{aligned} \det(L(\rho)) &= -\rho[(f_1(\lambda)f_4(\lambda) - f_2(\lambda)f_3(\lambda))\rho^2 - 2(f_1(\lambda)f_2(\lambda) - f_3(\lambda)f_4(\lambda))\rho \\ &+ (f_1(\lambda)f_4(\lambda) - f_2(\lambda)f_3(\lambda))]. \end{aligned}$$

We distinguish two cases.

Case 1 : $f_1(\lambda)f_4(\lambda) - f_2(\lambda)f_3(\lambda) = 0$. In this case 0 is a double eigenvalue of L(ρ) and there is only one eigenvector associated with it. So, we have

$$X_F = \begin{bmatrix} f_2(\lambda) & -f_4(\lambda) \\ -f_4(\lambda) & f_2(\lambda) \end{bmatrix}, \quad J_F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad g = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix}.$$

Applying the boundary conditions, it follows that $g_1 = 0$ if $f_2(\lambda) \pm f_4(\lambda) \neq 0$. On the other hand, if $f_2(\lambda) = \pm f_4(\lambda)$ ($\neq 0$) then $f_1(\lambda) = \pm f_3(\lambda)$, which contradicts the assumption we made on the f_i's for this case. So we obtain $g_1 = 0$, which implies that $Z_k = 0$, for $k = 1, 2, \ldots, n$. Hence there does not exist a nonzero solution to the matrix difference equation.

Case 2 : $f_1(\lambda)f_4(\lambda) - f_2(\lambda)f_3(\lambda) \neq 0$. In this case there are three eigenvalues of $L(\rho)$. Let them be $\rho_0 = 0$, ρ_1 and ρ_2. It is clear that $\rho_1\rho_2 = 1$ and $\rho_1 + \rho_2 = 2\ \frac{f_1(\lambda)f_2(\lambda)-f_3(\lambda)f_4(\lambda)}{f_1(\lambda)f_4(\lambda)-f_2(\lambda)f_3(\lambda)}$. If $\rho_1 \neq \rho_2$, then

$$X_F = \begin{bmatrix} f_2(\lambda) & 1 & 1 \\ -f_4(\lambda) & w_1 & w_2 \end{bmatrix}, \quad J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \rho_1 & 0 \\ 0 & 0 & \rho_2 \end{bmatrix}, \quad g = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix},$$

where $w_i = \frac{\rho_i f_2(\lambda) - f_4(\lambda)}{f_2(\lambda) - \rho_i f_4(\lambda)}, i = 1, 2$. Applying the boundary conditions, we arrive at the following linear system

$$\begin{bmatrix} w_1 f_2(\lambda) + f_4(\lambda) & w_2 f_2(\lambda) + f_4(\lambda) \\ (w_1 f_2(\lambda) + f_4(\lambda))\rho_1^n & (w_2 f_2(\lambda) + f_4(\lambda))\rho_2^n \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

In order to have a nonzero solution for $[g_1, g_2]^T$ the determinant of the matrix coefficient of the linear system must be zero. But $w_i f_2(\lambda) + f_4(\lambda) = 0$ implies $\rho_i = 0$ which contradicts the fact that $\rho_1 \rho_2 = 1$. It follows then that $\rho_1^n = \rho_2^n$. Combining this with $\rho_1 \rho_2 = 1$ and $\rho_1 \neq \rho_2$, it follows that $\rho_1 = e^{i\theta}$, $\rho_2 = e^{-i\theta}$, where $\theta = \frac{k\pi}{n}$, $k = 1, 2, \ldots, (n-1)$. Note that there are infinitely many solutions to the linear system above. If we pick $[g_1, g_2]^T = [1, g]^T$, where $g = -\frac{w_1 f_2(\lambda) + f_4(\lambda)}{w_2 f_2(\lambda) + f_4(\lambda)}$, assertion (iii) of the lemma follows.

If we consider now $\rho_1 = \rho_2$ then we follow a similar analysis for each particular case. The corresponding results are summarized below.

$\rho_1 = \rho_2 = 1$,

$$\begin{cases} f_2(\lambda) - f_4(\lambda) = 0,\ X_F = \begin{bmatrix} f_2(\lambda) & 1 & 0 \\ -f_4(\lambda) & -1 & -\frac{f_1(\lambda) - f_3(\lambda)}{f_1(\lambda) + f_3(\lambda)} \end{bmatrix},\ J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},\ g = \begin{bmatrix} g_0 \\ 1 \\ 0 \end{bmatrix} \\ f_1(\lambda) + f_3(\lambda) = 0,\ X_F = \begin{bmatrix} f_2(\lambda) & 1 & 0 \\ -f_4(\lambda) & 1 & \frac{f_2(\lambda) + f_4(\lambda)}{f_2(\lambda) - f_4(\lambda)} \end{bmatrix},\quad J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},\ g = \begin{bmatrix} g_0 \\ 0 \\ 0 \end{bmatrix} \end{cases}$$

$\rho_1 = \rho_2 = -1$,

$$\begin{cases} f_2(\lambda) + f_4(\lambda) = 0,\ X_F = \begin{bmatrix} f_2(\lambda) & 1 & 0 \\ -f_4(\lambda) & 1 & -\frac{f_1(\lambda) + f_3(\lambda)}{f_1(\lambda) - f_3(\lambda)} \end{bmatrix},\ J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix},\ g = \begin{bmatrix} g_0 \\ 1 \\ 0 \end{bmatrix} \\ f_1(\lambda) - f_3(\lambda) = 0,\ X_F = \begin{bmatrix} f_2(\lambda) & 1 & 0 \\ -f_4(\lambda) & -1 & \frac{f_2(\lambda) - f_4(\lambda)}{f_2(\lambda) + f_4(\lambda)} \end{bmatrix},\ J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix},\ g = \begin{bmatrix} g_0 \\ 0 \\ 0 \end{bmatrix}. \end{cases}$$

Assertions (i) and (ii) of the lemma follow from the solutions of $f_2(\lambda) \pm f_4(\lambda) = 0$ and considering the corresponding Z_k, $k = 1, 2, \ldots, n$. □

LEMMA 2.6. *Let A_i, $i = 1, 2, 3, 4$, be the matrices in (2.28). Then there exists a nonsingular matrix X such that $A_4^T X = A_2^T X D$ and $A_3^T X = A_1^T X \bar{D}$, where*

$$D = diag(\lambda_1, \lambda_2, \ldots, \lambda_{2n}) = diag\left(\frac{3 - 2\sqrt{3}}{3 + 2\sqrt{3}}, \frac{3 - 2\sqrt{3}}{-3 - 2\sqrt{3}}, \alpha_1^+, \alpha_1^-, \ldots, \alpha_{n-1}^+, \alpha_{n-1}^-\right), \tag{2.29}$$

$$\bar{D} = diag(\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_{2n}) = diag\left(\frac{9 - 7\sqrt{3}}{9 + 7\sqrt{3}}, \frac{15 - 9\sqrt{3}}{-15 - 9\sqrt{3}}, \beta_1^+, \beta_1^-, \ldots, \beta_{n-1}^+, \beta_{n-1}^-\right), \tag{2.30}$$

and

$$\begin{aligned} \alpha_k^\pm &= \frac{3\sqrt{3} \pm \sqrt{43 + 40\cos\theta_k - 2\cos^2\theta_k}}{(-28 - 16\sqrt{3}) + (\sqrt{3} + 1)\cos\theta_k}, \\ \beta_k^\pm &= \frac{(37 + 8\cos\theta_k) \pm 3\sqrt{3}\sqrt{43 + 40\cos\theta_k - 2\cos^2\theta_k}}{(-64 - 36\sqrt{3}) + (19 + 9\sqrt{3})\cos\theta_k}, \\ \theta_k &= \frac{k\pi}{n}. \end{aligned}$$

Proof. First, note that the matrices of A_1 and A_2 are nonsingular by applying Theorem 2.1 in [30]. Then by setting $A = A_4$ and $B = A_2$ in the preceding lemma, a lengthy computation shows that all the eigenvalues of the pencil (A_4^T, A_2^T) (cf. [11]) lie on the curves λ_- and λ_+ of Figure 2.2 and have the values of the entries of the matrix given in (2.29).

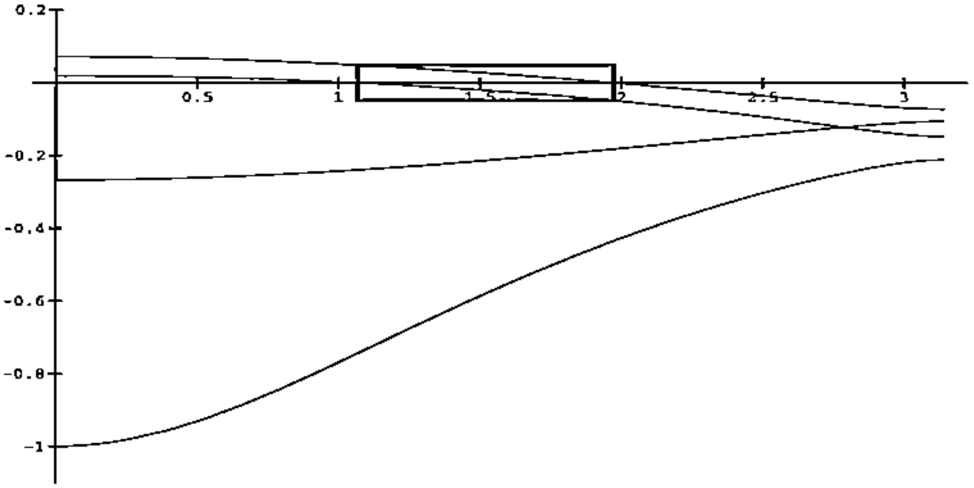


FIG. 2.2. *Curves contain the eigenvalues of the matrix pencils* (A_4^T, A_2^T) *and* (A_3^T, A_1^T). *The four curves ordered from top left represent* λ_-, $\bar{\lambda}_-$, λ_+ *and* $\bar{\lambda}_+$ *as functions of* θ, *where* $\lambda_\pm = \frac{3(3)^{1/2} \pm (43+40\cos\theta - 2\cos^2\theta)^{1/2}}{(-28-16(3)^{1/2})+(3^{1/2}+1)\cos\theta}$, $\bar{\lambda}_\pm = \frac{37+8\cos\theta \pm 3(3)^{1/2}(43+40\cos\theta-2\cos^2\theta)^{1/2}}{(-64-36(3)^{1/2})+(10+9(3)^{1/2})\cos\theta}$. *The rectangle in the figure is defined for every* θ *in* $[\cos^{-1}(\frac{122-54(3)^{1/2}}{59}), \cos^{-1}(10-6(3)^{1/2})]$.

From the sign of the derivatives of the functions of the curves λ_- and λ_+ on the interval $(0, \pi)$, it is concluded that λ_- is decreasing, while λ_+ is increasing. Moreover, they do not have any intersection point since $\lambda_-(\pi) > \lambda_+(\pi)$. It follows then that all the eigenvalues $\lambda_i, i = 1, 2, \ldots, (2n)$, are distinct. By rearranging the corresponding eigenvectors it follows that there does exist a nonsingular matrix X such that $A_4^T X = A_2^T X D$, where D is defined as in (2.29).

To complete the proof, it suffices to show that if $A_4^T x_i = \lambda_i A_2^T x_i$, then $A_3^T x_i = \bar{\lambda}_i A_1^T x_i$, with x_i being the ith column of X. It is clear that the claim holds for $i = 1, 2$. So we consider $i = 3, 4, \ldots, (2n)$ and fix $x = x_i$, $\lambda = \lambda_i$. By virtue of Lemma 2.5, there exist θ and g such that $x = [\rho_1 + \rho_2 g, w_1\rho_1 + w_2\rho_2 g, \ldots, \rho_1^n + \rho_2^n g, w_1\rho_1^n + w_2\rho_2^n g]^T$, where ρ_1, ρ_2, w_1, w_2 are as defined there. Let θ, x and λ be fixed. Set then $A = A_3$, $B = A_1$ in the previous lemma, and use the same symbols with a bar (to distinguish them from the ones in the previous case) to denote the corresponding quantities in the present case. For $\bar{\theta} = \theta$ there are two solutions for $\bar{\lambda}$ from

$$\frac{\bar{f}_1(\bar{\lambda})\bar{f}_2(\bar{\lambda}) - \bar{f}_3(\bar{\lambda})\bar{f}_4(\bar{\lambda})}{\bar{f}_1(\bar{\lambda})\bar{f}_4(\bar{\lambda}) - \bar{f}_2(\bar{\lambda})\bar{f}_3(\bar{\lambda})} = \cos\theta = \frac{f_1(\lambda)f_2(\lambda) - f_3(\lambda)f_4(\lambda)}{f_1(\lambda)f_4(\lambda) - f_2(\lambda)f_3(\lambda)}. \tag{2.31}$$

Since $\frac{A}{B} = \frac{C}{D}$ implies $\frac{A-B}{A+B} = \frac{C-D}{C+D}$, (2.31) implies

$$\frac{(\bar{f}_1(\bar{\lambda}) + \bar{f}_3(\bar{\lambda})(\bar{f}_2(\bar{\lambda}) - \bar{f}_4(\bar{\lambda}))}{(\bar{f}_1(\bar{\lambda}) - \bar{f}_3(\bar{\lambda})(\bar{f}_2(\bar{\lambda}) + \bar{f}_4(\bar{\lambda}))} = \frac{(f_1(\lambda) + f_3(\lambda))(f_2(\lambda) - f_4(\lambda))}{(f_1(\lambda) - f_3(\lambda))(f_2(\lambda) + f_4(\lambda))}. \tag{2.32}$$

Recall that we are in the situation where λ is given, and we want to solve for $\bar{\lambda}$ from equation (2.31). On the other hand, it is easy to verify that for this case the following statement holds. If

$$\frac{\bar{f}_2(\bar{\lambda}) - \bar{f}_4(\bar{\lambda})}{\bar{f}_2(\bar{\lambda}) + \bar{f}_4(\bar{\lambda})} = \frac{f_2(\lambda) - f_4(\lambda)}{f_2(\lambda) + f_4(\lambda)} \tag{2.33}$$

then

$$\frac{\bar{f}_1(\bar{\lambda}) + \bar{f}_3(\bar{\lambda})}{\bar{f}_1(\bar{\lambda}) - \bar{f}_3(\bar{\lambda})} = \frac{f_1(\lambda) + f_3(\lambda)}{f_1(\lambda) - f_3(\lambda)}. \tag{2.34}$$

It follows then that one of the solutions of (2.31) is the solution of (2.33). We take it as being the $\bar{\lambda}$ we have been seeking. For this $\bar{\lambda}$, we obtain the corresponding $\bar{w}_1$, $\bar{w}_2$, $\bar{g}$. Since $\bar{\lambda}$ satisfies (2.33), we also get that $\bar{w}_i = w_i$, $\bar{g} = g$. So we obtain the equality $A_3^T x = \bar{\lambda} A_1^T x$. Note that we have not shown that the elements of $\bar{D}$ have the order that corresponds to the one in (2.30). For this we go back to equation (2.31), and see that for a given θ, we get two solutions for λ and $\bar{\lambda}$. Let us call them $\lambda_{\pm}$ and $\bar{\lambda}_{\pm}$, respectively. Since both sides of equation (2.32) are decreasing with respect to either λ, or $\bar{\lambda}$ individually, $\lambda_+(\lambda_-)$ corresponds to $\bar{\lambda}_+(\bar{\lambda}_-)$. Hence D_2 is determined by D_1 as in (2.29). This completes the proof. □

2.4.1.1 Spectra of the Block Jacobi Iteration Matrix Let J_1 and J_2 be the block Jacobi iteration matrices associated with the partitionings P_I and P_{II} of the INTCOL coefficient matrix, respectively. We now derive analytic expressions for $\sigma(J_1)$ and $\sigma(J_2)$.

Since it can be shown that the matrices A_1 and A_2 are nonsingular by Theorem 2.1 in [30], then from Lemma 2.6 we have that the matrix $A_4 A_2^{-1} A_3 A_1^{-1}$ is invertible. Therefore the blocks of R in (2.4) can be found explicitly. More specifically

$$R_{22} = (A_1 - A_4 A_2^{-1} A_3)^{-1}(-A_4 A_2^{-1} A_1 + A_3).$$

Using the fact that for any two matrices A and B, $\sigma(AB) = \sigma(BA)$, we get that

$$\begin{aligned} \sigma(R_{22}) &= \sigma((-A_4 A_2^{-1} A_1 + A_3)(A_1 - A_4 A_2^{-1} A_3)^{-1}) \\ &= \sigma((-A_4 A_2^{-1} + A_3 A_1^{-1})(I - A_4 A_2^{-1} A_3 A_1^{-1})^{-1}). \end{aligned}$$

Applying Lemma 2.6 we have that

$$\sigma(R_{22}) = \left\{ \frac{\bar{\lambda}_i - \lambda_i}{1 - \bar{\lambda}_i \lambda_i}, \quad i = 1, 2, \ldots, 2n \right\},$$

since $A_2^{-T} A_4^T$ and $A_1^{-T} A_3^T$ commute. From Lemma 2.6, we have that $X^T A_4 A_2^{-1} (X^T)^{-1} = D$ and $X^T A_3 A_1^{-1} (X^T)^{-1} = \bar{D}$. By a similarity transformation with the transformation matrix $\text{diag}(X^T, X^T)$ and an obvious permutation of rows and columns, it is seen that T_k

of Theorem 2.2 is similar to diag$(D_1, D_2, \ldots, D_n)$, where $D_i = \begin{bmatrix} \lambda_i \cos\frac{k\pi}{n}, & \lambda_i \sin\frac{k\pi}{n} \\ \bar{\lambda}_i \sin\frac{k\pi}{n}, & -\bar{\lambda}_i \cos\frac{k\pi}{n} \end{bmatrix}$. So, we have

$$\sigma(T_k) = \{\mu | \mu^2 - (\bar{\lambda}_i - \lambda_i)\mu \cos\frac{k\pi}{n} - \bar{\lambda}_i\lambda_i = 0, i = 1, 2, \ldots, 2n\}.$$

Combining the above results with those of Theorems 2.1 and 2.2, we conclude that

$$\sigma(J_1) = \{0\} \cup \left\{ \cup_{k=1}^{n-1} \{\mu | \mu + \frac{1}{\mu} = 2\frac{\bar{\lambda}_i - \lambda_i}{1 - \bar{\lambda}_i\lambda_i} \cos\frac{k\pi}{n}, \;\; i = 1, 2, \ldots, 2n\} \right\}, \tag{2.35}$$

$$\begin{aligned} \sigma(J_2) &= \{\pm\lambda_1, \ldots, \pm\lambda_{2n}\} \\ &\cup \left\{ \cup_{k=1}^{n-1} \{\mu | \mu^2 - (\bar{\lambda}_i - \lambda_i)\mu \cos\tfrac{k\pi}{n} - \bar{\lambda}_i\lambda_i = 0, \;\; i = 1, 2, \ldots, 2n\} \right\}, \end{aligned} \tag{2.36}$$

where $\lambda_i, \bar{\lambda}_i$ are the ones of Lemma 2.6. Note also that zero is an eigenvalue of J_1 with multiplicity $4n$.

Recall now that equations (2.29) and (2.30) imply that the λ_i, $\bar{\lambda}_i$ lie on the curves in Figure 2.2. This implies that λ_i, $\bar{\lambda}_i$ are real numbers with magnitudes less than 1. It follows then that $\mu + \frac{1}{\mu}$ is real and has absolute value less than 2 which implies that all the eigenvalues of J_1, except 0, are complex and lie on the circumference of the unit circle. Therefore, the spectral radius $\rho(J_1)$ of J_1 is equal to 1. On the other hand, in view of Figure 2.2 and solving the equation in (2.36), we have that the spectral radius $\rho(J_2)$ of J_2 is equal to

$$\rho(J_2) := a = \frac{1}{2}\left((\lambda_3 - \bar{\lambda}_3)\cos\frac{\pi}{n} + \sqrt{(\lambda_3 - \bar{\lambda}_3)^2 \cos^2\left(\frac{\pi}{n}\right) + 4\lambda_3\bar{\lambda}_3} \right), \tag{2.37}$$

where λ_3, $\bar{\lambda}_3$ are those of Lemma 2.6. By inspecting the expression above, we also find that it is bounded above by $|\bar{\lambda}_3|$. Thus we conclude that for any discretization grid size n, $\rho(J_2) < |\bar{\lambda}_3| < \rho(J_1) = 1$. Consequently for the model problem in Section 2.4.1, the Jacobi iterative method associated with the partitioning P_{II} converges, but the same method associated with the partitioning P_I does not converge (because there does exist at least one complex $\mu \in \sigma(J_1)$ with modulus 1).

