



Sinc domain decomposition methods for elliptic problems  
by Nancy Jean Lybeck

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in  
Mathematical Sciences  
Montana State University  
© Copyright by Nancy Jean Lybeck (1994)

Abstract:

Sinc basis functions form a desirable basis to use for solving singular problems via domain decomposition. This is because both the Sinc-Galerkin and sinc-collocation methods converge exponentially, even in the presence of boundary singularities. For Poisson's equation with homogeneous Dirichlet boundary conditions posed on a rectangular domain, the Sinc-Galerkin and sinc-collocation methods have been well developed. The sinc methods have also been developed for any domain which can be mapped to a rectangular domain via an invertible or conformal mapping. In order to increase the number and complexity of domains which can be handled via sinc methods, domain decomposition techniques are used.

The Sinc-Galerkin and sinc-collocation domain decomposition methods are first studied for a two-point boundary-value problem. Both of the traditional methods of domain decomposition, overlapping and patching, are developed. This lays the groundwork to readily determine which method is most suited to any given problem. Because the goal is to clearly develop and test the sinc domain decomposition methods, techniques such as subdomain iterations and preconditioning are not employed here. The number of subdomains is limited to two in order to limit the complexity of the presentation. Numerical results are presented for both decomposition methods that exhibit the nearly identical errors achieved whether one uses the sinc-collocation or Sinc-Galerkin method.

Next the patching and overlapping Sinc-Galerkin methods are presented for Poisson's equation presented on a rectangle. For certain parameter choices the sinc-collocation system is identical for these problems, and is thus not presented separately. Again the number of subdomains is limited to two in order to present the material more clearly. Both domain decomposition methods perform well, and this is highlighted in the numerical examples.

Finally, Poisson's equation is studied on an  $\epsilon$ -shaped domain. In the derivation of the discrete system, it becomes evident that the patching domain decomposition method is the method of choice for this problem. The derivation and numerical examples are presented using three subdomains, although multiple subdomains could certainly be used. Numerical examples illustrate the ability of this method to handle these types of problems.

SINC DOMAIN DECOMPOSITION METHODS  
FOR ELLIPTIC PROBLEMS

by

Nancy Jean Lybeck

A thesis submitted in partial fulfillment  
of the requirements for the degree

of

Doctor of Philosophy

in

Mathematical Sciences

MONTANA STATE UNIVERSITY  
Bozeman, Montana

May 1994

D378  
L9816

**APPROVAL**

of a thesis submitted by

Nancy Jean Lybeck

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

April 20, 1994  
Date

Kenneth L. Bowers  
Kenneth L. Bowers  
Chairperson, Graduate Committee

Approved for the Major Department

April 20, 1994  
Date

John Lund  
John Lund  
Head, Mathematical Sciences

Approved for the College of Graduate Studies

4/20/94  
Date

Robert Brown  
Robert Brown  
Graduate Dean

**STATEMENT OF PERMISSION TO USE**

In presenting this thesis in partial fulfillment of the requirements for a doctoral degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. I further agree that copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for extensive copying or reproduction of this thesis should be referred to University Microfilms International, 300 North Zeeb Road, Ann Arbor, Michigan 48106, to whom I have granted "the exclusive right to reproduce and distribute my dissertation for sale in and from microform or electronic format, along with the right to reproduce and distribute my abstract in any format in whole or in part."

Signature Nancy Jean Lybeck

Date April 20, 1994

## ACKNOWLEDGEMENTS

I would like to thank Ken Bowers for introducing me to sinc functions and assigning a class project that eventually lead to this dissertation. His patience and friendship over the past few years have been deeply appreciated.

I would like to thank John Lund for persuading me to begin graduate school and to take a class in sinc functions. I also want to thank John Lund and Gary Bogar for their continued interest in my work and help in preparing this dissertation.

I would like to thank my family for their continued support during graduate school. I would especially like to thank my parents for their unwavering faith in me. Finally, I would like to thank my husband, Eric, for everything.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	ix
ABSTRACT . . . . .	xi
1. INTRODUCTION . . . . .	1
2. SINC METHODS FOR DIFFERENTIAL EQUATIONS . . . . .	5
Introduction . . . . .	5
Sinc Interpolation and Quadrature Methods . . . . .	6
Two-Point Boundary-Value Problem . . . . .	11
The Sinc-Galerkin Method . . . . .	11
The Sinc-Collocation Method . . . . .	15
Poisson's Equation on a Rectangle . . . . .	17
3. DOMAIN DECOMPOSITION FOR ORDINARY DIFFERENTIAL EQUATIONS . . . . .	22
Introduction . . . . .	22
The Overlapping Method of Domain Decomposition . . . . .	23
The Sinc-Galerkin Overlapping Method . . . . .	25
The Sinc-Collocation Overlapping Method . . . . .	33
Patching Domain Decomposition Methods . . . . .	39
The Sinc-Galerkin Patching Method . . . . .	40
The Sinc-Collocation Patching Method . . . . .	47
4. DOMAIN DECOMPOSITION FOR POISSON'S EQUATION . . . . .	52
Introduction . . . . .	52
Overlapping Domain Decomposition for Poisson's Equation . . . . .	53
Patching Domain Decomposition for Poisson's Equation . . . . .	67
5. POISSON'S EQUATION ON AN EL-SHAPED DOMAIN . . . . .	78