2.4.1.2 Optimal SOR The optimal SOR method for the case where J_1 is the Jacobi matrix has been already obtained in [14], so we consider only the case where the Jacobi matrix is J_2. Recall that J_2 is consistently ordered weakly cyclic of index 2. Therefore the Young-Eidson's algorithm [42] (see also [41, pp. 194–200]) can be applied to determine the optimal SOR method. To apply the algorithm, the hull (smallest convex polygon) of $\sigma(J_2)$ is required. For this we solve the equation for μ in (2.36) to obtain

$$\mu = \frac{(\lambda_j - \bar{\lambda}_j)\cos\frac{k\pi}{n} \pm \sqrt{\left((\lambda_j - \bar{\lambda}_j)\cos\frac{k\pi}{n}\right)^2 + 4\lambda_j\bar{\lambda}_j}}{2}. \tag{2.38}$$

For real μ we have already found that $\max|\mu| = a$ in (2.37). However, μ is a complex number when λ_j and $\bar{\lambda}_j$ lie inside the rectangle illustrated and defined in Figure 2.2 . Furthermore, for a given pair λ_j, $\bar{\lambda}_j$ satisfying $\lambda_j\bar{\lambda}_j < 0$, all the complex eigenvalues of J_2 associated with them must lie on the circumference of the circle centered at (0,0) and with radius $\sqrt{-\lambda_j\bar{\lambda}_j}$. Let b be the maximum value of $\sqrt{-\lambda_j\bar{\lambda}_j}$ among those j such that $-\bar{\lambda}_j\lambda_j > 0$, i.e.,

$$b = \max_k \sqrt{-\lambda_-(\frac{k\pi}{n})\bar{\lambda}_-(\frac{k\pi}{n})}, \ \frac{k\pi}{n} \in (\cos^{-1}(\frac{122-54\sqrt{3}}{59}) \ , \ \cos^{-1}(10-6\sqrt{3})). \tag{2.39}$$

Then, it follows that all the complex eigenvalues of J_2 lie inside or on the circumference of the circle with center at (0,0) and radius b. On the other hand, from (2.37), we have $a = \rho(J_2) \in \sigma(J_2)$. If n is even, we may put $k = \frac{n}{2}$ in (2.38) which implies that $bi \in \sigma(J_2)$, where i is the imaginary unit. Thus the ellipse with semiaxes a and b is the optimal enclosing ellipse of $\sigma(J_2)$. Therefore in this case we get

$$\omega_{opt} = \frac{2}{1+(1+b^2-a^2)^{1/2}}, \quad \rho(\mathcal{L}_{\omega_{opt}}) = \left(\frac{a+b}{1+(1+b^2-a^2)^{1/2}}\right)^2 \tag{2.40}$$

where $\mathcal{L}_\omega$ is the associated block SOR iteration matrix with overrelaxation parameter ω. In case n is odd, $bi \notin \sigma(J_2)$. However, the value of ω given in (2.40) is still a very good approximation to ω_{opt} in the present case, because that b is only slightly greater than the imaginary semiaxis of the corresponding optimum capturing ellipse and tends to the optimal one ($b = 0.0237973$) when $n \to \infty$. Two examples of $\sigma(J_2)$ for each of the two cases of n even and n odd are illustrated in Figure 2.3.

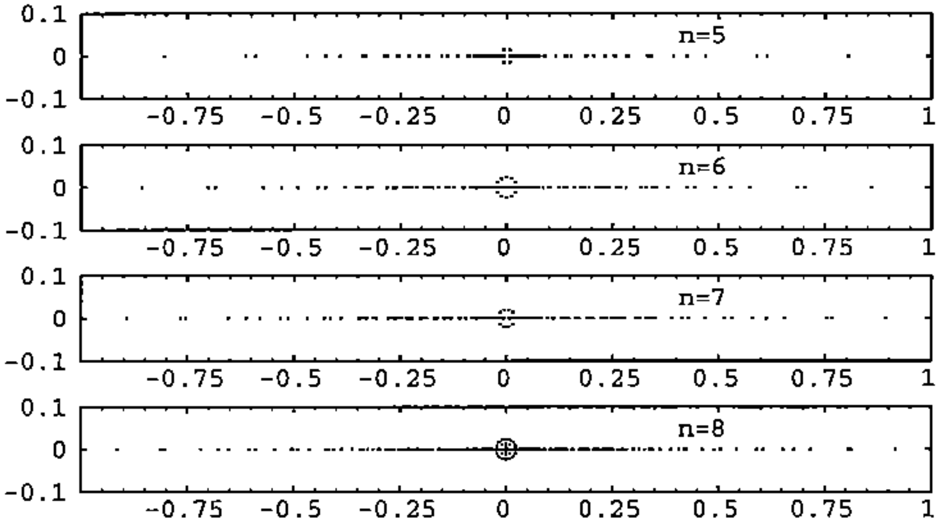


FIG. 2.3. *The spectrum $\sigma(J_2)$ of the Jacobi matrix J_2 associated with the partitioning P_{II} of the iterior collocation matrix.*

2.4.2 The Neumann Case Here we consider the iterative solution of the interior collocation equations associated with the following Neumann boundary value problem

$$\begin{aligned} u_{xx} + u_{yy} &= f \ \text{ in } \ R = (0,1)\times(0,1)\ , \\ \partial u/\partial n &= g \ \text{ on } \ \partial R\ . \end{aligned} \tag{2.41}$$

and a uniform mesh. For the analysis below we introduce a similar notation to that in preliminary section of this chapter consisting of the matrix

$$[A|B]_{\otimes(2\bar{n})} = \begin{bmatrix} a & B & & & \\ & A & B & & \\ & & \ddots & & \\ & & & A & B \\ & & & & A & b \end{bmatrix}, \quad a = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}, \quad b = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}.$$

which differs only in the definition of the vectors a and b.

Using Papatheodorou's tensor-product ordering of Section 1.3 and factoring out $(1/9h^2)$, the INTCOL coefficient matrix has the form

$$A = \left[\begin{array}{cc|cc} A_1 & A_2 & A_3 & -A_4 \\ A_3 & A_4 & A_1 & -A_2 \end{array}\right]_{\otimes(2\bar{n})}.$$

For this particular problem, the entries of A_i, $i = 1, 2, 3, 4$, are independent of h and have the same structure as before, namely

$$A_i = \left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & -a_4 \\ a_3 & a_4 & a_1 & -a_2 \end{array}\right]_{\otimes(2\bar{n})}.$$

The values of a_j corresponding to A_i are the ones given in Section 2.4.1. Following the analysis developed in Section 2.3, we obtain that the corresponding block Jacobi iteration matrix J' is given by

$$J' = \left[\begin{array}{ccccccccc} 0 & P & Q & & & & & & \\ P+Q & 0 & 0 & & & & & & \\ & 0 & 0 & P & Q & & & & \\ & Q & P & 0 & 0 & & & & \\ & & & & \ddots & & & & \\ & & & & & 0 & 0 & P & Q \\ & & & & & Q & P & 0 & 0 \\ & & & & & & 0 & 0 & P+Q \\ & & & & & & Q & P & 0 \end{array}\right]$$

where P and Q are defined in the same way as in (2.14). Through the similarity transformation $SJ'S^{-1}$, where S=diag$(1, 1, -1, -1, 1, 1, \ldots)$, J' is transformed to the matrix J''

$$J'' = \left[\begin{array}{ccccccccc} 0 & P & -Q & & & & & & \\ P+Q & 0 & 0 & & & & & & \\ & 0 & 0 & P & -Q & & & & \\ & -Q & P & 0 & 0 & & & & \\ & & & & \ddots & & & & \\ & & & & & 0 & 0 & P & -Q \\ & & & & & -Q & P & 0 & 0 \\ & & & & & & 0 & 0 & P+Q \\ & & & & & & -Q & P & 0 \end{array}\right].$$

Note that J'' is of exactly the same structure as J' in (2.14) with the only difference being that $-Q$ is replaced by Q. Applying Lemma 2.3, we then have that, in this case

$$\sigma(J') = \sigma(J'') = (\cup_{k=1}^{n-1}\sigma(T_k)) \cup \sigma(P+Q) \cup \sigma(-Q-P)$$

where

$$T_k = \begin{bmatrix} (-Q-P)\cos\frac{k\pi}{n}, & (-Q-P)\sin\frac{k\pi}{n} \\ -(-Q+P)\sin\frac{k\pi}{n}, & (-Q+P)\cos\frac{k\pi}{n} \end{bmatrix}.$$

Note that $Q - P = A_4A_2^{-1}$, $Q + P = -A_3A_1^{-1}$. Hence, we have

$$\sigma(J') = \cup_{k=1}^{n-1}\sigma(T_k) \cup \sigma(A_3A_1^{-1}) \cup \sigma(-A_3A_1^{-1}) \tag{2.42}$$

where

$$T_k = \begin{bmatrix} A_3A_1^{-1}\cos\frac{k\pi}{n}, & A_3A_1^{-1}\sin\frac{k\pi}{n} \\ A_4A_2\sin\frac{k\pi}{n}, & -A_4A_2^{-1}\cos\frac{k\pi}{n} \end{bmatrix}.$$

Let $P_1 = \text{diag}(I, -I, I, -I, \ldots)$, where I is the 2×2 identity matrix, and $P_2 = \text{diag}(1, -I_2, I_2, -I_2, I_2, \ldots, (-1)^n)$, where $I_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Then we have

$$P_1 \left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & -a_4 \\ a_3 & a_4 & a_1 & -a_2 \end{array}\right]_{\otimes(2n)} P_2 = \left[\begin{array}{cc|cc} a_2 & a_1 & a_4 & -a_3 \\ a_4 & a_3 & a_2 & -a_1 \end{array}\right]_{\otimes(2n)}.$$

We can apply Lemmas 2.5 and 2.6 to this case by interchanging the roles of a_4 and a_2 with those of a_3 and a_1, respectively. It follows then that there exists a nonsingular matrix X such that $A_4^TX = A_2^TXD$ and $A_3^TX = A_1^TX\bar{D}$, where

$$D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{2n}) = \text{diag}\left(\frac{15-7\sqrt{3}}{-15-7\sqrt{3}}, \frac{9-9\sqrt{3}}{9+9\sqrt{3}}, \alpha_1^+, \alpha_1^-, \ldots, \alpha_{n-1}^+, \alpha_{n-1}^-\right),$$

$$\bar{D} = \text{diag}(\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_{2n}) = \text{diag}\left(\frac{48-18\sqrt{3}}{-48-18\sqrt{3}}, \frac{18\sqrt{3}}{-18\sqrt{3}}, \beta_1^+, \beta_1^-, \ldots, \beta_{n-1}^+, \beta_{n-1}^-\right),$$

$$\begin{aligned} \alpha_k^\pm &= \tfrac{3\sqrt{3}\pm\sqrt{43+40\cos\theta_k-2\cos^2\theta_k}}{(-28-16\sqrt{3})+(\sqrt{3}+1)\cos\theta_k}, \\ \beta_k^\pm &= \tfrac{(37+8\cos\theta_k)\pm3\sqrt{3}\sqrt{43+40\cos\theta_k-2\cos^2\theta_k}}{(-64-36\sqrt{3})+(19+9\sqrt{3})\cos\theta_k}, \\ \theta_k &= \tfrac{k\pi}{n}. \end{aligned}$$

Combining the above results with (2.42) and following the analysis of Section 2.4.2, we conclude that

$$\begin{aligned} \sigma(J') &= \{\pm\bar{\lambda}_1, \ldots, \pm\bar{\lambda}_{2n}\} \\ &\cup \left\{\cup_{k=1}^{n-1}\{\mu|\mu^2 - (\bar{\lambda}_i - \lambda_i)\mu\cos\tfrac{k\pi}{n} - \bar{\lambda}_i\lambda_i = 0,\ i = 1, 2, \ldots, 2n\}\right\}. \end{aligned} \tag{2.43}$$

It is clear from the analytic expression for $\sigma(J')$ that all the eigenvalues of J except ±1, which are simple ones, have magnitudes less than 1. Therefore $\rho(J') = 1$ and $\text{index}(I - J') = 1$ (i.e., $\text{rank}(I - J)^2 = \text{rank}(I - J')$). This, together with the block 2-cyclic nature of J', implies that we can apply the analysis in [13] and of Section 2.4.1.2 to obtain the optimal SOR method for n even and a very nearly optimal one for n odd by means of the formulas (2.40). Note that b is exactly the same as in the Dirichlet case while $a = \bar{\lambda}_3$.

2.5 Numerical Results In this section we present some numerical results to confirm some of the formulas and the convergence behavior of various iterative methods considered in this chapter. We also compare the time and space performance of optimal SOR, LINPACK BAND GE, and GMRES software for solving the INTCOL and HERMCOL equations. All numerical computations were carried out on a Sun 4/470 with 32Mbytes of main memory in double precision. The execution times measured are given in seconds and the space is measured in words.

First, we attempt to confirm numerically the formulas (2.36) and (2.43). For this we choose $n = m = 3$ and find the eigenvalues of the block Jacobi iteration matrices J_2 and J' by using the subroutine EVLRG from IMSL/MATH library. The eigenvalues are presented in Tables 2.1 and 2.2, respectively. They agree with the ones obtained from the formulas (2.36) and (2.43) at least up to the the number of the decimal digits displayed in these tables.

TABLE 2.1

The 36 eigenvalues of the Jacobi matrix J_2 for $n = m = 3$.

±0.5726	±0.3272	±0.3169	±0.2411	±0.2136	±0.1741
±0.1238	±0.0858	±0.0718	±0.0718	±0.0526	±0.0499
±0.0374	±0.0263	±0.0260	±0.0123	±0.0079	±0.0014

TABLE 2.2

The 36 eigenvalues of the Jacobi matrix J' for $n = m = 3$.

±1.000	±0.753	±0.732	±0.573	±0.401	±0.366
±0.327	±0.317	±0.214	±0.212	±0.179	±0.126
±0.058	±0.037	±0.026	±0.012	±0.001	±0.001

Second, we verify some of the convergence results obtained in this paper. For this we apply the INTCOL and HERMCOL algorithms from the ELLPACK system [33] to discretize several PDE problems on the unit square. For the solution of these equations we have developed three new solution modules in ELLPACK based on block AOR, SOR and adaptive SOR methods (cf. [15]) and new indexing modules based on the tensor-product ordering. Depending on the initial value of ω_0 selected for the adaptive SOR, we introduce the following notation: SOR_1 if $\omega_0 = 1$, SOR_2 if ω_0 is equal to the optimal ω for a model problem, and SOR_3 if ω_0 is the final adaptive ω found by solving the same problem on a coarser mesh unless $n = m = 2$ in which case we take $\omega_0 = 1.0$. Throughout, we denote the semi-optimal SOR with ω the optimal value for a model problem by SOR_0. We have implemented the adaptive procedure used by the ITPACK routines [36]. For completeness, we note that the AOR method for the solution of $Ax = b$ is defined by

$$(D - rL)x_{n+1} = [(1 - \omega)D + (\omega - r)L + \omega U]x_n + \omega b,$$

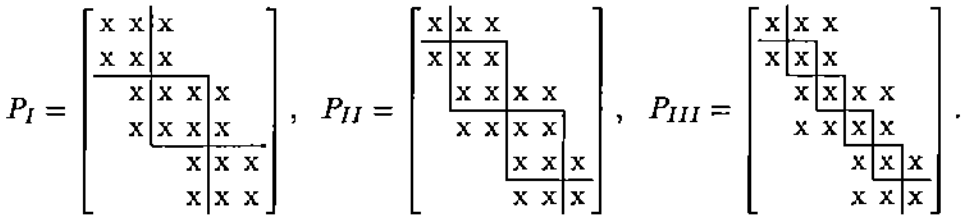


FIG. 2.4. *Displays three partitionings of the INTCOL equations associated with a uniform mesh of size $n = m = 3$. They are denoted by P_I, P_{II}, and P_{III} where each* x *denotes a $2m \times 2m$ matrix and has the same structure as the global one.*

assuming the splitting $A = D - L - U$. Its convergence properties depend on the choice of the pair of parameters (ω, r) [12]. The pairs (1,0), (1,1) and (ω,ω) yield the Jacobi, Gauss-Seidel and SOR methods while the pairs $(\omega, 0)$, $(\omega, 1)$ and (ω, r) with $r \neq 0$ give their extrapolated counterparts. For comparison purposes we use AOR with $(\omega, r) = (0.5, 1.0)$ pair of parameters to solve the INTCOL equations. This is the optimal AOR method used by Papatheodorou in [30].

The iterative solvers implemented depend on the block partitioning of the collocation coefficient matrix. In this study we consider three different matrix partitionings depicted in Figure 2.4 for a specific mesh size $n = m = 3$.

The efficiency of the block iterative methods depends on the time required to solve the linear subsystems $D_i x = b$, where D_i is the ith block diagonal element of A. In general we expect the bandwidth of the matrices D_i to be small. However, for the block partitionings in Figure 2.4 the upper and lower bandwidth of some D_i's is $(2n+2)$. For these D_i's, instead of solving the corresponding linear subsystem $D_i x = b$ directly, we solve the transformed system $PD_iP^{-1}y = Pb$ where $y = Px$, and $P = [e_1, e_{n+1}, e_2, e_{n+2}, \ldots, e_n, e_{2n}]$, with e_i being the standard unit vectors. Figure 2.5 depicts the effect of this transformation for a 3×3 mesh. It is easy to show that the bandwidth of PD_iP^{-1} is only 5. Thus the transformed diagonal subsystem can be solved much faster using BAND GE without pivoting.

```
dxx   xxx              dxxxxx
xdx   xxx              xdxxxx
 xdxx  xxxx            xxdxxx
 xxdx  xxxx            xxxdxx
   xdx   xxx             xxdxxxxx
   xxd   xxx             xxxdxxxx
xxx   dxx                xxxxdxxx
xxx   xdx                xxxxxdxx
 xxxx  xdxx                  xxdxxx
 xxxx  xxdx                  xxxdxx
   xxx   xdx                 xxxxdx
   xxx   xxd                 xxxxxd
```

FIG. 2.5. *Illustrates the PD_iP^{-1} transformation for a 3×3 mesh.*

In the tables below we display the maximum discretization error $\|u - u_h\|_\infty$ based on a 65×65 grid , where u is the exact solution of the PDE problem and u_h is the computed

Hermite cubic piecewise polynomial solution. In order to compare the efficiency among various iterative solvers considered, we used the same stopping criterion, namely

$$\frac{||x_{n+1} - x_n||_\infty}{||x_{n+1}||_\infty} < \epsilon = 5 * 10^{-6},$$

and the same initial solution x_0.

Tables 2.3a and 2.4a indicate the convergence of four block iterative methods applied to the system of INTCOL equations corresponding to different mesh sizes. The AOR implemented is based on the partitioning P_I while the rest of the block methods (i.e Jacobi, Gauss-Seidel, and SOR) are based on the partitioning P_{II} of the collocation matrix. The optimal parameters of AOR used are $(\omega, r) = (0.5, 1.0)$ according to the analysis in [30]. The optimal SOR parameter ω_{opt} was obtained based on the analysis presented in Section 2.4. The data in these tables suggest that the block SOR has the largest asymptotic rate of convergence.

Tables 2.4b and 2.3b depict the convergence behavior of three of the four iterative methods considered in Tables 2.3a and 2.4a for the HERMCOL equations. AOR (0.5,1.0) is not efficient for these type of equations. In this case all methods were implemented based on the block partitioning P_I of the HERMCOL coefficient matrix. It is worth noticing that the spectral analysis of the Jacobi iteration matrix for INTCOL and HERMCOL equations has shown that the ω_{opt} is the same for both cases. The data in these tables suggest that the block SOR has the fastest convergence.

Tables 2.4c and 2.3c depict the convergence data (number of iterations and discretization error) of optimal SOR and adaptive SOR_3 for both INTCOL and HERMCOL equations. These data suggest that the adaptive SOR behaves almost as the optimal SOR for the two model problems considered for relative coarse meshes.

Table 2.5 depicts the time and memory complexity of optimal SOR, the LINPACK BAND GE with partial pivoting, and GMRES (generalized minimal residual) [34] under three different preconditioners to solve the INTCOL equations associated with a model problem under different mesh sizes.

In the case of SOR and GMRES the initial guess of the solution corresponding to an $n \times n$ mesh is estimated from the previous collocation approximation based on an $(n/2) \times (n/2)$ mesh. Throughout we refer to it as the multigrid type initialization. The execution times of iterative methods include the total time to estimate the initial guess. The direct solver is applied to the system obtained using the natural ordering while the block SOR utilizes the mentioned above transformations to diagonal subsystems. These subsystems were solved using BAND GE *without* pivoting. It should be added that in general BAND GE with partial pivoting is necessary to solve the general collocation systems.

Among CG preconditioning iterative solvers GMRES method is recommended for nonsymmetric systems provided a good preconditioner is available. In these experiments we consider right preconditioning, which are simply the block diagonal matrices associated with the block matrices P_I, P_{II} and P_{III} of the collocation matrix A. We refer to them as PREC1, PREC2 and PREC3. The GMRES procedure is restarted every 50 steps and the stopping criterion is set to be

$$\frac{||b - Ax_n||_2}{||b - Ax_0||_2} < \epsilon = 5 * 10^{-5}.$$

The data suggest that the iterative methods have much smaller memory requirements. This of course was expected. However, we were surprised by the time efficiency of the optimal SOR that is better than the rest of the solvers considered and occurs at a level of relatively coarse meshes. In the case of GMRES, the preconditioner based on the block diagonal matrix corresponding to P_{II} block structure is the best performing.

Table 2.6 indicates the performance of SOR (ω takes the optimal values for the Dirichlet model problem in Table 2.3), adaptive SOR_3, BAND GE, and GMRES (restarted every 50 steps) for solving the INTCOL equations obtained from the discretization of a general elliptic PDE with Dirichlet boundary conditions on the unit square. All applied solvers were based on P_{II} block structure. The multigrid type approach was used to start the iterations. The data displayed include maximum discretization error and execution times. The data indicate that the semi-optimal SOR is the fastest for fine meshes without effecting the discretization error. Adaptive SOR_3 appears to effect the discretization error.

Table 2.7 compares the convergence and efficiency of the semi-optimal SOR and the three adaptive SOR methods considered in this section under different initial approximations x_0. It is clear that the multigrid initialization is the best based on the number of SOR iterations required to achieve the pre-defined tolerance. Among the adaptive SORs considered SOR_2 behaves closest to the semi-optimal one. This is due to the fact that they use almost the same ω.

Table 2.8 compares the performance and convergence behavior of optimal SOR, adaptive SOR_3, BAND GE, and GMRES(50) for model problem with Neumann (Tables 2.8a and 2.8b) and uncoupled boundary conditions (Table 2.8c). The exact solution is the one used in Table 2.3. Again, from the data including number of iterations required to achieve tolerance, maximum discretization error, the exact and estimated value of the SOR parameter ω used and execution times, we observe that optimal SOR outperforms the rest of methods with GMRES(50) being the slowest. In this table, all iterative solvers used multigrid type initialization. Moreover, it is noticed that the BAND GE could not run for mesh size 128 × 128 on the machine used due to memory limitations.