TABLE OF CONTENTS — Continued

	Page
Introduction . . . . .	78
Sinc-Galerkin Patching Method on an EI-Shaped Domain . . . . .	80
REFERENCES CITED . . . . .	99

## LIST OF TABLES

Table		Page
1	Error in the approximation of $u$ for Example 3.1 using the Sinc-Galerkin overlapping method . . . . .	29
2	Error in the approximation of $u$ for Example 3.2 using the Sinc-Galerkin overlapping method . . . . .	31
3	Error in the approximation of $u$ for Example 3.3 using the Sinc-Galerkin overlapping method . . . . .	32
4	Error in the Sinc-Galerkin overlapping approximation of $u$ for Example 3.3 with $M^1 = 64$ and $h^1 = .27768$ on the left-hand subinterval . . . .	33
5	Error in the approximation of $u$ for Example 3.4 using the sinc-collocation overlapping method . . . . .	35
6	Error in the approximation of $u$ for Example 3.5 using the sinc-collocation overlapping method . . . . .	37
7	Error in the approximation of $u$ for Example 3.6 using the sinc-collocation overlapping method . . . . .	38
8	Error in the sinc-collocation overlapping approximation of $u$ for Example 3.6 with $M^1 = 64$ and $h^1 = .27768$ on the left-hand subinterval . .	39
9	Error in the approximation of $u$ for Example 3.7 using the Sinc-Galerkin patching method . . . . .	43
10	Error in the approximation of $u$ for Example 3.8 using the Sinc-Galerkin patching method . . . . .	45
11	Error in the approximation of $u$ for Example 3.9 using the Sinc-Galerkin patching method . . . . .	46
12	Error in the approximation of $u$ for Example 3.9 using the Sinc-Galerkin patching method with $M^1 = 64$ and $h^1 = .27768$ . . . . .	47
13	Error in the approximation of $u$ for Example 3.10 using the sinc-collocation patching method . . . . .	48
14	Error in the approximation of $u$ for Example 3.11 using the sinc-collocation patching method . . . . .	50
15	Error in the approximation of $u$ for Example 3.12 using the sinc-collocation patching method . . . . .	50
16	Error in the approximation of $u$ for Example 3.12 using the sinc-collocation patching method with $M^1 = 64$ and $h^1 = .27768$ . . . . .	51
17	Error in the Sinc-Galerkin overlapping method for Example 4.1 . . . .	63
18	Error in the Sinc-Galerkin overlapping method for Example 4.2 . . . .	64
19	Error in the Sinc-Galerkin overlapping method for Example 4.3 . . . .	66



LIST OF TABLES — Continued

Table		Page
20	Error in the Sinc-Galerkin overlapping method for Example 4.3 with $M_x^2 = 2$ . . . . .	67
21	Error in the Sinc-Galerkin patching method for Example 4.4 . . . . .	74
22	Error in the Sinc-Galerkin patching method for Example 4.5 . . . . .	75
23	Error in the Sinc-Galerkin patching method for Example 4.6 . . . . .	76
24	Error in the Sinc-Galerkin patching method for Example 4.6 with $M_x^2 = 6$ . . . . .	77
25	Error in the Sinc-Galerkin patching method for Example 5.1 . . . . .	92
26	Error in the Sinc-Galerkin patching method for Example 5.2 . . . . .	94
27	Error in the Sinc-Galerkin patching method for Example 5.3 . . . . .	96
28	Error in the Sinc-Galerkin patching method for Example 5.3 with $M_x^1 = M_y^1 = 6$ . . . . .	97
29	Error in the Sinc-Galerkin patching method for Example 5.4 . . . . .	98

## LIST OF FIGURES

Figure		Page
1	The sinc translates $S(j, h)(x)$ for $h = \pi/4$ shown on $[-5, 5]$ . . . . .	6
2	The domain $D_S$ . . . . .	7
3	The domain $D_E$ for $d = \pi/3$ , $a = 0$ , and $b = 1$ . . . . .	10
4	The sinc basis functions $S_j(x)$ for $h = \pi/4$ shown on $(0, 1)$ . . . . .	12
5	The boundary basis function $\omega_1$ on the interval $\Omega^1 = (-1, 1)$ . . . . .	24
6	The boundary basis function $\omega_2$ on the interval $\Omega^2 = (.9, 4)$ . . . . .	24
7	Sinc-Galerkin overlapping solution for Example 3.1 . . . . .	29
8	Sinc-Galerkin overlapping solution for Example 3.3 . . . . .	32
9	Sinc-collocation overlapping solution for Example 3.4 . . . . .	36
10	Sinc-collocation overlapping solution for Example 3.6 . . . . .	38
11	Sinc-Galerkin patching solution for Example 3.7 . . . . .	44
12	Sinc-Galerkin patching solution for Example 3.9 . . . . .	46
13	Sinc-collocation patching solution for Example 3.10 . . . . .	49
14	Sinc-collocation patching solution for Example 3.12 . . . . .	51
15	Sinc-Galerkin overlapping solution for Example 4.1 with $M = 8$ . . . . .	62
16	Contour plots from the Sinc-Galerkin overlapping solution for Example 4.1 with contour lines at $u = .8, .6, .4$ , and $.2$ . . . . .	62
17	Sinc-Galerkin overlapping solution for Example 4.2 with $M = 8$ . . . . .	64
18	Contour plots from the Sinc-Galerkin overlapping solution for Example 4.2 with contour lines at $u = .9, .75$ , and $.6$ . . . . .	64
19	Sinc-Galerkin overlapping solution for Example 4.3 with $M = 8$ . . . . .	66
20	Contour plots from the Sinc-Galerkin overlapping solution for Example 4.3 with contour lines at $u = .8, .6, .4$ , and $.2$ . . . . .	66
21	Contour plots from the Sinc-Galerkin patching solution for Example 4.4 with contour lines at $u = .8, .6, .4$ , and $.2$ . . . . .	74
22	Contour plots from the Sinc-Galerkin patching solution for Example 4.5 with contour lines at $u = .9, .75$ , and $.6$ . . . . .	75
23	Contour plots from the Sinc-Galerkin patching solution for Example 4.6 with contour lines at $u = .8, .6, .4$ , and $.2$ . . . . .	77
24	The domain $\Omega$ . . . . .	79
25	The subdomains for $\Omega$ . . . . .	80
26	The rotated domain $\Omega$ . . . . .	91
27	Sinc-Galerkin patching solution for Example 5.1 with $M = 8$ . . . . .	92

LIST OF FIGURES — Continued

Figure		Page
28	Contour plots from the Sinc-Galerkin patching solution for Example 5.1	92
29	Sinc-Galerkin patching solution for Example 5.2 with $M = 8$ . . . . .	93
30	Contour plots from the Sinc-Galerkin patching solution for Example 5.2	94
31	Sinc-Galerkin patching solution for Example 5.3 with $M = 8$ . . . . .	96
32	Contour plots from the Sinc-Galerkin patching solution for Example 5.3	96
33	Sinc-Galerkin patching solution for Example 5.4 with $M = 8$ . . . . .	98
34	Contour plots from the Sinc-Galerkin patching solution for Example 5.4	98

## ABSTRACT

Sinc basis functions form a desirable basis to use for solving singular problems via domain decomposition. This is because both the Sinc-Galerkin and sinc-collocation methods converge exponentially, even in the presence of boundary singularities. For Poisson's equation with homogeneous Dirichlet boundary conditions posed on a rectangular domain, the Sinc-Galerkin and sinc-collocation methods have been well developed. The sinc methods have also been developed for any domain which can be mapped to a rectangular domain via an invertible or conformal mapping. In order to increase the number and complexity of domains which can be handled via sinc methods, domain decomposition techniques are used.

The Sinc-Galerkin and sinc-collocation domain decomposition methods are first studied for a two-point boundary-value problem. Both of the traditional methods of domain decomposition, overlapping and patching, are developed. This lays the groundwork to readily determine which method is most suited to any given problem. Because the goal is to clearly develop and test the sinc domain decomposition methods, techniques such as subdomain iterations and preconditioning are not employed here. The number of subdomains is limited to two in order to limit the complexity of the presentation. Numerical results are presented for both decomposition methods that exhibit the nearly identical errors achieved whether one uses the sinc-collocation or Sinc-Galerkin method.

Next the patching and overlapping Sinc-Galerkin methods are presented for Poisson's equation presented on a rectangle. For certain parameter choices the sinc-collocation system is identical for these problems, and is thus not presented separately. Again the number of subdomains is limited to two in order to present the material more clearly. Both domain decomposition methods perform well, and this is highlighted in the numerical examples.