TABLE 2.3

The convergence behavior of four block iterative methods for solving the INTCOL and HERMCOL equations obtained by discretizing the equation $u_{xx} + u_{yy} = f$ with Dirichlet boundary condition ($u = g$). The functions f and g are selected so that $u(x,y) = \phi(x)\phi(y)$, where $\phi(x) = 0$, if $x \leq 0.35$, or if $x \geq 0.65$, otherwise $\phi(x)$ is a quintic polynomial determined so that it has two continuous derivatives.

INTCOL									
mesh	AOR(0.5,1.0)		Jacobi		Gauss-Seidel		Optimal SOR		
size	iter	error	iter	error	iter	error	ω_{opt}	iter	error
2 ×2	17	1.21	9	1.21	6	1.21	1.0314	6	1.21
4 ×4	17	1.28e-2	29	1.28e-1	15	1.28e-1	1.1786	9	1.28e-1
8 ×8	41	7.56e-2	94	7.55e-2	48	7.56e-2	1.4271	19	7.56e-2
16 ×16	200	2.59e-2	305	2.63e-2	154	2.62e-2	1.6536	40	2.59e-2

(a)

HERMCOL							
mesh	Jacobi		Gauss-Seidel		Optimal SOR		
size	iter	error	iter	erSOR	ω_{opt}	iter	error
2 X 2	12	1.19	7	1.19	1.0314	6	1.19
4 ×4	32	1.28e-1	18	1.28e-1	1.1786	11	1.28e-1
8 ×8	104	7.56e-2	56	7.56e-2	1.4271	21	7.57e-2
16 ×16	344	2.63e-2	182	2.61e-2	1.6536	46	2.59e-2

(b)

	INTCOL						HERMCOL				
mesh	Optimal SOR			Adaptive SOR_3			Optimal SOR		Adaptive SOR_3		
size	ω_{opt}	iter	error	ω	iter	error	iter	error	ω	iter	error
2 ×2	1.0314	6	1.21	1.0131	6	1.21	6	1.19	1.0176	7	1.19
4 ×4	1.1786	9	1.28e-1	1.0131	15	1.28e-1	11	1.12e-1	1.0176	15	1.28e-1
8 ×8	1.4271	19	7.57e-2	1.2685	32	7.56e-2	21	7.57e-2	1.2829	31	7.57e-2
16 ×16	1.6536	40	2.59e-2	1.5821	59	2.59e-2	46	2.59e-2	1.6528	68	2.59e-2

(c)

TABLE 2.4

The convergence behavior of four block iterative methods for solving the INTCOL and HERMCOL equations obtained by discretizing the equation $u_{xx} + u_{yy} = f$ with Dirichlet boundary condition $(u = 0)$. The function f is selected so that $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$.

INTCOL									
mesh	AOR (0.5,1.0)		Jacobi		Gauss-Seidel		Optimal SOR		
size	iter	error	iter	error	iter	error	ω_{opt}	iter	error
2 ×2	18	2.91e-1	11	2.9e-1	6	2.91e-1	1.0314	5	2.91e-1
4 ×4	19	1.46e-1	35	1.46e-1	20	1.46e-1	1.1786	11	1.46e-1
8 ×8	76	1.56e-2	131	1.56e-2	68	1.56e-2	1.4271	22	1.56e-2
16 ×16	247	6.08e-4	385	6.31e-4	199	6.28e-4	1.6536	43	6.08e-4

(a)

HERMCOL							
mesh	Jacobi		Gauss-Seidel		Optimal SOR		
size	iter	error	iter	error	ω_{opt}	iter	error
2 ×2	12	2.91e-1	7	2.91e-1	1.0314	6	2.91e-1
4 ×4	36	1.46e-1	20	1.46e-1	1.1786	11	1.46e-1
8 ×8	129	1.56e-2	69	1.56e-2	1.4271	24	1.56e-2
16 ×16	376	6.37e-4	200	6.28e-4	1.6536	47	6.08e-4

(b)

	INTCOL						HERMCOL				
mesh	Optimal SOR			Adaptive SOR$_3$			Optimal SOR		Adaptive SOR$_3$		
size	ω_{opt}	iter	error	ω	iter	error	iter	error	ω	iter	error
2 ×2	1.0314	5	2.91e-1	1.0178	6	2.91e-1	6	2.91e-1	1.0287	7	2.91e-1
4 ×4	1.1786	11	1.46e-1	1.0178	19	1.46e-1	11	1.46e-1	1.0287	19	1.46e-1
8 ×8	1.4271	22	1.56e-2	1.3761	33	1.56e-2	24	1.56e-2	1.3605	35	1.56e-2
16 ×16	1.6536	43	6.08e-4	1.3761	98	6.16e-4	47	6.08e-4	1.3605	101	6.17e-4

(c)

Table 2.5

The time and memory complexity of five solvers for solving the discrete equations obtained by applying INTCOL procedure to the equation $u_{xx} + u_{yy} = f$ with Dirichlet boundary conditions. The function f is selected so that $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$.

		Optimal SOR				BAND GE		
mesh	equations	time	iter	workspace	error	time	workspace	error
2 ×2	16	0.02	5	264	2.905e-1	0.02	464	2.905e-1
4 ×4	64	0.14	10	1136	1.456e-1	0.07	2624	1.456e-1
8 ×8	256	1.02	19	4704	1.563e-2	0.53	16640	1.563e-2
16 ×16	1024	6.22	27	19136	6.083e-4	5.03	115712	6.082e-4
32 ×32	4096	50.35	57	77184	5.795e-5	60.77	856064	5.795e-5
64 ×64	16384	360.28	99	310016	2.035e-6	797.75	6569984	2.035e-6
128 ×128	65536	3031.63	213	1242624	1.263e-7	NA	NA	NA

(a)

GMRES (restarted every 50 steps)								
			PREC1		PREC2		PREC3	
mesh	equations	error[1]	time	iter	time	iter	time	iter
2 ×2	16	2.905e-1	0.02	3	0.03	6	0.03	7
4 ×4	64	1.456e-1	0.20	15	0.16	10	0.24	18
8 ×8	256	1.563e-2	1.87	28	1.15	18	1.80	28
16 ×16	1024	6.082e-4	19.52	64	9.56	33	14.89	48
32 ×32	4096	5.766e-5	108.25	79	48.96	36	83.65	66
64 ×64	16384	2.056e-6	1255.03	244	371.66	66	559.91	107
128 ×128	65536	1.1400e-7	9134.46	400[2]	2571.77	106	5685.47	282

(b)

[1] Approximately the same error is found by using any of the three preconditioners as long as the same stopping criterion is satisfied.

[2] At this step the stopping criterion was not satisfied. The corresponding error was 1.18e-7

Table 2.6

The performance and convergence data of SOR$_0$, *Adaptive* SOR$_3$, BAND GE, *and* GMRES*(50) for solving the INTCOL equations obtained from the discretization of the equation $[2 + (y-1)e^{-y^4}]u_{xx} + [1 + \frac{1}{(1+4x^2)}]u_{yy} + 5[x(x-1) + (y-0.3)(y-0.7)]u = f$, with boundary conditions $(u = g)$. The functions f and g are selected so that $u(x,y) = \frac{x+y^2}{1+2x} + (1+x)(y-1)e^{-y^4} + 5(x+y)cos(xy)$.*

mesh	BAND GE		Adaptive SOR $_3$		SOR$_0$		GMRES(50)	
size	time	error	time	error	time	error	time	error
2 ×2	0.05	7.67e-3	0.03	7.67e-3	0.0	7.67e-3	0.02	7.67e-3
4 ×4	0.25	1.57e-3	0.17	1.57e-3	0.12	1.57e-3	0.15	1.57e-3
8 ×8	1.80	1.24e-4	0.84	1.25e-4	0.67	1.24e-4	0.97	1.24e-4
16 ×16	15.95	8.61e-6	3.05	1.24e-5	4.59	8.62e-6	8.07	8.61e-6
32 ×32	66.21	6.06e-7	12.15	9.30e-6	31.58	6.06e-7	70.23	6.06e-7
64 ×64	849.99	4.35e-9	56.58	8.92e-6	216.13	8.58e-9	466.88	1.26e-8

TABLE 2.7

The performance and convergence data of SOR *and the three adaptive* SORs *for solving the INTCOL equations obtained from the discretization of the PDE problem used in the previous table.*

	x_0 estimated by the 2x2 solution found by BAND GE						
mesh	SOR_0			Adaptive SOR_1			
size	iter	error	time	iter	error	time	ω
4 ×4	8	1.57e-3	0.08	10	1.57e-3	0.12	1.1934
8 ×8	14	1.24e-4	0.63	23	1.25e-4	1.03	1.2278
16 ×16	29	8.63e-6	5.42	39	1.37e-5	7.20	1.6173
32 ×32	66	6.07e-7	49.38	75	2.50e-5	56.2	1.7896
64 ×64	175	7.66e-8	521.97	153	3.09e-4	464.0	1.7796

	$x_0 = [0.5, 0.5, ..., 0.5]^T$						
mesh	SOR_0			Adaptive SOR_3			
size	iter	error	time	iter	error	time	ω
2 ×2	6	7.67e-3	0.02	7	7.67e-3	0.02	1.0246
4 ×4	11	1.57e-3	0.12	20	1.57e-3	0.20	1.0246
8 ×8	22	1.24e-4	1.03	32	1.24e-4	1.42	1.3898
16 ×16	48	8.62e-6	9.30	54	8.57e-6	10.03	1.6735
32 ×32	110	6.06e-7	84.65	104	1.91e-5	81.18	1.8000
64 ×64	283	4.25e-8	854.1	396	2.57e-4	1196.4	1.8000

Multigrid type initialization									
mesh	SOR_0			Adaptive SOR_3			Adaptive SOR_2		
size	iter	error	ω	iter	error	ω	iter	error	ω
2 ×2	6	7.67e-3	1.0314	7	7.67e-3	1.024	6	7.67e-3	1.0314
4 ×4	8	1.57e-3	1.1786	11	1.57e-3	1.024	8	1.57e-3	1.1786
8 ×8	12	1.24e-4	1.4271	13	1.25e-4	1.3165	12	1.24e-4	1.4271
16 ×16	19	8.62e-6	1.6536	9	1.24e-5	1.3165	17	8.62e-6	1.600
32 ×32	32	6.06e-7	1.8054	8	9.30e-6	1.3165	15	7.37e-7	1.600
64 ×64	51	8.58e-9	1.8907	6	8.92e-6	1.3165	12	5.48e-7	1.600

TABLE 2.8

The performance and convergence data of the optimal SOR, *adaptive* SORs, GMRES*(50) and the* BAND GE *for the solution of INTCOL equations obtained from the discretization of the PDE $u_{xx} + u_{yy} = f$ with Neumann boundary conditions (Tables a and b) and uncoupled mixed boundary conditions (Table c).*

	Optimal SOR				Adaptive SOR$_3$				BAND GE	
mesh	ω_{opt}	time	iter	error	ω	time	iter	error	time	error
2 ×2	1.2926	0.03	10	2.48	1.091	0.02	9	2.48	0.02	2.48
4 ×4	1.3042	0.18	13	3.22e-1	1.091	0.25	23	3.22e-1	0.07	3.22e-1
8 ×8	1.5498	1.17	22	1.40e-1	1.436	1.64	31	1.40e-1	0.52	1.40e-1
16 ×16	1.7392	9.56	46	4.76e-2	1.704	12.21	58	4.69e-2	5.01	4.76e-2
32 ×32	1.8550	79.09	94	1.40e-2	1.800	82.21	94	1.15e-2	58.03	1.40e-2
64 ×64	1.9153	664.84	197	2.18e-3	1.600	467.76	125	6.21e-3	797.97	2.20e-3
128 ×128	1.9413	7746.36	599	8.05e-4	1.800	1584.63	77	5.24e-3	NA	NA

(a)

mesh	Optimal SOR		Adaptive SOR$_2$				GMRES(50)		
size	iter	error	ω	iter	error	time	iter	error	time
2 ×2	10	2.48	1.2926	10	2.48	0.02	7	2.48	0.02
4 ×4	13	3.22e-1	1.3042	13	3.22e-1	0.17	12	3.22e-1	0.17
8 ×8	22	1.40e-1	1.5498	22	1.40e-1	1.17	19	1.40e-1	1.21
16 ×16	46	4.76e-2	1.600	55	4.63e-2	11.39	35	4.76e-2	10.23
32 ×32	94	1.40e-2	1.800	93	1.14e-2	80.49	91	1.40e-2	113.95
64 ×64	197	2.18e-3	1.600	125	6.15e-3	481.64	194	2.19e-3	1188.78
128 ×128	599	8.05e-4	1.800	77	5.19e-3	1592.89	684	8.02e-4	14505.91

(b)

with boundary condition $u = g_1$ at $x = 0$ or $y = 1$ and $u_n = g_2$ at $x = 1$ or $y = 0$										
	SOR$_0$				Adaptive SOR$_3$				BAND GE	
mesh	ω	time	iter	error	ω	time	iter	error	time	error
2 ×2	1.162	0.02	9	1.22	1.150	0.03	11	1.22	0.00	1.22
4 ×4	1.2414	0.27	26	1.31e-1	1.150	0.35	32	1.31e-1	0.07	1.31e-1
8 ×8	1.4885	1.64	31	7.38e-2	1.494	2.21	43	7.38e-2	0.53	7.40e-2
16 ×16	1.6964	12.48	60	2.60e-2	1.750	14.83	70	2.59e-2	5.02	2.57e-2
32 ×32	1.8304	68.5	75	7.78e-3	1.900	100.65	116	7.44e-3	59.15	7.28e-2
64 ×64	1.903	520.75	150	1.27e-3	1.600	437.67	108	2.52e-3	794.17	1.14e-3
128 ×128	1.9364	5773.44	434	4.35e-4	1.800	1601.95	81	2.07e-3	NA	NA

(c)

3. GENERAL INTERIOR HERMITE COLLOCATION METHODS FOR SECOND ORDER ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

In Chapter 2, we studied the iterative solution of the INTCOL and HERMCOL equations using the tensor-product ordering. However, the applicability of the INTCOL and HERMCOL algorithms is limited to PDEs defined on rectangular domains. For the case of general PDE domains, finding a method for the iterative solution of the discrete cubic Hermite collocation equations is still an *open problem*. In this chapter, first we extend the INTCOL algorithm for general rectilinear domains (by rectilinear we mean the boundaries are parallel to one of the axes). Throughout, we refer to it by the acronym GINCOL. Then we develop two indexing modules for the GINCOL algorithm. One is based on the finite-element ordering [43] and the other is based on the tensor-product ordering [30]. Using the tensor-product ordering, the linear system derived by the GINCOL algorithm generates the same block structure that is produced by the INTCOL algorithm. We experimentally explore the applicability and the convergence properties of the block iterative methods for GINCOL equations for some PDEs defined on an L-shaped domain and a more general rectilinear domain. Furthermore, the tensor-product ordering is successfully applied to the discrete equations produced by GENCOL together with the SOR and CG iterative solvers. A number of experiments were carried out to study the computational behavior of these iterative schemes and to estimate the various parameters involved.

The organization of this chapter is as follows. In Section 3.1, we formulate the GINCOL algorithm. In Section 3.2, two different indexing modules to be used with GINCOL are developed and one tensor-product ordering is introduced for the GENCOL algorithm. Finally, in Section 3.3, a wide class of PDE problems are solved by using the GINCOL algorithm with some block iterative linear solvers and a number of concluding remarks are made based on observations from these experiments.

3.1 GINCOL: The General Interior Collocation Method for a Rectilinear Domain The GENCOL method presented in Section 1.2.1 can be simplified in case (*i*) the domain Ω is rectilinear, and (*ii*) the problem has uncoupled boundary conditions, that is, at no point are the boundary conditions mixed, i.e.,

$$u \equiv \delta \quad on \quad \partial\Omega_1 \subset \partial\Omega,$$
$$\tfrac{\partial u}{\partial n} \equiv \delta \quad on \ \partial\Omega_2 = \partial\Omega - \partial\Omega_1 \subset \partial\Omega.$$

In order to distinguish this case from the general collocation method case, the simplified version is called general interior collocation (GINCOL). First, we use the algorithm in the

Section 1.2.1 to generate a finite-element mesh Ω_h. Then, since an entire boundary piece is either horizontal or vertical, some unknowns associated with nodes on a boundary piece can be determined beforehand using the following two assumptions:
(i) The boundary condition changes type only at a boundary node.
(ii) The boundary of the mesh Ω_h coincides with the boundary of the domain Ω.
The assumption (ii) is satisfied for the domain Ω when its boundary pieces are contained in the union of the grid lines of the mesh. So the user is simply required to place a grid line on each boundary piece of the domain Ω as part of the discretization. In this case, the boundary collocation equations can be solved explicitly when the discretization of the boundary conditions takes place. It is implemented by the code about the boundary discretization in subsection 1.2.2.

Since the boundary element e_b coincides with $e_b \cap \Omega$, we can simply select the four Gaussian points on each mesh element as the interior collocation points. Note that there are three unknowns associated with a concave corner of Ω and they have been solved for in the boundary discretization procedure. This makes the corresponding linear system over-determined. To derive a completely determined linear system, we allege that there is only one unknown solved at a non-convex corner during the boundary discretization procedure according to the following rule : *if (U solved) then the three unknowns are U_y, U_x and U_{xy} else the three unknowns are U, U_x and U_{xy}.* Finally, we are left with the task of generating $COEF$ and $IDCO$ and then eliminating the nonactive unknowns, namely those predetermined during the boundary discretization process from $BBBB$. There are three local two-dimensional arrays that are used for this task.

$NODELM(i,l) =$ the global index of the ith local node in element l
$INUNKN(i,n) =$ the global index of the ith local unknown associated with node n
$OLUNKN(i,n) =$ the value of the nonactive ith local unknown associated with node n

A code skeleton for this procedure is:

```
LOOP OVER ELEMENTS OF $\Omega_h$:
   GENERATE $NODELM$, $COEF$ and $BBBB$
   IF INTERIOR ELEMENT
     THEN GENERATE $INUNKN$ ASSOCIATED WITH THE
     LOWER LEFT NODE OF THIS ELEMENT
   ELSE GENERATE $INUNKN$ ASSOCIATED WITH THE LOWER
     LEFT NODE OF THIS ELEMENT AND $INUNKN$
     ASSOCIATED WITH OTHER NODES OF THIS ELEMENT
     ON THE BOUNDARY.
     FOR THE NONACTIVE UNKNOWN SET $INUNKN$ TO
     ZERO AND SUPPLY THE VALUE OF $OLUNKN$
   ENDIF
ENDLOOP;
GENERATE $IDCO$ AND MODIFY $BBBB$ BASED ON $NODELM$, $INUNKN$
AND $OLUNKN$
```

3.2 The Ordering of Unknowns and Equations In this section, we develop finite-element and tensor-product orderings for the GINCOL algorithm. Moreover, we introduce the tensor-product ordering for the GENCOL algorithm.

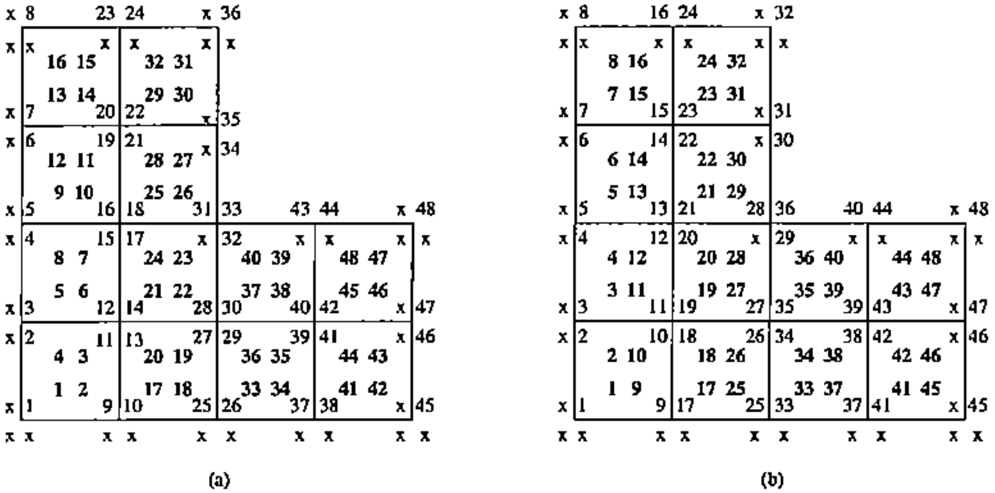


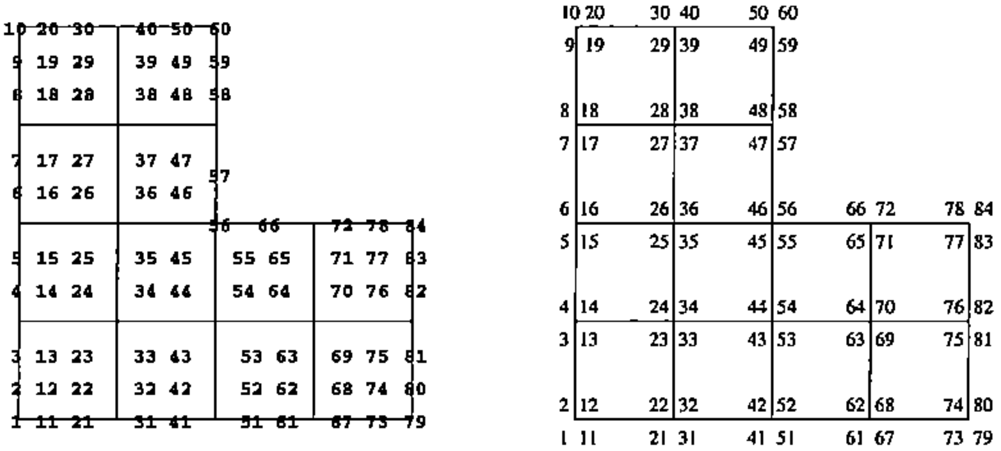
FIG. 3.1. *Two orderings of the collocation points and unknowns associated with GINCOL.*

The finite-element ordering for GINCOL equations is a straightforward extension of the ordering for INTCOL equations. So, we only illustrate this ordering in Figure 3.1(a) here.

The tensor-product ordering has been introduced for the INTCOL and HERMCOL algorithms. Here, we utilize it for the algorithms GINCOL and GENCOL. First, the GENCOL unknowns are split into two sets $\{u, u_y\}$ and $\{u_x, u_{xy}\}$. Then, on each x-grid line we number the unknowns $\{u, u_y\}$ node by node (south to north) followed by the numbering of $\{u_x, u_{xy}\}$ unknowns corresponding to the nodal points of the same grid line. For the tensor-product numbering of the GENCOL collocation points we consider the auxiliary exterior boundary collocation points introduced in [22] to determine the actual boundary points. By definition the auxiliary and interior collocation points are located on x-Gauss grid lines corresponding to x-coordinates of the Gauss points. Then, these points are numbered along the x-Gauss grid lines from south to north and west to east. The indices of the actual boundary collocation points and the auxiliary boundary points coincide. Figure 3.2 displays this scheme for an L-shaped region.

In the case of GINCOL, we have only interior collocation points, thus they are ordered from south to north along x-Gauss grid lines as in the case of GENCOL. Then the numbering of the active unknowns is determined by the indices of the interior collocation points as follows. At each nodal point, the active unknowns use the same index as the nearest interior collocation points. Figure 3.1(b) illustrates this ordering scheme for an L-shaped region.

The finite-element ordering is attractive because it yields a coefficient matrix which has smaller bandwidth than the one using the tensor-product ordering. The advantage of the tensor-product ordering is that the coefficient matrix for the GINCOL algorithm has



GENCOL

Fig. 3.2. *Tensor-product ordering of the collocation points and unknowns associated with GENCOL.*

the block structure indicated in Figure 1.5. It is worth noticing that for this case the x's denote submatrices of various orders. The coefficient matrix corresponding to GENCOL algorthim using tensor-product ordering is also a block matrix. However, its structure depends very much on the placement of the boundary collocation points. Figure 3.3(b) shows the detailed structure of the coefficient matrix for GINCOL in the case of the L-shaped domain of Figure 3.1. For GINCOL the diagonal blocks of the coefficient matrix is always a band matrix with bandwidth 2 and non-zero diagonal elements. Some block iterative linear solvers may benefit from this property.

Figure 3.3(a) shows the detailed structure of the coefficient matrix for GINCOL in the case of the L-shaped domain of Figure 3.1. The finite-element ordering provides the efficiency of bandedness but the presence of many zeros on the diagonal of the coefficient matrix prevents most iterative methods from being applied. So, the most reliable and preferable way to solve the linear system is to use Gauss elimination with scaling and partial pivoting [9]. However, direct methods tend to require much more memory as well as more time and their parallelization is difficult. It is very desirable to have a suitable iterative solver for the collocation equations in general, this can be accomplished by using the tensor-product ordering.

3.3 Application of Iterative Linear Solvers In this section, we use the algorithm GINCOL developed in the previous section to discretize a number of elliptic PDEs with uncoupled boundary conditions on an L-shaped domain Ω_1 as well as on a general rectilinear domain Ω_2 shown in Figure 3.4. We consider only the tensor-product ordering as the finite-element ordering prevents us from applying an iterative linear solver.

For the iterative solution of the GINCOL equations, we consider two approaches: the overrelaxation, AOR(SOR)-type, approach and the conjugate gradient, CG-type, approach.

Among the AOR-type methods, because of the presence of the block structure it is customary to use a block iterative method instead of point iteration. Three different block partitionings of the INTCOL coefficient matrix in Figure 2.4 are applied to the GINCOL coefficient matrix.

Among the CG-type methods, the preconditioned `GMRES` (generalized minimal residual) method [34] is an often successful method for solving nonsymmetric linear systems. The preconditioner used should be easily inverted and the diagonal blocks of P_I, P_{II} and P_{III} can be used. After experimentation we conclude that P_{II} preconditioner is the best for `GMRES`.

TABLE 3.1

The convergence behavior of block iterative methods for solving the GINCOL linear system obtained by discretizing the equation $u_{xx} + u_{yy} = f$ in Ω_1 with Dirichlet boundary condition $(u = g)$. The functions f and g are selected so that $u(x, y) = e^{x+y}$.

mesh	P_I								P_{III}		
size	*AOR*		adaptive SOR			SOR			SOR		
(neqn)	iter	error	$\omega(1.0)$	iter	error	ω	iter	error	ω	iter	error
4 ×4 (48)	24	1.91e-5	0.8285	41	2.04e-5	0.7537	16	2.00e-5	0.5	24	2.52e-5
8 ×8 (192)	68	1.18e-5	0.8285	58	8.20e-6	0.7374	48	8.73e-6	0.5	60	2.29e-5
16 ×16 (768)	225	4.62e-5	0.8285	219	3.46e-6	0.7334	191	7.63e-6	0.5	242	3.30e-5

mesh	P_{II}								
size	adaptive SOR			adaptive SOR			SOR		
(neqn)	$\omega(1.0)$	iter	error	$\omega(1.01)$	iter	error	ω	iter	error
4 ×4 (48)	0.8285	41	2.04e-5	1.6880	14	2.13e-5	1.1786	9	2.06e-5
8 ×8 (192)	0.8285	58	8.20e-6	1.7998	28	7.75e-6	1.4271	19	1.43e-6
16 ×16 (768)	0.8285	219	3.46e-6	1.7070	60[1]	8.34e-6	1.6536	45	8.34e-7

[1] At this step the stopping criterion is not satisfied. The corresponding iteration error is 6.35e-5.

TABLE 3.2

The convergence behavior of block iterative methods for solving the linear system obtained by discretizing the equation $u_{xx} + u_{yy} = f$ in Ω_2 with Dirichlet boundary condition $(u = g)$. The functions f and g are selected so that $u(x, y) = e^{x+y}$.

mesh	P_I								P_{III}		
size	AOR		adaptive SOR			SOR			SOR		
(neqn)	iter	error	$\omega(1.0)$	iter	error	ω	iter	error	ω	iter	error
8 ×8 (188)	49	1.22e-5	0.8284	40	3.52e-6	0.7374	36	2.86e-6	0.5	45	5.72e-6
16 ×16 (752)	179	1.35e-5	0.8285	156	1.97e-6	0.7334	138	5.42e-6	0.5	175	2.90e-5

mesh	P_{II}								
size	adaptive SOR			adaptive SOR			SOR		
(neqn)	$\omega(1.0)$	iter	error	$\omega(1.01)$	iter	error	ω	iter	error
8 ×8 (188)	0.8284	32	1.39e-5	1.7901	20	3.58e-6	1.4271	19	1.907e-6
16 ×16 (752)	0.8284	124	2.53e-5	1.6669	45	1.68e-6	1.6536	41	2.38e-6

In Tables 3.1 to 3.3, we study the convergence behavior of SOR under different block partitionings of the GINCOL collocation matrix in different rectilinear domains. Specifically, we display the maximum discretization error $||u - u_h||_\infty$ based on the grid points inside the domain, where u is the exact solution of the PDE problem and u_h is the computed cubic Hermite piecewise polynomial solution. These tables also give the number of iterations required for the various methods to converge. These numbers are good indicators of the actual efficiencies of the methods. The mesh size entry is the size of the mesh in the smallest rectangle that contains the domain. The values in parentheses beside them are the orders of the linear systems. For the adaptive SOR method, we also display the final value of ω used; the initial guess of ω is given in the heading. In order to compare the efficiency among the various iterative solvers, we use the stopping criterion, namely, $\frac{||x_{n+1}-x_n||_2}{||x_{n+1}||_2} < \epsilon$ for SOR and $\frac{||b-Ax_n||_2}{||b-Ax_0||_2} < \epsilon$ for GMRES with the same initial solution $x_0 = [0.5, 0.5, \ldots, 0.5]^T$.

In the iterative computation, one wants the error in solving the linear system to be less than the discretization error in approximating the PDE. In all tables the convergence tolerance $\epsilon = 10^{-5}$ is used for SOR and $\epsilon = 10^{-6}$ for GMRES. As the data in Tables 3.1 and 3.2 indicate, this tolerance is too large as the discretization error on the coarsest mesh is already about 2×10^{-5} for the first example and even less for the second one. Nevertheless, these data clearly show that all these iteration methods converge for the test cases used. For the non-adaptive SOR, the relaxation parameter ω is the optimal ω value corresponding to the case of the same problem defined on the smallest rectangle containing Ω. The AOR method used here is the one used in [30].

TABLE 3.3

The convergence behavior of block iterative methods for solving the linear system obtained by discretizing the equation $u_{xx} + u_{yy} = f$ in Ω_2 with Dirichlet boundary condition ($u = g$). The functions f and g are selected so that $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$.

mcsh	P_{II}								
size	adaptive SOR			adaptive SOR			SOR		
(neqn)	$\omega(1.0)$	iter	error	$\omega(1.01)$	iter	error	ω	iter	error
8 ×8 (188)	0.8284	30	7.89e-2	1.90	52	7.89e-2	1.4271	21	7.89e-2
16 ×16 (752)	0.8284	91	2.03e-2	1.9	69	2.03e-2	1.6536	44	2.03e-2
32 ×32 (3008)	0.8284	243	5.54e-4	1.8701	70	5.68e-4	1.8054	92	5.68e-4

The fewest iterations by a factor 3 to 5 are required using the P_{II} preconditioning and an SOR iteration with relaxation parameter near the usual 1.8 value. The adaptive SOR can locate a "locally optimum" parameter less than 1 which provides performance similar to that using the other preconditioners. These data suggest that this iteration approach has the promise to become an efficient and robust solver for the GINCOL collocation equations.

In Tables 3.4 to 3.6, we measure the computational complexity of the GMRES and SOR iterative schemes for solving the GINCOL equations and compare them with BAND GE direct solver [33]. BAND GE is applied with partial pivoting and "natural ordering" of the equations

TABLE 3.4

The performance data of some solvers for solving the discrete equations obtained by applying GINCOL algorithm to the equation $u_{xx} + u_{yy} = f$ with Dirichlet boundary conditions in domain Ω_2. The function f is selected so that $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$.

		BAND GE		GMRES(50)			SOR			
mesh	neqn	error	time	error	iter	time	error	iter	time	ω
8x8	188	7.89e-2	0.21	7.89e-2	16	0.40	7.89e-2	17	0.8	1.1786
16x16	752	2.03e-2	5.75	2.03e-2	51	6.23	2.03e-2	38	4.37	1.4271
32x32	3008	5.68e-4	22.65	5.61e-4	58	27.97	5.68e-4	71	28.63	1.6536
64x64	12032	3.03e-5	279.60	7.27e-5	122	242.97	3.09e-5	145	233.93	1.8054

and unknowns. The iterative solver is used to solve the linear system using tensor-product ordering. The data indicate that iterative solvers are more efficient for fine grids and produce solutions with the same level of discretization error. Furthermore, the convergence behavior of GMRES and SOR does not depend on the PDE operators considered in these experiments. For example, in the case of the SOR method the same ω values were used for a model problem and a general one. Finally, Table 3.6 shows the application of the iterative schemes to the solution of the GENCOL equations using the tensor-product ordering. The PDE problem used here is defined on a rectangle, thus the optimal value of ω can be found in Chapter 2 for SOR. In this case we see that the iterative schemes are becoming more efficient than direct solvers even for coarse meshes.

Additional preliminary experiments indicate the GMRES is an efficient alternative to BAND GE for the solution of GENCOL equations with tensor-product ordering obtained from the discretization of PDE problems defined on general domains. All results indicate that SOR is applicable to solve the GINCOL equations with tensor-product ordering, at least for rectilinear domains. The extension of GINCOL to general domains is part of our ongoing research efforts.

TABLE 3.5

The performance data of some solvers for solving the discrete equations obtained by applying GINCOL algorithm to the equation $[2+(y-1)e^{-y^4}]u_{xx}+[1+\frac{1}{(1+4x^2)}]u_{yy}+5[x(x-1)+(y-0.3)(y-0.7)]u=f$ *with Dirichlet boundary conditions in domain* Ω_2. *The function* f *is selected so that* $u(x,y)=10\phi(x)\phi(y)$, *where* $\phi(x)=e^{-100(x-0.1)^2}(x^2-x)$.

	BAND GE		GMRES(50)			SOR			
mesh	error	time	error	iter	time	error	iter	time	ω
8x8	8.65e-2	0.23	8.65e-2	16	0.40	8.65e-2	17	0.8	1.1786
16x16	2.05e-2	2.12	2.05e-2	51	6.18	2.05e-2	39	4.22	1.4271
32x32	5.68e-4	21.57	5.45e-4	63	29.75	5.68e-4	72	28.87	1.6536
64x64	2.98e-5	266.83	9.05e-5	127	250.35	3.17e-5	150	240.70	1.8054

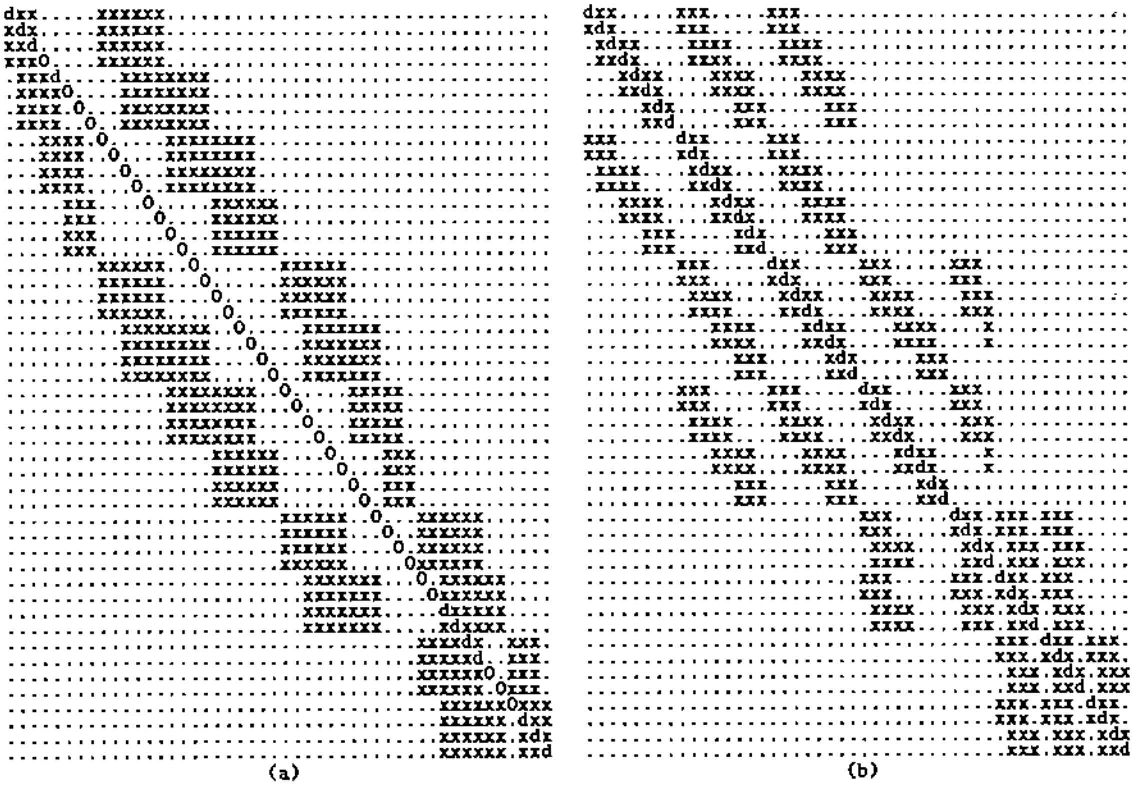


FIG. 3.3. *The detailed structure of the collocation matrices derived from GINCOL using the finite-element and tensor-product ordering, respectively, for the model problem on the L-shaped domain.*

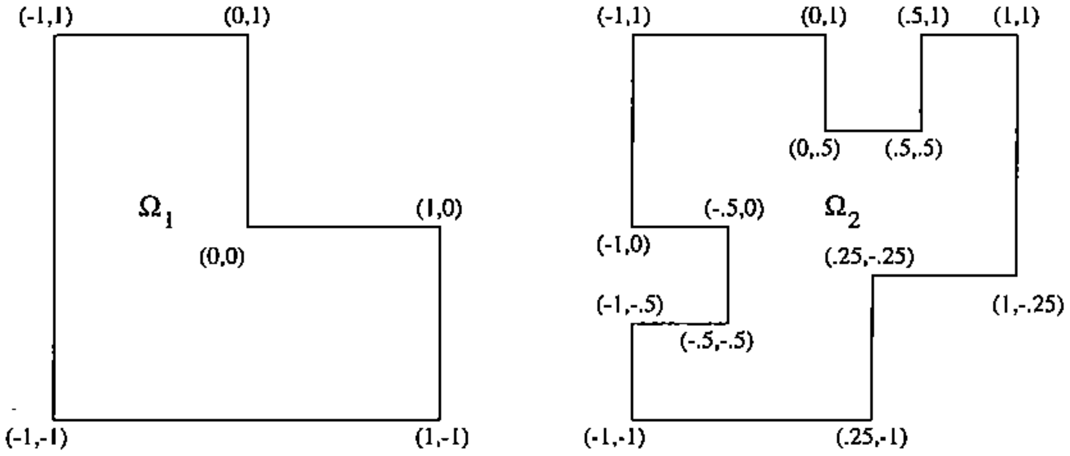


FIG. 3.4. *The domains used in the computational experiments.*

TABLE 3.6

The performance data of some solvers for solving the discrete equations obtained by applying GENCOL and GINCOL procedures to the equation $u_{xx} + u_{yy} = f$ with Dirichlet boundary conditions on the rectangle $(-1,1)\times(-1,1)$. The function f is selected so that $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}\,(x^2-x)$.

GENCOL									
		BAND GE		GMRES(50)			Opt SOR		
mesh	neqn	error	time	error	iter	time	error	iter	time
2x2	36	2.99e-1	0.04	2.99e-1	8	0.05	2.99e-1	6	0.48
4x4	100	8.45e-1	0.39	8.45e-1	13	0.17	8.45e-1	10	0.93
8x8	324	1.34e-1	0.83	1.34e-1	36	4.30	1.34e-1	23	1.52
16x16	1156	2.33e-2	8.55	2.33e-2	53	9.883	2.33e-2	47	7.72
32x32	4356	5.68e-4	104.98	5.69e-4	73	49.95	5.69e-4	99	57.05
64x64	16900	2.91e-5	968.83	3.35e-5	191	589.633	3.09e-5	284	625.88

GINCOL									
		BAND GE		GMRES(50)			Opt SOR		
mesh	neqn	error	time	error	iter	time	error	iter	time
2x2	16	2.99e-1	0.02	2.99e-1	7	0.02	2.99e-1	6	0.48
4x4	64	8.45e-1	0.07	8.45e-1	12	0.08	8.45e-1	10	0.65
8x8	256	1.34e-1	0.40	1.34e-1	20	0.68	1.34e-1	22	1.15
16x16	1024	2.33e-2	8.88	2.33e-2	51	19.7	2.33e-2	43	6.30
32x32	4096	5.68e-4	41.68	5.86e-4	66	54.417	5.69e-4	92	49.3
64x64	16384	2.91e-5	648.75	5.98e-5	186	498.35	3.09e-5	246	540.93

4. A GENERALIZED SCHWARZ SPLITTING METHOD BASED ON HERMITE COLLOCATION FOR ELLIPTIC BOUNDARY VALUE PROBLEMS

The Schwarz Alternating Method (SAM) coupled with various numerical discretization schemes has been already established as an efficient alternative for solving differential equations on various parallel machines. In this chapter we consider an extension (Generalized Schwarz Splitting-GSS) of SAM for solving elliptic boundary value problems with generalized interface conditions that depend on a parameter that might differ in each overlapping region [38]. The work in [3, 40] and the results of Chapters 2 and 3 motivate us to study the convergence properties of GSS associated with the cubic Hermite collocation discretization technique [22]. Following the work in [38] and [26], we explore this problem at the matrix equation level formulation. More specifically, we study the iterative solution of the corresponding enhanced GSS collocation discrete matrix equation for a model elliptic boundary value problem.

This chapter is organized as follows. In Section 4.1, we give a brief description of the GSS on a rectangle at functional and matrix levels. In Section 4.2, first we define a matrix with a specific structure and then we investigate some basic properties associated with it. Using the results obtained, we derive the block Jacobi iteration matrix corresponding to applying the GSS with bicubic Hermite collocation discretization for the solution of the Poisson equation under Dirichlet boundary conditions on a rectangular domain split into overlapping stripes. In Section 4.3, we carry out a spectral analysis of the enhanced block Jacobi iteration matrix corresponding to the one-dimensional problem. Furthermore, we determine the domain of convergence and find a subinterval of it in which the optimal parameter for the one-parameter GSS case lies; moreover, we obtain sets of optimal parameters for the multi-parameter GSS case. In Section 4.4, we analyze the convergence properties of the one-parameter GSS case for the two-dimensional problem. Finally, in Section 4.5, we present a number of numerical examples in the one- and two-dimensional spaces that verify the theoretical results obtained in this chapter. In addition, we compare the convergence rates of the SAM and GSS methods with minimum and maximum overlap and draw several conclusions.

4.1 A Generalized Schwarz Alternating Method We consider the Dirichlet problem

$$\left\{ \begin{array}{lcl} Lu & = & f \;\; in \;\; \Omega, \\ u & = & g \;\; on \;\; \partial\Omega \end{array} \right. \tag{4.1}$$

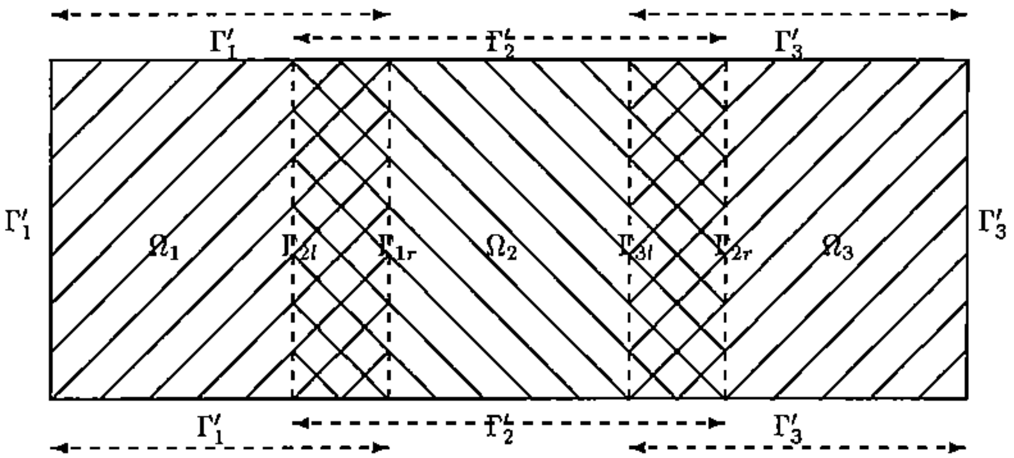


FIG. 4.1. *A decomposition of* Ω *for* $k = 3$

where L is a second-order linear elliptic partial differential operator, Ω is a rectangle $(a, b) \times (c, d) \in R^2$ and $\partial\Omega$ is its boundary.

In order to formulate the GSS for PDE problem (4.1), we decompose Ω into k overlapping rectangles (stripes) $\Omega_1, \ldots, \Omega_k$, defined as $\Omega_i = (t_{il}, t_{ir}) \times (c, d)$ with $a = t_{1l} < t_{2l} < \ldots < t_{kl} < b$ and $a < t_{1r} < t_{2r} < \ldots < t_{kr} = b$. Furthermore, for $k \geq 3$ we assume that $t_{2l} < t_{1r}$ and $t_{(i-2)r} < t_{il} < t_{(i-1)r}$ for $i = 3, \ldots, k$. This assumption guarantees that no three consecutive stripes can have a common overlapping area and that any two consecutive stripes do overlap. We set $\Gamma_{il} = \{t_{il}\} \times (c, d)$ and $\Gamma_{ir} = \{t_{ir}\} \times (c, d)$, and assume that both sets Γ_{1l} and Γ_{kr} are empty. We also define $\Gamma'_i = \partial\Omega_i - (\Gamma_{il} \cup \Gamma_{ir})$. An example of such a decomposition for $k = 3$ is depicted in Figure 4.1.

Then, the Generalized Schwarz Splitting method applied to problem (4.1), with a domain splitting as above, consists of solving the k coupled subproblems

$$\left\{ \begin{array}{rcll} L(u_i(x)) & = & f(x), & x \in \Omega_i, \\ u_i(x) & = & g(x), & x \in \Gamma'_i, \\ \omega_l u_i(x) + (1 - \omega_l)\frac{\partial u_i(x)}{\partial n} & = & \omega_l u_{i-1}(x) + (1 - \omega_l)\frac{\partial u_{i-1}(x)}{\partial n}, & x \in \Gamma_{il}, \\ \omega_r u_i(x) + (1 - \omega_r)\frac{\partial u_i(x)}{\partial n} & = & \omega_r u_{i+1}(x) + (1 - \omega_r)\frac{\partial u_{i+1}(x)}{\partial n}, & x \in \Gamma_{ir}, \end{array} \right. \tag{4.2}$$

for $i = 1, \ldots, k$, where the ω's are user defined parameters.

Problem (4.2) can be solved iteratively for a given initial guess $(u_1^{(0)}, \ldots, u_k^{(0)})$. Following, we illustrate the application of Gauss-Seidel type iteration for the GSS PDE subproblems:

$$\left\{ \begin{array}{rcll} L(u_i^{(j)}(x)) & = & f(x), & x \in \Omega_i, \\ u_i^{(j)}(x) & = & g(x), & x \in \Gamma'_i, \\ \omega_l u_i^{(j)}(x) + (1 - \omega_l)\frac{\partial u_i^{(j)}(x)}{\partial n} & = & \omega_l u_{i-1}^{(j)}(x) + (1 - \omega_l)\frac{\partial u_{i-1}^{(j)}(x)}{\partial n}, & x \in \Gamma_{il}, \\ \omega_r u_i^{(j)}(x) + (1 - \omega_r)\frac{\partial u_i^{(j)}(x)}{\partial n} & = & \omega_r u_{i+1}^{(j-1)}(x) + (1 - \omega_r)\frac{\partial u_{i+1}^{(j-1)}(x)}{\partial n}, & x \in \Gamma_{ir}, \end{array} \right. \tag{4.3}$$

where $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots$. There are many ways of implementing a discrete analog of the algorithm (4.3). This is due to the many choices for the parameter ω and to the many alternatives of the discretization technique to be selected for each subproblem.

If the discretization scheme used to solve the subproblems in (4.2) is the same as the one used for the solution of the original problem (4.1) , then it is easy to see that problem (4.1) is reduced to the solution of a linear system, say $Ay = f$,

$$\begin{bmatrix} A_{11} & A_{12} & & & \\ A_{21} & A_{22} & A_{23} & & \\ & A_{32} & A_{33} & A_{34} & \\ & & A_{43} & A_{44} & A_{45} \\ & & & A_{54} & A_{55} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

while problem (4.2) is reduced to solving the larger system (enhanced) $\bar{A}\bar{y} = \bar{f}$

$$\begin{bmatrix} A_{11} & A_{12} & & & & & & \\ A'_{21} & B'_{22} & C'_{22} & A'_{23} & & & & \\ A''_{21} & B''_{22} & C''_{22} & A''_{23} & & & & \\ & & A_{32} & A_{33} & A_{34} & & & \\ & & & A'_{43} & B'_{44} & C'_{44} & A'_{45} & \\ & & & A''_{43} & B''_{44} & C''_{44} & A_{45} & \\ & & & & & A_{54} & A_{55} & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y'_2 \\ y_3 \\ y_4 \\ y'_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f'_2 \\ f''_2 \\ f_3 \\ f'_4 \\ f''_4 \\ f_5 \end{bmatrix} \tag{4.4}$$

where $\bar{A}$ is a $k \times k$ block tridiagonal matrix. In both cases, we assume that the unknowns have been decomposed according to the splitting in Figure 4.1. Notice that, corresponding to the overlapping region $\Omega_1 \cap \Omega_2$, we have $f'_2 = \begin{bmatrix} f_2 \\ 0 \end{bmatrix}$, $f''_2 = \begin{bmatrix} 0 \\ f_2 \end{bmatrix}$, $A'_{21} = \begin{bmatrix} A_{21} \\ 0 \end{bmatrix}$, $A''_{23} = \begin{bmatrix} 0 \\ A_{23} \end{bmatrix}$, $A''_{21} = \begin{bmatrix} 0 \\ A_{21} \end{bmatrix}$, $A''_{23} = \begin{bmatrix} 0 \\ A_{23} \end{bmatrix}$, $B'_{22} = \begin{bmatrix} A_{22} \\ E_1 \end{bmatrix}$, $C'_{22} = \begin{bmatrix} 0 \\ -E_2 \end{bmatrix}$, $B''_{22} = \begin{bmatrix} -E_2 \\ 0 \end{bmatrix}$ and $C''_{22} = \begin{bmatrix} E_2 \\ A_{22} \end{bmatrix}$ with $E_1 = [0, 0, \ldots, 0, h_1(\omega_r), h_2(\omega_r)]$, $E_2 = [h_1(\omega_l), h_2(\omega_l), 0, \ldots, 0]$ and 0 being zero vectors or submatrices, where h_i's are vectors derived from the interface boundary conditions. Similar relations hold for the equations associated with the overlapping region $\Omega_2 \cap \Omega_3$.

In view of the way algorithm (4.3) is derived, it is apparent that in order to study the convergence properties for a given discrete implementation of it, it suffices to study the corresponding properties of the block Jacobi iteration matrix associated with the enhanced linear system (4.4). On the other hand, one should bear in mind that for different implementations of the algorithm (4.3) the convergence properties of the corresponding iterative methods based on the linear system (4.4) may be different for the same problem. So, one may not have a single block Jacobi matrix to study for the different implementations of the algorithm (4.3). To simplify the subsequent discussion, we shall confine ourselves to selecting the cubic Hermite collocation discretization technique to discretize all the subproblems. For this specific implementation, we shall derive the corresponding block Jacobi iteration matrix for a model problem and shall study the impact of the various choices of the parameter ω, subject among others to the restriction $\omega_l = \omega_r$, on the spectral radius of the Jacobi matrix. For this study we will exploit some basic properties of a specific matrix structure in the section that follows.

4.2 Spectral Analysis of the Block Jacobi Iteration Matrices In this section, we define a set of matrices which share a particular structure, study their properties, and develop the preliminaries needed for the rest of the analysis. Then we will use the results we shall obtain to derive the block Jacobi iteration matrix corresponding to a GSS scheme with bicubic Hermite collocation discretization technique for a model problem in the two-dimensional space. It is worth noticing that the analysis and the results of this section can also be used to handle the one-dimensional problem.

4.2.1 Preliminaries First we define a square matrix $T(m, n, \alpha_1, \beta_1, \alpha_2, \beta_2)$ of order $4mn$ such that

$$\begin{bmatrix}
\alpha_1A_1+\beta_1A_2 & A_3 & -A_4 & & & & & & \\
\alpha_1A_3+\beta_1A_4 & A_1 & -A_2 & & & & & & \\
 & A_1 & A_2 & A_3 & -A_4 & & & & \\
 & A_3 & A_4 & A_1 & -A_2 & & & & \\
 & & & & \ddots & & & & \\
 & & & & A_1 & A_2 & A_3 & -A_4 & \\
 & & & & A_3 & A_4 & A_1 & -A_2 & \\
 & & & & & & A_1 & A_2 & \alpha_2A_3-\beta_2A_4 \\
 & & & & & & A_3 & A_4 & \alpha_2A_1-\beta_2A_2
\end{bmatrix}, \tag{4.5}$$

where each A_i, $i = 1, 2, 3, 4$, is a square matrix of order $2m$ and $\alpha_1, \beta_1, \alpha_2, \beta_2$ are scalars. For simplicity, we denote it by T in the remainder of this section.

Next, we introduce the two matrices

$$N = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \text{ and } M = \begin{bmatrix} A_3 & -A_4 \\ A_1 & -A_2 \end{bmatrix}. \tag{4.6}$$

We assume that N is nonsingular and its inverse is written in the same block form as N, namely $N^{-1} = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}$. Then, it follows from the obvious relation $N = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} M \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$, where I denotes the identity matrix of order $2m$, that $M^{-1} = \begin{bmatrix} B_2 & B_1 \\ -B_4 & -B_3 \end{bmatrix}$. Based on the material introduced so far we can state and prove the following statement.

LEMMA 4.1. *If the matrices $-\beta_1B_1 + \alpha_1B_3$ and $\beta_2B_1 + \alpha_2B_3$ are invertible, then the following relations hold*

$$T\begin{bmatrix} C_1 \\ (-N^{-1}M)^{n-1}\begin{bmatrix}\alpha_2I \\ \beta_2I\end{bmatrix} \\ \vdots \\ (-N^{-1}M)\begin{bmatrix}\alpha_2I \\ \beta_2I\end{bmatrix} \\ I \end{bmatrix} = \begin{bmatrix} S_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; T\begin{bmatrix} I \\ (-M^{-1}N)\begin{bmatrix}\alpha_1I \\ \beta_1I\end{bmatrix} \\ \vdots \\ (-M^{-1}N)^{n-1}\begin{bmatrix}\alpha_1I \\ \beta_1I\end{bmatrix} \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ S_2 \end{bmatrix} \tag{4.7}$$

where

$$\begin{cases} S_1 = (-\beta_1B_1 + \alpha_1B_3)^{-1}([\beta_1I, -\alpha_1I](-N^{-1}M)^n\begin{bmatrix}\alpha_2I \\ \beta_2I\end{bmatrix}), \\ S_2 = (-\beta_2B_1 - \alpha_2B_3)^{-1}([\beta_2I, -\alpha_2I](-M^{-1}N)^n\begin{bmatrix}\alpha_1I \\ \beta_1I\end{bmatrix}) \end{cases} \tag{4.8}$$

with C_1 and C_2 being matrices of order $2m$ that can be uniquely determined.

Proof. It is sufficient to show the first part of (4.7), since the second part follows by a similar argument. It is trivial to show, using (4.5) and (4.6), that the last $2n-2$ block elements of both sides of (4.7) are equal. To determine S_1, we use the first two blocks of both sides to get

$$N\begin{bmatrix}\alpha_1 I\\ \beta_1 I\end{bmatrix}C_1 + M(-N^{-1}M)^{n-1}\begin{bmatrix}\alpha_2 I\\ \beta_2 I\end{bmatrix} = \begin{bmatrix}S_1\\ 0\end{bmatrix}. \tag{4.9}$$

Then, premultiplying both members of (4.9) by $[\beta_1 I, -\alpha_1 I]N^{-1}$ we obtain

$$[\beta_1 I, -\alpha_1 I](-N^{-1}M)^n\begin{bmatrix}\alpha_2 I\\ \beta_2 I\end{bmatrix} = (-\beta_1 B_1 + \alpha_1 B_3)S_1$$

which determines S_1. C_1 can be determined uniquely from (4.9) by premultiplying it first by N^{-1} and then solving for C_1 from either the first or the second block component of the resulting equation. Since C_1 is not used later, its explicit expression is not needed. □

Now, let us assume that T and $\alpha_1 A_3 + \beta_1 A_4$ are nonsingular and V_1 and V_2 denote the submatrices of the last $2n-1$ block components of the matrix products $T^{-1}[I,0,\ldots,0,0]^T$ and $T^{-1}[0,I,0,\ldots,0]^T$ respectively. Then, since $T^{-1}T$ is the identity matrix, we obtain $V_1(\alpha_1 A_1 + \beta_1 A_2) + V_2(\alpha_1 A_3 + \beta_1 A_4) = 0$. This implies $V_2 = -V_1(\alpha_1 A_1 + \beta_1 A_2)(\alpha_1 A_3 + \beta_1 A_4)^{-1}$. Hence, the matrix of the last $2n-1$ block components of the matrix product $T^{-1}[A_1^T, A_3^T, 0\ldots,0]^T$ is

$$V_1(A_1 - (\alpha_1 A_1 + \beta_1 A_2)(\alpha_1 A_3 + \beta_1 A_4)^{-1}A_3). \tag{4.10}$$

To simplify the expression above, we state and show the following lemma.

LEMMA 4.2. *If both A_1 and A_3 are nonsingular, then*

$$(\beta_1 B_1 - \alpha_1 B_3)(A_1 - (\alpha_1 A_1 + \beta_1 A_2)(\alpha_1 A_3 + \beta_1 A_4)^{-1}A_3) = \beta_1 I$$

Proof. We have

$$\begin{aligned}
&A_1 - (\alpha_1 A_1 + \beta_1 A_2)(\alpha_1 A_3 + \beta_1 A_4)^{-1}A_3\\
=\ &A_1(I - (\alpha_1 I + \beta_1 A_1^{-1}A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1})\\
=\ &A_1(\alpha_1 I + \beta_1 A_3^{-1}A_4 - \alpha_1 I - \beta_1 A_1^{-1}A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}\\
=\ &\beta_1 A_1(A_3^{-1}A_4 - A_1^{-1}A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}
\end{aligned}$$

and the fact that $N^{-1}N = I_{4m}$ implies $\begin{bmatrix}B_1 & B_2\\ B_3 & B_4\end{bmatrix}\begin{bmatrix}A_1 & A_2\\ A_3 & A_4\end{bmatrix} = \begin{bmatrix}I & 0\\ 0 & I\end{bmatrix}$. Thus, the following relations hold

$$\begin{aligned}
&(\beta_1 B_1 - \alpha_1 B_3)(A_1 - (\alpha_1 A_1 + \beta_1 A_2)(\alpha_1 A_3 + \beta_1 A_4)^{-1}A_3)\\
=\ &(\beta_1 B_1 - \alpha_1 B_3)\beta_1 A_1(A_3^{-1}A_4 - A_1^{-1}A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}\\
=\ &\beta_1(\beta_1 B_1 A_1 A_3^{-1}A_4 - \beta_1 B_1 A_2 - \alpha_1 B_3 A_1 A_3^{-1}A_4 + \alpha_1 B_3 A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}\\
=\ &\beta_1(\beta_1(I - B_2 A_3)A_3^{-1}A_4 - \beta_1 B_1 A_2 + \alpha_1 B_4 A_3 A_3^{-1}A_4 + \alpha_1 B_3 A_2)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}\\
=\ &\beta_1(\beta_1 A_3^{-1}A_4 - \beta_1(B_2 A_4 + B_1 A_2) + \alpha_1(B_4 A_4 + B_3 A_2))(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1}\\
=\ &\beta_1(\alpha_1 I + \beta_1 A_3^{-1}A_4)(\alpha_1 I + \beta_1 A_3^{-1}A_4)^{-1} = \beta_1 I.\ \square
\end{aligned}$$

Now, combining Lemma 4.2 with the expression in (4.10), we can easily show the first relation of the Lemma 4.3. Similarly, using the second equality of (4.7) we can derive the second relation of the same lemma.

LEMMA 4.3. *Let the assumptions of Lemma 4.1 hold and the matrices T, A_1, A_3 S_1, and S_2 be invertible. Then, we have*

$$T^{-1}\begin{bmatrix} A_1 \\ A_3 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_1' \\ (-N^{-1}M)^{n-1}\begin{bmatrix}\alpha_2 I \\ \beta_2 I\end{bmatrix} \\ \vdots \\ (-N^{-1}M)\begin{bmatrix}\alpha_2 I \\ \beta_2 I\end{bmatrix} \\ I \end{bmatrix} (-\beta_1)([\beta_1 I, -\alpha_1 I](-N^{-1}M)^n \begin{bmatrix}\alpha_2 I \\ \beta_2 I\end{bmatrix})^{-1}$$

and

$$T^{-1}\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ A_3 \\ A_1 \end{bmatrix} = \begin{bmatrix} I \\ (-M^{-1}N)\begin{bmatrix}\alpha_1 I \\ \beta_1 I\end{bmatrix} \\ \vdots \\ (-M^{-1}N)^{n-1}\begin{bmatrix}\alpha_1 I \\ \beta_1 I\end{bmatrix} \\ C_2' \end{bmatrix} (-\beta_2)([\beta_2 I, -\alpha_2 I](-M^{-1}N)^n \begin{bmatrix}\alpha_1 I \\ \beta_1 I\end{bmatrix})^{-1}$$

where C_1' and C_2' are matrices of order $2m$ that can be uniquely determined.

4.2.2 Derivation of the Block Jacobi Iteration Matrix In this section we consider the Dirichlet problem for Poisson equation on the rectangular domain Ω and the Ω splitting defined in Section 4.1. We use the bicubic Hermite collocation technique to discretize the corresponding continuous GSS PDE subproblems. To simplify the discussion, in the sequel, we use a uniform mesh with $m+1$ y-grid points and $l+1$ x-grid points for each subdomain. Moreover, it is assumed that the overlaps $\Omega_i \cap \Omega_{i+1}$, $i = 1, \ldots, k-1$ are of equal size with (l_0+1) x-grid points in each of them, $h = h_x = \frac{d-c}{m} = \frac{b-a}{n} = h_y$ and $n = lk - (k-1)l_0$. In order to make the entries of the collocation coefficient matrix independent of the mesh size h, the basis functions for the standard bicubic Hermite collocation are modified as in [30], and instead of imposing the interface boundary conditions

$$\begin{cases} \omega_l u_i(x) + (1-\omega_l)\frac{\partial u_i(x)}{\partial n} = \omega_l u_{i-1}(x) + (1-\omega_l)\frac{\partial u_{i-1}(x)}{\partial n}, & x \in \Gamma_{il}, \\ \omega_r u_i(x) + (1-\omega_r)\frac{\partial u_i(x)}{\partial n} = \omega_r u_{i+1}(x) + (1-\omega_r)\frac{\partial u_{i+1}(x)}{\partial n}, & x \in \Gamma_{ir}, \end{cases}$$

we impose

$$\begin{cases} \omega_l' u_i(x) + (1-\omega_l')h\frac{\partial u_i(x)}{\partial n} = \omega_l' u_{i-1}(x) + (1-\omega_l')h\frac{\partial u_{i-1}(x)}{\partial n}, & x \in \Gamma_{il}, \\ \omega_r' u_i(x) + (1-\omega_r')h\frac{\partial u_i(x)}{\partial n} = \omega_r' u_{i+1}(x) + (1-\omega_r')h\frac{\partial u_{i+1}(x)}{\partial n}, & x \in \Gamma_{ir}. \end{cases}$$

It is worth noticing that

$$\omega_l = \frac{\omega_l'}{h - \omega_l' h + \omega_l'} \quad \text{and} \quad \omega_r = \frac{\omega_r'}{h - \omega_r' h + \omega_r'}.$$

To form the corresponding linear system we use Papatheodorou's tensor-product ordering (see also [30]) to order the unknowns and the equations. Therefore, the original problem (without applying the GSS scheme) leads to the solution of the linear system $Ay = f$ with the unknown and the right hand side vector being $[y_1^T, y_2^T, \ldots, y_{2n}^T]^T$ and $[f_1^T, f_2^T, \ldots, f_{2n}^T]^T$ respectively, where y_i and f_i are vectors of length $2m$. More specifically, the components of y_{2n} and y_{2i-1}, $i = 1, \ldots, n$, are the approximate values of u and $h\frac{\partial u}{\partial y}$ at the nodes on the corresponding x-grid line while y_1 and y_{2i}, $i = 1, \ldots, n-1$, are the approximate values of $h\frac{\partial u}{\partial x}$ and $h^2\frac{\partial^2 u}{\partial x \partial y}$ at the nodes on the corresponding x-grid line. The enhanced linear system (4.4), $\bar{A}\bar{y} = \bar{f}$, after eliminating the unknowns associated with the values of u_i and $\frac{\partial u_i}{\partial y}$ on the interface boundaries by using the interface boundary conditions from each subproblem, is expressed in a block form as follows

$$\begin{bmatrix} D_1 & U & & & \\ L & D_2 & U & & \\ & \ddots & \ddots & \ddots & \\ & & L & D_2 & U \\ & & & L & D_k \end{bmatrix} \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \vdots \\ \bar{y}_k \end{bmatrix} = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \vdots \\ \vdots \\ \bar{f}_k \end{bmatrix}. \qquad (4.11)$$

In (4.11),
$D_1 = T(m, l, 0, 1, \frac{\omega_r'-1}{\omega_r'}, 1)$, $D_2 = T(m, l, \frac{1-\omega_l'}{\omega_l'}, 1, \frac{\omega_r'-1}{\omega_r'}, 1)$ and $D_k = T(m, l, \frac{1-\omega_l'}{\omega_l'}, 1, 0, 1)$, while U is a matrix of block order $2l$ with $A_3, A_1, \frac{1-\omega_r'}{\omega_r'}A_3, \frac{1-\omega_r'}{\omega_r'}A_1$ as its $(2l-1, 2l_0)$, $(2l, 2l_0)$, $(2l-1, 2l_0+1)$, $(2l, 2l_0+1)$ block elements and 0's elsewhere and L is a matrix of block order $2l$ with $A_1, A_3, \frac{\omega_l'-1}{\omega_l'}A_1, \frac{\omega_l'-1}{\omega_l'}A_3$ as its $(1, 2l-2l_0)$, $(2, 2l-2l_0)$, $(1, 2l-2l_0+1)$, $(2, 2l-2l_0+1)$ block elements and 0's elsewhere. $\bar{y}_i$ and $\bar{f}_i$ are vectors consisting of $4ml$ elements each and their relations to those of the original linear system are the following $\bar{y}_i = [\bar{y}_{2(i-1)(l-l_0)+1}^T, \ldots, \bar{y}_{2(i-1)(l-l_0)+2l_0-1}^T, y_{2(i-1)(l-l_0)+2l_0}^T, \ldots, y_{2(i-1)(l-l_0)+2l-1}^T, \bar{y}_{2(i-1)(l-l_0)+2l+1}^T]^T$ $\bar{f}_i = [f_{2(i-1)(l-l_0)+1}^T, \ldots, f_{2(i-1)(l-l_0)+2l}^T]^T$. It is worth noticing that it can be shown that $\tilde{y}_i = y_i$ if the matrix $\bar{A}$ on the left hand side of equation (4.11) is invertible (see, e.g., [37]).

Let J be the block Jacobi iteration matrix associated with the matrix coefficient $\bar{A}$ of (4.11). To simplify the notation, we assume that

$$\begin{array}{ll} D_1^{-1}[0, \ldots, 0, A_3^T, A_1^T]^T = [X_1^T, X_2^T, \ldots, X_{2l}^T]^T & D_2^{-1}[A_1^T, A_3^T, 0, \ldots, 0]^T = [Y_1^T, Y_2^T, \ldots, Y_{2l}^T]^T \\ D_2^{-1}[0, \ldots, 0, A_3^T, A_1^T]^T = [Z_1^T, Z_2^T, \ldots, Z_{2l}^T]^T & D_k^{-1}[A_1^T, A_3^T, 0, \ldots, 0]^T = [W_1^T, W_2^T, \ldots, W_{2l}^T]^T \end{array}$$

and introduce the new quantities

$$c_l = \frac{1-\omega_l'}{\omega_l'}, \quad c_r = \frac{1-\omega_r'}{\omega_r'}.$$

Then, it is easy to show that the spectrum $\sigma(J)$ of J satisfies $\sigma(J) = \sigma(J') \cup \{0\}$, where

$$J' = \begin{bmatrix} 0 & X & & & & & & \\ Y & 0 & 0 & Z & & & & \\ \bar{Y} & 0 & 0 & \bar{Z} & & & & \\ & & & \ddots & & & & \\ & & & Y & 0 & 0 & Z & \\ & & & \bar{Y} & 0 & 0 & \bar{Z} & \\ & & & & & & W & 0 \end{bmatrix}, \quad \begin{array}{l} X = \begin{bmatrix} X_{2l-2l_0} & c_r X_{2l-2l_0} \\ X_{2l-2l_0+1} & c_r X_{2l-2l_0+1} \end{bmatrix}, \\ \tilde{Y} = \begin{bmatrix} Y_{2l-2l_0} & -c_l Y_{2l-2l_0} \\ Y_{2l-2l_0+1} & -c_l Y_{2l-2l_0+1} \end{bmatrix}, \\ \tilde{Z} = \begin{bmatrix} Z_{2l-2l_0} & c_r Z_{2l-2l_0} \\ Z_{2l-2l_0+1} & c_r Z_{2l-2l_0+1} \end{bmatrix} \end{array}$$

$$Y = \begin{bmatrix} Y_{2l_0} & -c_l Y_{2l_0} \\ Y_{2l_0+1} & -c_l Y_{2l_0+1} \end{bmatrix}, Z = \begin{bmatrix} Z_{2l_0} & c_r Z_{2l_0} \\ Z_{2l_0+1} & c_r Z_{2l_0+1} \end{bmatrix}, W = \begin{bmatrix} W_{2l_0} & -c_l W_{2l_0} \\ W_{2l_0+1} & -c_l W_{2l_0+1} \end{bmatrix}.$$

In view of the structure of J', it is not difficult to see, through a similarity transformation using the matrix diag$(\begin{bmatrix} I & -c_l I \\ 0 & I \end{bmatrix}, \begin{bmatrix} I & c_r I \\ 0 & I \end{bmatrix}, \ldots)$, that $\sigma(J') = \sigma(J'') \cup \{0\}$, where J'' is of the same structure as J' with its entries being

$$\begin{array}{lll} X = X_{2l-2l_0} - c_l X_{2l-2l_0+1}, & Y = Y_{2l_0} + c_r Y_{2l_0+1}, & Z = Z_{2l_0} + c_r Z_{2l_0+1}, \\ \tilde{Y} = Y_{2l-2l_0} - c_l Y_{2l-2l_0+1}, & \tilde{Z} = Z_{2l-2l_0} - c_l Z_{2l-2l_0+1}, & W = W_{2l_0} + c_r W_{2l_0+1}. \end{array}$$

Applying now Lemma 4.3 we can obtain that

$$\begin{cases} X &= -[I, -c_l I](-M^{-1}N)^{l-l_0} \begin{bmatrix} 0 \\ I \end{bmatrix} ([I, c_r I](-M^{-1}N)^l \begin{bmatrix} 0 \\ I \end{bmatrix})^{-1}, \\ Y &= -[I, c_r I](-N^{-1}M)^{l-l_0} \begin{bmatrix} -c_r I \\ I \end{bmatrix} ([I, -c_l I](-N^{-1}M)^l \begin{bmatrix} -c_r I \\ I \end{bmatrix})^{-1}, \\ Z &= -[I, c_r I](-M^{-1}N)^{l_0} \begin{bmatrix} c_l I \\ I \end{bmatrix} ([I, c_r I](-M^{-1}N)^l \begin{bmatrix} c_l I \\ I \end{bmatrix})^{-1}, \\ \tilde{Y} &= -[I, -c_l I](-N^{-1}M)^{l_0} \begin{bmatrix} -c_r I \\ 1 \end{bmatrix} ([I, -c_l I](-N^{-1}M)^l \begin{bmatrix} -c_r I \\ I \end{bmatrix})^{-1}, \\ \bar{Z} &= -[I, -c_l I](-M^{-1}N)^{l-l_0} \begin{bmatrix} c_l I \\ 1 \end{bmatrix} ([I, c_r I](-M^{-1}N)^l \begin{bmatrix} c_l I \\ I \end{bmatrix})^{-1}, \\ W &= -[I, c_r I](-N^{-1}M)^{l-l_0} \begin{bmatrix} 0 \\ I \end{bmatrix} ([I, -c_l I](-N^{-1}M)^l \begin{bmatrix} 0 \\ I \end{bmatrix})^{-1}. \end{cases}$$

To simplify the notation further, we restrict ourselves to considering the case $c_r = c_l$. That is, we assume that the interface boundary conditions are of the same type. Then, using the fact that $(M^{-1}N) = \text{diag}(I, -I)(N^{-1}M)\text{diag}(I, -I)$, it is shown that $X = W$, $Y = \tilde{Z}$ and $Z = \tilde{Y}$. Consequently, we take that $\sigma(J) = \sigma(-G_k) \cup \{0\}$, where

$$G_k = \begin{bmatrix} 0 & X & & & & & & \\ Y & 0 & 0 & Z & & & & \\ Z & 0 & 0 & Y & & & & \\ & & & \ddots & & & & \\ & & & Y & 0 & 0 & Z & \\ & & & Z & 0 & 0 & Y & \\ & & & & & & X & 0 \end{bmatrix}, \tag{4.12}$$

$$\text{with}\begin{cases} X &= [I, c_r I](-N^{-1}M)^{l-l_0}\begin{bmatrix} 0 \\ I \end{bmatrix}([I, -c_r I](-N^{-1}M)^{l}\begin{bmatrix} 0 \\ I \end{bmatrix})^{-1}, \\ Y &= [I, c_r I](-N^{-1}M)^{l-l_0}\begin{bmatrix} -c_r I \\ I \end{bmatrix}([I, -c_r I](-N^{-1}M)^{l}\begin{bmatrix} -c_r I \\ I \end{bmatrix})^{-1}, \\ Z &= [I, -c_r I](-N^{-1}M)^{l_0}\begin{bmatrix} -c_r I \\ 1 \end{bmatrix}([I, -c_r I](-N^{-1}M)^{l}\begin{bmatrix} -c_r I \\ I \end{bmatrix})^{-1}. \end{cases}$$

Note that G_k is a $2(k-1) \times 2(k-1)$ block matrix.

4.3 One-Dimensional Case

4.3.1 The One-Parameter GSS First, we consider the case where ω_l and ω_r are the same in each overlapping region. In this section, we consider the GSS algorithm (4.3) together with the cubic Hermite collocation discretization scheme for the boundary value problem

$$\begin{cases} u_{xx} = f, \ a < x < b, \\ u(a) = g_a, \ u(b) = g_b. \end{cases}$$

For this problem, we have $A_1 = -2\sqrt{3}$, $A_2 = -1 - \sqrt{3}$, $A_3 = 2\sqrt{3}$ and $A_4 = -1 + \sqrt{3}$, where the A_i's are defined in Section 4.2.1. Since these entities are scalars and not matrices of order $2m$, we can now write $(-N^{-1}M)$ explicitly. Simple computations show that $(-N^{-1}M) = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$. In turn, this implies that $(-N^{-1}M)^j = \begin{bmatrix} 1 & -j \\ 0 & 1 \end{bmatrix}$. Therefore, after some simplification of the previously found expressions takes place we can obtain that for the case $c_r = c_l$

$$\sigma(J) = \sigma(-G_k) \cup \{0\},$$

where G_k is the following matrix of order $2(k-1)$

$$G_k = \begin{bmatrix} 0 & \frac{(l-l_0)-c_r}{l+c_r} & & & & & \\ \frac{l-l_0}{l+2c_r} & 0 & 0 & \frac{l_0+2c_r}{l+2c_r} & & & \\ \frac{l_0+2c_r}{l+2c_r} & 0 & 0 & \frac{l-l_0}{l+2c_r} & & & \\ & & & \ddots & & & \\ & & & \frac{l-l_0}{l+2c_r} & 0 & 0 & \frac{l_0+2c_r}{l+2c_r} \\ & & & \frac{l_0+2c_r}{l+2c_r} & 0 & 0 & \frac{l-l_0}{l+2c_r} \\ & & & & & \frac{(l-l_0)-c_r}{l+c_r} & 0 \end{bmatrix}.$$

From the expression above, it is readily observed that G_k is block 2-cyclic consistently ordered or weakly cyclic of index 2 [39} (see also [41] or [1]), therefore $\sigma(G_k) = \sigma(-G_k)$. It is worth mentioning that in [26] a matrix of precisely the same structure is considered and recurrence relationships to minimize the spectral radius of G_k are obtained. However, for the cases $k = 4$ and 5 the expressions that can be obtained are very difficult to handle, while for $k > 5$ we do not know how to slove the equations analytically. We have exactly the same situation. In the present work as in [26], it is shown that $\rho(G_k)$ can be made zero for $c_r = l - l_0$ and for the case $k = 2$ or $k = 3$. Thus, we have the theorem below.

THEOREM 4.1. *For $k = 2, 3$ and $c_r = l - l_0$, we have $\rho(J) = 0$.*

For the case $k > 3$, the analysis in [26] holds except that the expressions for the corresponding entries of the G_k matrix are different. However, in order to go a step further in the direction of determining the optimal value of c_r we shall focus on two issues: i) determine the interval of c_r for which the block Jacobi method converges and ii) determine a genuine subinterval of the interval in (i) in which the optimal c_r lies. For this we state and prove the following theorem.

THEOREM 4.2. *Under the assumptions made and the notation used so far the following relation holds*

$$\rho(J) = \rho(G_k) < 1$$

if and only if $c_r \in (-\frac{l_0}{2}, \infty)$. *Moreover, the minimum (optimal) value of* $\rho(J)$ *is attained at some* $c_r \in (l - l_0, \infty)$.

Proof. First, we consider the case $c_r < -l$. Since G_k is similar to

$$G'_k = \left[\begin{array}{ccccccc} 0 & \frac{(l-l_0)-c_r}{l+c_r} & & & & & \\ \frac{l-l_0}{l+2c_r} & 0 & 0 & -\frac{l_0+2c_r}{l+2c_r} & & & \\ -\frac{l_0+2c_r}{l+2c_r} & 0 & 0 & \frac{l-l_0}{l+2c_r} & & & \\ & & & \ddots & & & \\ & & & \frac{l-l_0}{l+2c_r} & 0 & 0 & -\frac{l_0+2c_r}{l+2c_r} \\ & & & -\frac{l_0+2c_r}{l+2c_r} & 0 & 0 & \frac{l-l_0}{l+2c_r} \\ & & & & & \frac{(l-l_0)-c_r}{l+c_r} & 0 \end{array}\right]$$

and $-G'_k$ is an irreducible nonnegative matrix with all nonzero entries strictly increasing with c_r in $(-\infty, -l)$, it follows that $\rho(G_k)$ is strictly increasing. Moreover, it is easy to show that $\lim_{c_r \to -\infty} \rho(G_k) = 1$. Consequently, we obtain $\rho(J) > 1$. In case $-l < c_r < -\frac{l}{2}$, we have

$$\begin{aligned} |\det(G_k)| &= |\det(G_k[e_2, e_1, e_4, e_3, \ldots, e_{2k-2}, e_{2k-3}])| \\ &= (\tfrac{l-l_0-c_r}{l+c_r})^2 |2\tfrac{l-l_0}{l+2c_r} - 1|^{k-2} > 1. \end{aligned}$$

Note that the inequality above is satisfied because $\frac{l-l_0-c_r}{l+c_r} > 1$ and $\frac{l-l_0}{l+2c_r} < 0$. Therefore, at least one of the eigenvalues of G_k must have modulus greater than 1, implying that $\rho(J) > 1$. In the case $-\frac{l}{2} < c_r \le -\frac{l_0}{2}$, G'_k is an irreducible nonnegative matrix with all nonzero entries strictly decreasing with c_r increasing. Specifically, we have $\lim_{c_r \to (-l_0/2)^-} \rho(G_k) = 1$. This implies $\rho(J) \ge 1$. For the case $c_r > -\frac{l_0}{2}$ and $c_r \ne l - l_0$ we have $\rho(J) < 1$ because G_k is irreducible and the absolute sum of the first row is less than $||G_k||_\infty = 1$. As for the specific case $c_r = l - l_0$, G_k is reducible, since its first and last rows are null vectors. However, after deleting the first and last two rows and columns, the reduced matrix is irreducible and its spectral radius is the same as that of G_k. Then, following the same arguments as previously, we obtain again $\rho(J) < 1$.

Coming to the second assertion of the present theorem, it is apparent that the minimum value of $\rho(J)$ is attained for some $c_r \in (-\frac{l_0}{2}, \infty)$. However, to obtain the genuine subinterval mentioned in the statement of the theorem a much deeper theoretical analysis, based on a number of other statements, is required. This analysis is presented in the next subsection. □

Note 1 We have $\omega_r = \frac{1}{1+c_r h}$, since $c_r = \frac{1-\omega_r'}{\omega_r'}$ and $\omega_r = \frac{\omega_r'}{h-\omega_r' h+\omega_r'}$. Thus the convergence interval in terms of ω $(= \omega_r = \omega_l)$ is $(0, \frac{2}{2-l_0 h}) \supset (0,1]$ and the optimum occurs for some ω in the interval $(0, \frac{1}{1+(l-l_0)h}) \subset (0,1)$. In addition, as $h \to 0^+$ the convergence interval tends to $(0,1]$ while the interval in which the optimum occurs tends to $(0,1)$.

Note 2 The problem of determining a "better" interval in which the optimum c_r lies than the one already obtained, i.e., $(l-l_0, \infty)$, is an open problem that is being investigated. However, a number of numerical experiments have shown that the value $c_r = l - l_0$ (i.e., $\omega = \frac{1}{1+(l-l_0)h}$) is a good approximation to the optimal value of c_r.

4.3.2 Appendix to Section 4.3.1 In this subsection we prove the second part of Theorem 4.2. This is accomplished after a number of statements presented as lemmas are proved.

LEMMA 4.4. *Let $\rho(B)$ be the spectral radius of any matrix B of even order. Then, we have* $\det(B - \lambda I) > 0$ *for all* $\lambda > \rho(B)$.

Proof. Let $p(\lambda) = \det(B - \lambda I)$. It is clear that $p(\lambda)$ is a monic polynomial. It then follows that $p(\lambda) \to \infty$ as $\lambda \to \infty$. Suppose that $det(B - \rho_1 I) \leq 0$ for some $\rho_1 > \rho(B)$, then there must exist a number $\rho_2 \geq \rho_1$ such that $p(\rho_2) = 0$. This, however, contradicts the fact that $\rho(B)$ is the spectral radius of B. □

For the following statements, we define the three matrices below

$$B_{2n} = \begin{bmatrix} 0 & t & & & & & \\ t & 0 & 0 & 1-t & & & \\ 1-t & 0 & 0 & t & & & \\ & & & \ddots & & & \\ & & & t & 0 & 0 & 1-t \\ & & & 1-t & 0 & 0 & t \\ & & & & & t & 0 \end{bmatrix}, A_{2n} = \begin{bmatrix} 0 & t & & & & & \\ t & 0 & 0 & 1-t & & & \\ 1-t & 0 & 0 & t & & & \\ & & & \ddots & & & \\ & & & t & 0 & 0 & 1-t \\ & & & 1-t & 0 & 0 & t \\ & & & & & 1 & 0 \end{bmatrix},$$

and

$$C_{2n-1} = \begin{bmatrix} t & 0 & 1-t & & & & & & \\ 1-t & -\rho(t) & t & & & & & & \\ 0 & t & -\rho(t) & 0 & 1-t & & & & \\ 0 & 1-t & 0 & -\rho(t) & t & & & & \\ & & & & \ddots & & & & \\ & & & & t & -\rho(t) & 0 & 1-t \\ & & & & 1-t & 0 & -\rho(t) & t \\ & & & & & & t & -\rho(t) \end{bmatrix},$$

where $0 < t < 1$. Notice that the indices denote the order of the corresponding matrices and n is any nonnegative integer.

LEMMA 4.5. *For the spectral radii $\rho(A_{2n})$ and $\rho(B_{4n})$ of A_{2n} and B_{4n} we have $\rho(A_{2n}) \leq \rho(B_{4n})$.*

Proof. Let $[x_1, x_2, \ldots, x_{2n}]^T$ be the eigenvector of the irreducible nonnegative matrix A_{2n} corresponding to the spectral radius $\rho(A_{2n})$. Then, it is easy to show that the vector $[x_1, x_2, \ldots, x_{2n}, x_{2n}, \ldots, x_2, x_1]^T$ is an eigenvector of B_{4n} with corresponding eigenvalue $\lambda = \rho(A_{2n})$ from which it follows that $\rho(A_{2n}) \leq \rho(B_{4n})$. □

LEMMA 4.6. *If $\rho(t)$ is the spectral radius of B_{2n}, then* $\det(C_{2k-1}) > 0$ *for all* $k = 1, \ldots, n$.

Proof. It is easy to see that for $k = 1$ our assertion holds. For $k > 1$, we expand $\det(C_{2k-1})$ with respect to its first row to get $\det(C_{2k-1}) = t\det(B_{2(k-1)} - \rho(t)I) + (1-t)^2 \det(C_{2(k-1)-1})$. Since $B_{2(k-1)}$ is a principal submatrix of the nonnegative matrix B_{2n}, it will be $\rho(B_{2(k-1)}) \leq \rho(B_{2n})$. On the other hand, by Lemma 4.4 we obtain that $\det(B_{2(k-1)} - \rho(t)I) \geq 0$ for $k = 2, \ldots, n$. Thus, the proof of the present lemma is completed by induction on k. □

LEMMA 4.7. *The spectral radius $\rho(t) := \rho(B_{2n}(t))$ of B_{2n} strictly increases with t for* $0 < t < 1$.

Proof. We first observe that B_{2n} is a nonnegative and irreducible matrix as $0 < t < 1$. Then, it follows that $\rho(t)$ is a simple eigenvalue of B_{2n} and $\det(B_{2n} - \rho(t)I) = 0$. Taking the derivative of $\det(B_{2n} - \rho(t)I) = 0$ with respect to t and using the following two basic properties

$$\begin{aligned} \tfrac{d}{dt}(\det([a_1, a_2, \ldots, a_{2n}]^T)) &= \det([\tfrac{d}{dt}a_1, a_2, \ldots, a_{2n}]^T) + \det([a_1, \tfrac{d}{dt}a_2, \ldots, a_{2n}]^T) + \ldots \\ &+ \det([a_1, a_2, \ldots, \tfrac{d}{dt}a_{2n}]^T) \end{aligned}$$

and

$$\begin{aligned} &\det([a_1, \ldots, a_{i-1}, a_i + b_i, a_{i+1}, \ldots, a_{2n}]^T) \\ = \ &\det([a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{2n}]^T) + \det([a_1, \ldots, a_{i-1}, b_i, a_{i+1}, \ldots, a_{2n}]^T), \end{aligned}$$

with each a_i or b_i denoting a vector of length $2n$, we obtain

$$2\{\sum_{k=0}^{n-1} \det(B_{2k} - \rho(t)I) \det(B_{2(n-1)-2k} - \rho(t)I)\}\rho'(t)\rho(t) + \sum_{k=1}^{2n} \det(\tilde{B}_k) = 0. \qquad (4.13)$$

In (4.13), $\det(B_0 - \rho(t)I)$ is defined to be 1 and $\tilde{B}_k$ is a matrix with the same entries as $B_{2n} - \rho(t)I$ except that its entries in the positions (k, k), $(k, k + 2(-1)^k)$ and $(k, k - (-1)^k)$ are 0, -1 and 1, respectively. Since $\rho(t)$ is a simple root of $\det(B_{2n} - \lambda I) = 0$ and from Lemma 4.4 we have $\det(B_{2n} - \lambda I) > 0$ for $\lambda > \rho(t)$, it is implied that $\frac{d}{d\lambda}\det(B_{2n} - \lambda I) > 0$ for $\lambda = \rho(t)$. Thus, the coefficient of $\rho'(t)$ in (4.13) is positive, because it is equal to the value of $\frac{d}{d\lambda}\det(B_{2n} - \lambda I)$ at $\lambda = \rho(t)$. So, to show that $\rho'(t) > 0$, it suffices to show that $\sum_{k=1}^{2n} \det(\tilde{B}_k) < 0$. For the terms corresponding to $k = 1$ and $k = 2n$, it is easy to show that $\det(\tilde{B}_1) = \det(\tilde{B}_{2n}) = -\det(C_{2n-1})$. Thus, from Lemma 4.6 we obtain $\det(\tilde{B}_1) = \det(\tilde{B}_{2n}) < 0$. For the remaining terms, we shall consider pairs such that $k = 2i$ and $k = 2i + 1$ simultaneously. First, we switch the $(2i)$th row of $\tilde{B}_{2i+1}$ with its $(2i+1)$st one and multiply the new $(2i)$th row by -1. Note that the determinant of the resulting matrix is equal to $\det(\tilde{B}_{2i+1})$ and only its $(2i+1)$st row differs from that of the matrix $\tilde{B}_{2i}$.

Then, we apply the second property above to get $\det(\tilde{B}_{2i}) + \det(\tilde{B}_{2i+1}) = \det(T_i)$, where T_i is the matrix of order $2n$ shown below

$$\begin{array}{r}
\\ \\ \\ \\
(2i)th \rightarrow \\
(2i+1)st \rightarrow \\
\\ \\ \\ \\
\end{array}
\begin{bmatrix}
-\rho(t) & t & & & & & & & & \\
t & -\rho(t) & 0 & 1-t & & & & & & \\
1-t & 0 & -\rho(t) & t & & & & & & \\
 & & & \ddots & & & & & & \\
 & & & 1 & 0 & 0 & -1 & & & \\
 & & & 1 & -\rho(t) & -\rho(t) & 1 & & & \\
 & & & & & & \ddots & & & \\
 & & & & & & t & -\rho(t) & 0 & 1-t \\
 & & & & & & 1-t & 0 & -\rho(t) & t \\
 & & & & & & & & t & -\rho(t)
\end{bmatrix}.$$

Expanding the determinant of T_i with respect to its $(2i)$th row, we obtain

$$\det(T_i) = -\det(A_{2n-2i} - \rho(t)I)\det(C_{2i-1})) - \det(A_{2i} - \rho(t)I)\det(C_{2n-(2i+1)}). \qquad (4.14)$$

Now, we will derive another expression for $\det(T_i)$. For this, first we add the $(2i)$th row of $B_{2n} - \rho(t)I$ to its $(2i+1)$st one. The resulting matrix has all its rows the same as those of the matrix T_i except for its $2ith$ row; its determinant is zero because $\det(B_{2n} - \rho(t)I) = 0$. Then, we multiply its $(2i)$th row by $1/(1-t)$ and add the new matrix to the matrix T_i. Note that the determinant of the resulting matrix is the same as $\det(T_i)$. In view of the structure of the new resulting matrix, we can easily get

$$\det(T_i) = \frac{1}{1-t}\det(A_{2n-2i} - \rho(t)I)\det(A_{2i} - \rho(t)I). \qquad (4.15)$$

From (4.15), we readily see that $\det(T_i) = \det(T_{n-i})$. In the discussion that follows, we assume that $1 < i \leq j$, where $j = [\frac{n}{2}]$ is the largest integer not exceeding $n/2$. Since A_{2i} is a principal submatrix of A_{2j}, we get $\rho(A_{2i}) \leq \rho(A_{2j})$. Furthermore, by Lemma 4.5 we have $\rho(A_{2i}) \leq \rho(A_{2j}) \leq \rho(B_{4j}) \leq \rho(B_{2n})$. It then follows that $\det(A_{2i} - \rho(t)I) \geq 0$ by Lemma 4.4. If we assume that $\det(T_i) > 0$ then both $\det(A_{2n-2i} - \rho(t)I)$ and $\det(A_{2i} - \rho(t)I)$ are nonnegative by (4.15). On the other hand, from Lemma 4.6 we know that both $\det(C_{2i-1})$ and $\det(C_{2n-(2i+1)})$ are positive, therefore, the right hand side of (4.14) is nonpositive, which contradicts the assumption $\det(T_i) > 0$. Consequently, we obtain $\det(\tilde{B}_{2i}) + \det(\tilde{B}_{2i+1}) \leq 0$ for $1 \leq i < n$. This together with the negativeness of the first and the last terms completes the proof. □

Let us now consider the case where $c_r \in (-\frac{l_0}{2}, l - l_0)$. Since G_k is a nonnegative matrix, $\rho(G_k) \geq \rho(B_{2(k-3)})$ for $t = \frac{l-l_0}{l+2c_r}$ because $B_{2(k-3)}$ is a principal submatrix of G_k. On the other hand, it is easy to show that $\rho(G_k) = \rho(B_{2(k-3)})$ for $t = \frac{l-l_0}{3l-2l_0}$ and $c_r = l - l_0$. Therefore, applying Lemma 4.7 and the fact that $\frac{l-l_0}{l+2c_r} > \frac{l-l_0}{3l-2l_0}$ for $c_r \in (-\frac{l_0}{2}, l - l_0]$ lead us to the conclusion that the spectral radius of the matrix G_k in (4.12) with $c_r = l - l_0$ is less than any one corresponding to $c_r \in (-\frac{l_0}{2}, l - l_0)$. This result with the first part of Theorem 4.2 show that the optimal value of c_r is attained at some point in the interval $[l - l_0, \infty)$. This completes the proof of the second part of Theorem 4.2.

4.3.3 The Multi-Parameter GSS As we have observed, there are many choices for the parameters ω in algorithm (4.3) and therefore in the linear system (4.11). Here, we shall consider the most general case, that is the one where there are two pairs of parameters $\omega_l^{(i)}$ and $\omega_r^{(i)}$ introduced for each subdomain Ω_i. Let $c_l^{(i)}$ and $c_r^{(i)}$ be defined in the same way as c_l and c_r were defined from ω_l and ω_r before. Let J be the block Jacobi iteration matrix associated with (4.11). Then, following similar analysis to that in Section 4.3.1, we get

$$\sigma(J) = \sigma(-G_k) \cup \{0\}$$

where

$$G_k = \left[\begin{array}{ccccccc} 0 & X & & & & & \\ Y_2 & 0 & 0 & \bar{Y}_2 & & & \\ Z_2 & 0 & 0 & \bar{Z}_2 & & & \\ & & & \ddots & & & \\ & & & Y_{k-1} & 0 & 0 & \bar{Y}_{k-1} \\ & & & Z_{k-1} & 0 & 0 & \bar{Z}_{k-1} \\ & & & & & \bar{X} & 0 \end{array}\right]$$

and

$$X = \frac{l-l_0-c_l^{(2)}}{l+c_r^{(1)}}, \quad Y_i = \frac{l-l_0-c_r^{(i-1)}+c_r^{(i)}}{l+c_r^{(i)}+c_l^{(i)}}, \ \bar{Y}_i = \frac{l_0+c_r^{(i-1)}+c_l^{(i)}}{l+c_r^{(i)}+c_l^{(i)}},$$
$$\bar{X} = \frac{l-l_0-c_r^{(k-1)}}{l+c_l^{(k)}}, \ Z_i = \frac{l_0+c_r^{(i)}+c_l^{(i+1)}}{l+c_r^{(i)}+c_l^{(i)}}, \quad \bar{Z}_i = \frac{l-l_0+c_l^{(i)}-c_l^{(i+1)}}{l+c_r^{(i)}+c_l^{(i)}}.$$

Following the same approach as in the proof of [26, Theorem 3.1], it can be shown that the next theorem holds.

THEOREM 4.3. *Let $c_l^{(i)} = (i-1)(l-l_0)$, $c_r^{(i)} = (k-i)(l-l_0)$, $i = j, \ldots, k$, where j is any integer in $\{1, \ldots, k\}$, and the remaining parameters $c_r^{(i)}$ and $c_l^{(i)}$ be any numbers such that $l + c_r^{(1)} \neq 0$ and $l + c_r^{(i)} + c_l^{(i)} \neq 0$. Then, we have $\rho(J) = 0$.*

Note In view of the structure of the corresponding G_k matrix, it is observed that among all the sets of parameters the set $c_l^{(i)} = (i-1)(l-l_0)$, $c_r^{(i)} = (k-i)(l-l_0)$, $i = 1, \ldots, k$, minimizes the maximum order of the Jordan blocks of G_k which is $k-1$. We have also observed from a number of experiments carried out that the maximum order of Jordan blocks affects very slightly the number of iterations required to achieve a specified accuracy.

4.4 Two-Dimensional Case We consider the Poisson equation

$$\Delta u = f \ in \ \Omega$$

with boundary conditions

$$u = g \ on \ \partial\Omega$$

where Ω is a rectangle. For this problem, we have that $A_1 = -rT_1 + tT_2$, $A_2 = sT_1 + wT_2$, $A_3 = rT_1 + \bar{t}T_2$ and $A_4 = \bar{s}T_1 + \bar{w}T_2$, where

$$T_i = \left[\begin{array}{cc|cc} a_1 & a_2 & a_3 & -a_4 \\ a_3 & a_4 & a_1 & -a_2 \end{array}\right]_{(2m)}, i = 1, 2,$$

with

	a_1	a_2	a_3	a_4
T_1	t	w	$\bar{t}$	$\bar{w}$
T_2	$-r$	s	r	$\bar{s}$

where $r = 2\sqrt{3}, \quad s = -1 - \sqrt{3},$
$t = \frac{1}{2} + \frac{2}{3\sqrt{3}}, \; w = \frac{3+\sqrt{3}}{36}.$

Note that $\bar{t}$ denotes the "conjugate" of $t = t_1 + t_2\sqrt{3}$, i.e. $\bar{t} = t_1 - t_2\sqrt{3}$ and T_i is defined as in [30].

Following the proof of Lemma 5.1 in [27], it can be shown that there exists a nonsingular matrix V such that $VT_2 = DVT_1$ with

$$D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{2m}) = \operatorname{diag}\left(-12, \; -36, \; x_1^+, x_1^-, \ldots, x_{m-1}^+, x_{m-1}^-\right),$$

where

$$\begin{aligned} x_j^{\pm} &= \tfrac{12[(8+\cos\theta_j)\pm\sqrt{43+40\cos\theta_j-2(\cos\theta_j)^2})]}{-7+\cos\theta_j}, \\ \theta_j &= \tfrac{j\pi}{m}. \end{aligned}$$

Thus, it follows that there exist nonsingular matrices P and Q of order $2m$ such that $PA_iQ = D_i$, $i = 1, 2, 3, 4$, and each D_i is a diagonal matrix. Then, it can be shown that

$$\begin{aligned} &\begin{bmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix} (-N^{-1}M) \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \\ = &\begin{bmatrix} (D_1D_4 - D_2D_3)^{-1} & 0 \\ 0 & (D_1D_4 - D_2D_3)^{-1} \end{bmatrix} \begin{bmatrix} D_1D_2 - D_3D_4 & D_4^2 - D_2^2 \\ D_3^2 - D_1^2 & D_1D_2 - D_3D_4 \end{bmatrix}. \end{aligned}$$

Let $\bar{P} = [e_1, e_{2m+1}, e_2, e_{2m+2}, \ldots, e_{2m}, e_{4m}]$, where e_i has 1 as its ith component and 0's elsewhere. It is clear that

$$\bar{P} \begin{bmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix} (-N^{-1}M) \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \bar{P} = \operatorname{diag}(\bar{D}_1, \ldots, \bar{D}_{2m}),$$

where $\bar{D}_j = \begin{bmatrix} \bar{d}_{1j} & \bar{d}_{2j} \\ \bar{d}_{3j} & \bar{d}_{1j} \end{bmatrix}$ with the property $\det(\bar{D}_j) = 1$, $j = 1, \cdots, 2m$. On the other hand, it is easily found out that

$$\bar{d}_{1j} = \frac{(-r + t\lambda_j)(s + w\lambda_j) - (r + \bar{t}\lambda_j)(\bar{s} + \bar{w}\lambda_j)}{(-r + t\lambda_j)(\bar{s} + \bar{w}\lambda_j) - (s + w\lambda_j)(r + \bar{t}\lambda_j)} = \frac{432 - 192\lambda_j + 7\lambda_j^2}{432 + 24\lambda_j + \lambda_j^2}.$$

Also, it is observed that $\bar{d}_{1j} > 1$ because all λ_j's are less than 0. Hence, we may set $\bar{d}_{1j} = cosh\theta_j$ for some $\theta_j > 0$. Using the fact that $\det(\bar{D}_j) = 1$, it is proved that $\bar{d}_{2j}\bar{d}_{3j} = \sinh\theta_j$. Therefore there exist nonsingular matrices $Q_j = \begin{bmatrix} 1 & 1 \\ \sqrt{\frac{\bar{d}_{3j}}{\bar{d}_{2j}}} & -\sqrt{\frac{\bar{d}_{3j}}{\bar{d}_{2j}}} \end{bmatrix}$ such that $Q_j^{-1}\bar{D}_jQ_j = \operatorname{diag}(\cosh\theta_j - \sinh\theta_j, \cosh\theta_j + \sinh\theta_j)$. Let $\bar{Q} = \operatorname{diag}(Q_1, \ldots, Q_{2m})$, then we obtain that $\bar{Q}^{-1}\bar{P} \begin{bmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix} (-N^{-1}M)^p \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \bar{P}\bar{Q} = \operatorname{diag}(\cosh p\theta_1 - \sinh p\theta_1,$ $\cosh p\theta_1 + \sinh p\theta_1, \ldots, \cosh p\theta_{2m} - \sinh p\theta_{2m}, \cosh p\theta_{2m} + \sinh p\theta_{2m})$. Thus, using the equation

$$Q_j \begin{bmatrix} \cosh p\theta_j - \sinh p\theta_j & 0 \\ 0 & \cosh p\theta_j + \sinh p\theta_j \end{bmatrix} Q_j^{-1} = \begin{bmatrix} \cosh p\theta_j & -\sqrt{\frac{\bar{d}_{2j}}{\bar{d}_{3j}}}\sinh p\theta_j \\ -\sqrt{\frac{\bar{d}_{3j}}{\bar{d}_{2j}}}\sinh p\theta_j & \cosh p\theta_j \end{bmatrix},$$

we can summarize the discussion and conclude that

$$\begin{bmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix} (-N^{-1}M)^p \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} = \begin{bmatrix} \bar{D}_{1p} & \bar{D}_{2p} \\ \tilde{D}_{3p} & \tilde{D}_{1p} \end{bmatrix}, \tag{4.16}$$

where

$$\begin{cases} \tilde{D}_{1p} = \operatorname{diag}(\cosh p\theta_1, \cosh p\theta_2, \ldots, \cosh p\theta_{2m}) \\ -\tilde{D}_{2p} = \operatorname{diag}(\sqrt{\frac{\bar{d}_{21}}{d_{31}}} \sinh p\theta_1, \sqrt{\frac{\bar{d}_{22}}{d_{32}}} \sinh p\theta_2, \ldots, \sqrt{\frac{\bar{d}_{2m}}{d_{3m}}} \sinh p\theta_{2m}) . \\ -\tilde{D}_{3p} = \operatorname{diag}(\sqrt{\frac{\bar{d}_{31}}{d_{21}}} \sinh p\theta_1, \sqrt{\frac{\bar{d}_{32}}{d_{22}}} \sinh p\theta_2, \ldots, \sqrt{\frac{\bar{d}_{3m}}{d_{2m}}} \sinh p\theta_{2m}) \end{cases} \tag{4.17}$$

Applying now (4.16) to express X, Y and Z of the matrix G_k defined in (4.12) we can come to the following conclusion.

THEOREM 4.4. *The spectrum of the block Jacobi iteration matrix associated with the enhanced linear system (4.11) for the two-dimensional Poisson model problem is given by*

$$\sigma(J) = \sigma(-G'_k) \cup \{0\}.$$

The matrix G'_k is of the same structure as G_k defined in (4.12) with

$$X = (\tilde{D}_{2(l-l_0)} + c_r \tilde{D}_{1(l-l_0)})(\tilde{D}_{2l} - c_r \tilde{D}_{1l})^{-1},$$
$$Y = (\bar{D}_{2(l-l_0)} - c_r^2 \bar{D}_{3(l-l_0)})(\bar{D}_{2l} - 2c_r \bar{D}_{1l} + c_r^2 \tilde{D}_{3l})^{-1},$$
$$Z = (\bar{D}_{2l_0} - 2c_r \bar{D}_{1l_0} + c_r^2 \tilde{D}_{3l_0})(\bar{D}_{2l} - 2c_r \tilde{D}_{1l} + c_r^2 \tilde{D}_{3l})^{-1}$$

and $\bar{D}_{ip}, i = 1, 2, 3$, being defined in (4.17).

COROLLARY 4.1. *The SAM algorithm converges for all possible combinations of l, l_0 and k.*

Proof. In the traditional approach to SAM (c_r is chosen to be zero) X, Y and Z can be simplified to

$$X = Y = \operatorname{diag}(\frac{\sinh(2l - 2l_0)\theta_1}{\sinh 2l\theta_1}, \ldots, \frac{\sinh(2l - 2l_0)\theta_{2m}}{\sinh 2l\theta_{2m}})$$

and

$$Z = \operatorname{diag}(\frac{\sinh 2l_0\theta_1}{\sinh 2l\theta_1}, \ldots, \frac{\sinh 2l_0\theta_{2m}}{\sinh 2l\theta_{2m}}).$$

Then it follows from

$$\frac{\sinh(2l - 2l_0)\theta + \sinh 2l_0\theta}{\sinh 2l\theta} = \frac{2\cosh(l - 2l_0)\theta \sinh l\theta}{2\sinh l\theta \cosh l\theta} = \frac{\cosh(l - 2l_0)\theta}{\cosh l\theta}$$

that

$$\rho(G'_k) \leq \|G'_k\|_\infty = \max_{\theta_i, i=1,\ldots,2m} \frac{\cosh(l - 2l_0)\theta_i}{\cosh l\theta_i} < 1. \quad \square$$

Note It is well understood from the proof of the corollary above that the amount of l_0/l is a key factor that affects the convergence rate of SAM.

4.5 Numerical Examples In this section, we present a number of numerical examples to verify the theoretical results obtained in the previous sections. We use the zero vector as the initial guess of the solution of the enhanced linear system (4.11). We display the maximum error $||u - u_{\frac{1}{n}}||_\infty$ based on an $n \times n$ grid of points, where u is the theoretical solution of the continuous problem and $u_{\frac{1}{n}}$ is the computed one. The iteration step (*iter*) denotes the number of the block Gauss-Seidel iterations required to satisfy the stopping criterion $\frac{||y^{(j)}-y^{(j-1)}||_\infty}{||y^{(j)}||_\infty} < \epsilon$, where $y^{(j)}$ is the jth iteration approximation to the solution of the linear system (4.11) and $\epsilon = 1.0e-6$ and $\epsilon = 5.0e-6$ for 1-D and 2-D problems, respectively. Throughout, we denote by 1-GSS the one parameter GSS and m-GSS the muti-parameter GSS.

TABLE 4.1

The convergence of the SAM, 1-GSS and m-GSS methods for 1-dimensional model boundary value problem with exact solution $u(x) = e^{-100(x-0.1)^2}(x^2 - x)$. The number of subdomains (k), grid size, number of iterations taken for the splitting scheme to converge and the discretization error are displayed for two different domain splittings.

	1-GSS($c_r = l - l_0$)				SAM				m-GSS			
	$l_0 = l/2$		$l_0 = 1$		$l_0 = l/2$		$l_0 = 1$		$l_0 = l/2$		$l_0 = 1$	
$(k, grid)$	iter	error	iter	error	iter	error	iter	error	iter	error	iter	error
(2,10)	2	6.64e-3	2	6.64e-3	6	6.64e-3	13	6.64e-3	2	6.64e-3	2	6.64e-3
(2,22)	2	8.42e-5	2	8.42e-5	4	8.42e-5	10	8.36e-5	2	8.42e-5	2	8.42e-5
(3,17)	3	3.27e-4	3	3.27e-4	7	3.27e-4	23	3.27e-4	3	3.27e-4	3	3.27e-4
(3,29)	3	2.67e-5	3	2.67e-5	5	2.67e-5	38	2.57e-5	2	2.67e-5	3	2.67e-5
(4,26)	4	4.84e-5	4	4.84e-5	8	4.82e-5	73	4.72e-5	3	4.84e-5	3	4.84e-5
(4,46)	3	4.28e-6	3	4.25e-6	7	4.05e-6	125	6.12e-6	3	4.31e-6	3	4.30e-6

For the one-dimensional case, we are using the boundary value problem

$$\begin{aligned} u''(x) &= f(x), \ x \in (0,1), \\ u(0) &= g_0, \quad u(1) = g_1, \end{aligned}$$

where $f(x)$, g_0 and g_1 are selected such that the exact solution is $u(x) = e^{-100(x-0.1)^2}(x^2-x)$. We apply both the traditional SAM and the one-parameter GSS with $c_r = l - l_0$. This is the optimal value for the case $k = 2$ or 3 both for minimum and maximum overlaps. For the multi-parameter GSS and the domain split with minimum overlap, among the many choices of the parameters $c_r^{(i)}$ and $c_l^{(i)}$, we choose $c_l^{(i)} = (i-1)(l-l_0)$, and $c_r^{(i)} = (k-i)(l-l_0)$, $i = 1, \ldots, k$. The numerical results obtained are summarized in Table 4.1. The data in Table 4.1 verify our theoretical results, namely that the one-parameter GSS outperforms the traditional SAM and for $k = 2$ and 3 we get the optimal convergence. However, 1-GSS is slower (based on the number of iterations) than the multi-parameter GSS.

Figure 4.2 displays the relation between the number of iterations and the parameters c_r for the 1-GSS for four pairs of (k, l), where k denotes the number of subdomains and l denotes the number of subinterval in each sundomain. Our experiments are carried out for maximum (half) overlap. From these plots, we can conclude that $c_r = l - l_0$ is indeed the

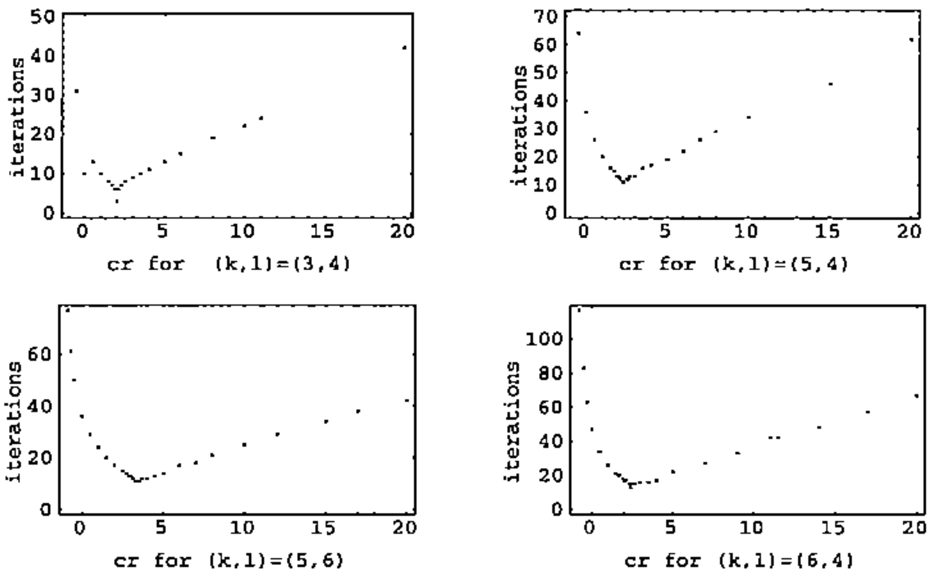


FIG. 4.2. *Plots of the number of iterations required by 1-GSS to achieve convergence. Iterations versus c_r for the one-dimensional problem with maximum overlap and different pairs (k, l).*

optimal value for the case $k = 3$ while the optimal value of c_r for $k > 3$ is on the right of $l - l_0$ as this was shown in Section 4.3.1. Moreover, it appears that the optimal value of c_r can be expressed as $\alpha(l - l_0)$ for some number α, which seems to increase with k. Also, from the same plots, we can observe that the traditional SAM (case $c_r = 0$) has a very poor convergence rate compared to that of the one-parameter GSS with $c_r = l - l_0$.

For the two-dimensional case, let the domain Ω be a unit square. First we consider the Poisson equation

$$\begin{aligned} u_{xx} + u_{yy} = f(x, y), \quad (x, y) \in \Omega \\ u(x, y) = g(x, y), \quad (x, y) \in \partial\Omega, \end{aligned} \tag{4.18}$$

where $f(x, y)$ and $g(x, y)$ are selected so that

$$u(x, y) = 10e^{-100(x-0.1)^2}(x^2 - x)e^{-100(y-0.1)^2}(y^2 - y). \tag{4.19}$$

Then, we consider the more general PDE operator

$$[2 + (y - 1)e^{-y^4}]u_{xx} + [1 + \frac{1}{(1 + 4x^2)}]u_{yy} + 5[x(x - 1) + (y - 0.3)(y - 0.7)]u = f, \tag{4.20}$$

taken from [33], with Dirichlet boundary conditions and the same solution (4.19) on the domain Ω.

For the Poisson problem using a 2-way splitting (k=2), we can derive all the eigenvalues of the corresponding Jacobi matrix by Theorem 4.4 for any l and l_0. In Figure 4.3, the relations between the spectral radii and the parameters ω, c_r are depicted for maximum overlap. In these figures, we can see that for a fixed parameter ω the spectral radius decreases with the value of l increasing. In addition, we observe that for a given l the minimum of the spectral radius always occurs near $\omega = 0.8$. Note that $c_r = \frac{3l}{8}$, which is close to the value $l - l_0 = l/2$ for maximum overlap and for small values of l.

Tables 4.2 and 4.3 display the convergence behavior of SAM and 1-GSS for thoese two problems for different splittings and grids with maximum and minimum overlap. Since the theoretical values of the optimal parameter c_r for these problems are not known, we experimented with the value $c_r = l - l_0$ which corresponds to the case $k = 2$ as this can be seen from Figure 4.3. In Table 4.2 and 4.3 we observe that an improvement regarding the number of iterations required is obtained by using $c_r = l - l_0$ rather than $c_r = 0$. The data indicate that the improvement is more significant when l is a small number. The reason is that the spectral radius of the corresponding Jacobi matrix might be very small when l is getting bigger (as shown in Figure 4.3), which implies that the stopping criterion can be achieved by a small number of iterations. However, in our experiments we also observe the value of $\frac{\|y^{(j)}-y^{(j-1)}\|_\infty}{\|y^{(j)}\|_\infty}$ at each iteration and conclude the one-parameter GSS with $c_r = l - l_0$ does outperform the traditional SAM for any l.

4.6 Concluding Remarks and Discussion In this paper we have studied the parameterized GSS at a discrete equation level (matrix formulation), coupled with the cubic Hermite collocation discretization scheme for both the one- and the two-dimensional model problems. For the one-dimensional problem, we have found the optimal parameter values which correspond to the smallest possible spectral radius of the block Jacobi iteration matrix associated with (4.11) for $k = 2, 3$ in the one-parameter case and for all k in the multi-parameter case. We also determined the interval in which the parameter c_r must lie so that the convergence of the Jacobi method would be guaranteed. Moreover, a subinterval of the previous one was found in which the optimal value of c_r for $k > 3$ should lie. The determination of the optimal parameter c_r in question is still an open problem but our analysis suggests that this optimal value is a number greater than $l - l_0$. For the two-dimensional case, our analysis consists of Theorem 4.4. This theorem improves our understanding of the relation between the parameter c_r and the convergence properties of the corresponding block Jacobi iteration matrix. In addition, it provides a simpler matrix G'_k to determine this relation. In particular, for $k = 2$ we have experimented with several combinations of l and ω or c_r with $l_0 = l/2$ to obtain the corresponding spectral radius as shown in Figure 4.3. From the experiments, we can see that $\omega = 0.8$ is independent of l and may give an almost optimal convergence rate among cases with l being fixed.

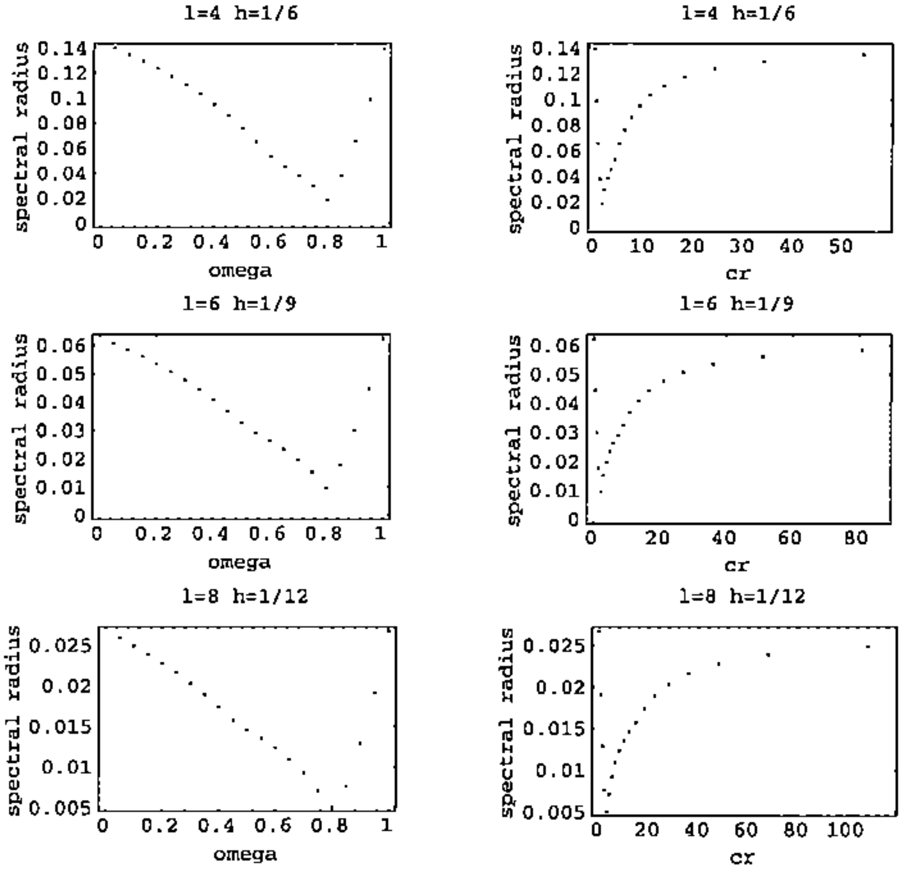


FIG. 4.3. *The Jacobi matrix spectral radius versus ω and c_r parameters for the two-dimensional Poisson model problem with 2-way domain splitting ($k = 2$) and maximum overlap*

TABLE 4.2

The convergence of the 1-GSS ($c_r = l - l_0$) and SAM ($c_r = 0.$) for a Dirichlet model problem with minimum and maximum overlap splittings of the PDE domains. The exact solution is $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$. The number of subdomains (k), grid size, number of iteration and discretization error are displayed for both splittings.

		$l_0 = l/2$				$l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
$(k, grid)$	l	iter	error	iter	error	iter	error	iter	error	l
(2,10×10)	6	3	7.74e-3	3	7.74-3	4	7.74e-3	5	7.74e-3	5
(2,22×22)	14	2	1.53e-4	2	1.53e-4	2	1.53e-4	2	1.53e-4	11
(3,17×17)	8	2	6.08e-4	3	6.08e-4	5	6.08e-4	7	6.08e-4	6
(3,29×29)	14	2	4.69e-5	2	4.69e-5	6	4.69e-5	10	4.61e-5	10
(4,26×26)	10	3	9.09e-5	7	9.09e-5	6	9.08e-5	20	8.94e-5	7

	$l = 4, l_0 = l/2$					$l = 4, l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
k	grid	iter	error	iter	error	iter	error	iter	error	grid
2	7×7	3	1.74e-2	5	1.74e-2	5	2.29e-2	10	2.29e-2	8x8
3	9×9	5	1.56e-2	7	1.56e-2	4	3.19e-3	6	3.19e-3	11x11
4	11×11	4	3.19e-3	5	3.19e-3	6	7.34e-4	10	7.33e-4	14x14
5	13×13	5	3.24e-4	6	3.24e-4	7	6.08e-4	16	6.06e-4	17x17
6	15×15	4	8.51e-4	8	8.50e-4	7	2.21e-4	24	2.18e-4	20x20

	$l = 6, l_0 = l/2$					$l = 6, l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
k	grid	iter	error	iter	error	iter	error	iter	error	grid
2	10×10	3	7.74e-3	3	7.74e-3	3	1.13e-3	4	1.13e-3	12x12
3	13×13	2	3.24e-4	3	3.24e-4	5	6.08e-4	7	6.08e-4	17x17
4	16×16	3	7.64e-4	4	7.64e-4	6	1.53e-4	16	1.51e-4	22x22
5	19×19	5	3.21e-4	12	3.21e-4	7	7.37e-5	30	7.18e-5	27x27

TABLE 4.3

The convergence of the 1-GSS ($c_r = l - l_0$) and SAM ($c_r = 0.$) for a general PDE with minimum and maximum overlap splittings of the unit square. The exact solution is $u(x,y) = 10\phi(x)\phi(y)$, where $\phi(x) = e^{-100(x-0.1)^2}(x^2 - x)$. The number of subdomains (k), grid size, number of iteration and discretization error are displayed for both splittings.

	$l_0 = l/2$					$l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
$(k, grid)$	l	iter	error	iter	error	iter	error	iter	error	l
(2,10×10)	6	3	7.72e-3	3	7.72e-3	4	7.72e-3	5	7.72e-3	5
(2,22×22)	14	2	1.54e-4	2	1.54e-4	2	1.54e-4	2	1.54e-4	11
(3,17×17)	8	3	6.08e-4	3	6.08e-4	5	6.08e-4	6	6.08e-4	6
(3,29×29)	14	2	4.70e-5	2	4.70e-5	6	4.70e-5	10	4.66e-5	10
(4,26×26)	10	3	9.12e-5	5	9.12e-5	6	4.70e-5	17	8.98e-5	7

	$l = 4, l_0 = l/2$					$l = 4, l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
k	grid	iter	error	iter	error	iter	error	iter	error	grid
2	7×7	3	1.75e-2	5	1.75e-2	5	2.31e-2	10	2.31e-2	8x8
3	9×9	5	1.57e-2	7	1.57e-2	4	3.17e-3	6	3.17e-3	11x11
4	11×11	3	3.17e-3	5	3.17e-3	6	7.34e-4	9	7.33e-4	14x14
5	13×13	5	3.24e-4	6	3.23e-4	6	6.08e-4	15	6.06e-4	17x17
6	15×15	6	8.50e-4	8	8.49e-4	6	2.21e-4	18	2.18e-4	20x20

	$l = 6, l_0 = l/2$					$l = 6, l_0 = 1$				
		$c_r = l - l_0$		$c_r = 0.$		$c_r = l - l_0$		$c_r = 0.$		
k	grid	iter	error	iter	error	iter	error	iter	error	grid
2	10×10	3	7.72e-3	3	7.72e-3	3	1.13e-3	3	1.13e-3	12x12
3	13×13	3	3.24e-4	3	3.24e-4	5	6.08e-4	6	6.08e-4	17x17
4	16×16	3	7.63e-4	3	7.63e-4	6	1.54e-4	14	1.54e-4	22x22
5	19×19	6	3.21e-4	11	3.21e-4	7	7.40e-5	38	7.37e-5	27x27

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences.* Acedemic Press, New York, 1979.

[2] B. Bialecki, G. Fairweather, and K. R. Bennett. Fast Direct Solvers for Piecewise Hermite Bicubic Orthogonal Spline Collocation Equations. *SIAM J. Numer. Anal.*, 29:156–173, 1992.

[3] C. Canuto and D. Funaro. The Schwarz Algorithm for Spectral Methods. *SIAM J. Numer. Anal.*, 25:24–40, 1988.

[4] M. P. Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, N. J., 1976.

[5] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. Wiley-Interscience, New York, 1962.

[6] W. R. Dyksen. Tensor Product Generalized ADI Method for Separable Elliptic Problems. *SIAM J. Numer. Anal.*, 24:59–76, 1987.

[7] W. R. Dyksen, E. N. Houstis, R. E. Lynch, and J. R. Rice. The Performance of the Collocation and Galerkin Methods with Hermite Bicubics. *SIAM J. Numer. Anal.*, 21:695–715, 1984.

[8] W. R. Dyksen and J. R. Rice. A New Ordering Scheme for the Hermite Bicubic Collocation Equations. In G. Birkhoff and eds A. Schoenstat, editors, *Elliptic Solver II.* Academic Press, New York, 1984.

[9] W. R. Dyksen and J. R. Rice. The Importance of Scaling for the Hermite Bicubic Collocation Equations. *SIAM J. Sci. Stat. Comput.*, 7:707–719, 1986.

[10] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials.* Academic Press, New York, 1982.

[11] G. H. Golub and C. F. VanLoan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1983.

[12] A. Hadjidimos. Accelerated Overrelaxation Method. *Math. Comp.*, 32:149–157, 1978.

[13] A. Hadjidimos. On the Optimization of the Classical Iterative Schemes for the Solution of Complex Singular Linear Systems. *SIAM J. Alg. Disc. Meth.*, 6:555–566, 1985.

[14] A. Hadjidimos, T. S. Papatheodorou, and Y. G. Saridakis. Optimal Block Iterative Schemes for Certain Large Sparse and Non-symmetric Linear Systems. *Linear Algebra Appl.*, 110:285–318, 1988.

[15] A. Hageman and D. M. Young. *Applied Iterative Methods.* Academic Press, New York, 1981.

[16] P. R. Halmos. *Finite Dimensional Vector Spaces.* Princeton University, 1948.

[17] E. N. Houstis. Collocation Methods for Linear Elliptic Problems. *BIT*, 18:301–310, 1978.

[18] E. N. Houstis. A Collocation Methods for Systems of Nonlinear Ordinary Differential Equations. *J. Math. Anal. Appl.*, 62:24–37, 1978.

[19] E. N. Houstis, R. E. Lynch, T. S. Papatheodorou, and J. R. Rice. Evaluation of Numerical Methods for Elliptic Partial Differential Equations. *J. Comput. Phy.*, 27:323–350, 1978.

[20] E. N. Houstis, W. Mitchell, and J. R. Rice. Algorithm GENCOL: Collocation on General Domains with Bicubic Hermite Polynomials. *ACM Trans. Math. Software*, 11:413–415, 1985.

[21] E. N. Houstis, W. Mitchell, and J. R. Rice. Algorithm INTCOL and HERMCOL: Collocation on Rectangular Domains with Bicubic Hermite Polynomials. *ACM Trans. Math. Software*, 11:416–418, 1985.

[22] E. N. Houstis, W. Mitchell, and J. R. Rice. Collocation Software for Second Order Elliptic Partial Differential Equations. *ACM Trans. Math. Software*, 11:379–412, 1985.

[23] E. N. Houstis and T. S. Papatheodorou. Numerical Methods for Mildly Nonlinear Elliptic Partial Differential Equations. *Inter. J. Numer. Meth. Engin.*, 19:665–709, 1982.

[24] E. N. Houstis, T. S. Papatheodorou, and R. Balart. On the Iterative Solution of Collocation Equations. In *10th IMACS World Congress Proceedings*, pages 98–100, Montreal, 1982.

[25] L. V. Kantorovich and V. Krylov. *Approximate Methods of Higher Analysis.* Wiley-Interscience, New York, 1958.

[26] S.-B. Kim, A. Hadjidimos, E. N. Houstis, and J. R. Rice. Multi-Parameterized Schwartz Splittings. Technical Report CSD-TR-92-073, Comp. Sci. Dept. Purdue Univ., 1992.

[27] Y. L. Lai, A. Hadjidimos, E. N. Houstis, and J. R. Rice. On the Iterative Solution of Hermite Collocation Equations. to apper in SIAM J. Matrix Anal. Appl., 1993.

[28] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct Solution of Partial Difference Equations by Tensor Product Methods. *Numer. Math.*, 6:185–189, 1964.

[29] K. Miller. Numerical Analogs to the Schwarz Alternating Procedure. *Numer. Math*, 7:91–103, 1965.

[30] T. S. Papatheodorou. Block AOR Iteration for Non-symmetric Matrices. *Math. Comp.*, 41:511–525, 1983.

[31] P. M. Prenter. *Splines and Variational Methods.* Wiley, New York, 1975.

[32] P. M. Prenter and R. D. Russell. Orthogonal Collocation for Elliptic Partial Differential Equation. *SIAM J. Numer. Anal.*, 13:923–939, 1976.

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

[34] Y. Saad and M. H. Schultz. A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear System. *SIAM J. Sci. Stat. Comput.*, 6:856–869, 1985.

[35] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2. Springer, Berlin, New York, 1890.

[36] W. Joubert T. Oppe and D. Kincaid. A Package for Solving Large Sparse Linear Systems by Various Iterative Methods. Technical Report CNA-216, Center for Numerical Analysis, University of Texas at Austin, 1988.

[37] W. P. Tang. *Schwarz Splitting and Template Operator.* PhD thesis, Dept. of Computer Sciences, Stanford Univ., 1987.

[38] W. P. Tang. Generalized Schwarz Splittings. *SIAM J. Stat. Comput.*, 13:573–595, 1992.

[39] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1962.

[40] E. G. Yanik. A Schwarz Alternating Procedure Using Spline Collocation Methods. *Inter. J. Numer. Methods Engin.*, 28:621–627, 1989.

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

[42] D. M. Young and H. E. Eidson. On the Determination of the Optimal Relaxation Factor for the SOR Method When the Eigenvalues of the Jacobi Method are Complex. Technical Report CNA-1, Center of Numerical Analysis, Univ. of Texas, Austin, TX, 1990.

[43] O. Zienkiewicz. *The Finite Element Method in Engineering Science.* McGraw-Hill, London, 1971.