Finally, Poisson's equation is studied on an  $el$ -shaped domain. In the derivation of the discrete system, it becomes evident that the patching domain decomposition method is the method of choice for this problem. The derivation and numerical examples are presented using three subdomains, although multiple subdomains could certainly be used. Numerical examples illustrate the ability of this method to handle these types of problems.

## CHAPTER 1

## INTRODUCTION

The subject of this thesis is the solution of Poisson's equation via domain decomposition coupled with sinc methods. For a general domain  $\Omega \in \mathbb{R}^2$ , Poisson's equation with homogeneous Dirichlet boundary conditions is given by

$$\begin{aligned} -\nabla^2 u(x, y) &\equiv -\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega \\ u(x, y) &= 0, \quad (x, y) \in \partial\Omega. \end{aligned} \tag{1.1}$$

Many methods for Poisson's equation have been highly developed. If  $\Omega$  is a rectangle, for example, using a finite difference method leads to a sparse matrix system which can be efficiently solved with specialized techniques for banded matrices. One such difference method has a quadratic rate of convergence depending on the properties of the fourth-order partials of the solution. See [3] for a development of this method and numerical results. Similarly, one can develop a finite element method (see [12]) that has a quadratic rate of convergence depending on the properties of the second-order partials of the solution. The matrix system has the same structure as that arising from a centered difference method.

Sinc methods for Poisson's equation on a rectangle have been well-studied in [2], [17], [18], [21], [23], and [27]. They are desirable methods to use for a variety of reasons. First, the sinc approximations converge exponentially to the true solution. The tradeoff for this rate of convergence is a full matrix system, as is true of most spectral methods. Since the procedure is a product method, its most convenient application occurs when  $\Omega = I \times J$ , where  $I$  and  $J$  are intervals.

Each method for solving partial differential equations excels on a particular class of problems. For example, finite differences yield simple methods which work well for problems with analytic solutions. The finite element methods work well for these problems and can more easily handle complicated geometries and boundary conditions. The sinc methods excel for problems with boundary singularities, as discussed in [27]. The convergence estimates for both the finite difference and finite element methods depend on the smoothness of the partial derivatives of the solution. Thus there is no guarantee that they will perform well on problems with any singularity. This is also true of other spectral methods. Thus the sinc methods are in a class of their own when it comes to dealing with boundary singularities.

If the domain  $\Omega$  is not a rectangle there are two basic methods for the discretization of (1.1). The first method involves redeveloping the discrete system and the error analysis for each new domain. Finite element methods take this approach. The second approach invokes a change of coordinates to exchange the domain  $\Omega$  for a domain on which the numerical method has been previously developed. In this case, the Laplacian becomes a more general elliptic operator. This method has received less attention than it warrants due to the fact that the coefficients of the transformed Laplacian may be singular. This has no effect on the sinc implementation or resulting calculations and accuracy. The method developed in this thesis, in combination with both of the above methods, is advocated in [27], and handles any domain whose boundary consists of finitely many analytic arcs.

Extensions to more general domains seem possible only if the domain can be split into two or more pieces, each of which could be mapped to its own rectangle. The solutions in each subdomain must then be matched in some manner. This leads to a need for general domain decomposition methods.

Domain decomposition techniques have been of great interest lately, especially

with the advancement of parallel computing technology. A series of conferences on domain decomposition methods began in Paris in 1987 with the First International Symposium on Domain Decomposition Methods for Partial Differential Equations. The proceedings from each of these conferences is a good source of information on domain decomposition. For example, [9] contains the proceedings from the conference held in Moscow in May, 1990. Applications of these methods include field-scale simulations of fluid flow in porous media and two-dimensional convection-diffusion problems. See [5] and [8] for more details on these applications. By breaking these large-scale problems into multiple subproblems, parallel processors may be used to efficiently solve these problems using iterative techniques.

Having made the decision to decompose the domain  $\Omega$ , there are two traditional methods of handling the decomposition: patching and overlapping. When the problem at hand does not motivate one method over another it is natural to ask which method is preferable. Such comparisons must include a measure of accuracy balanced with respect to implementation considerations. In certain cases, the two methods can be shown to be related, if not identical. See [6] for more details. When iterative procedures are used to solve the problems on parallel computers, the patching method has a lower overhead cost. On the other hand, the overlapping method is considered to be more robust (see [5]). Due to the potential advantages of each method, this thesis will carry out the discretization using both decompositions for the sinc basis.

As in any product method, there is a clarity of presentation imported by first fully understanding the implementation of the procedure for the one-dimensional problem. For this reason, the patching and overlapping methods are carried out for the sinc methods on an interval. Sinc methods here refer to both the Sinc-Galerkin and sinc-collocation procedures, which are introduced in Chapter 2. These procedures complement one another and provide the link to establish (numerically) the

convergence of the procedure. These two methods are spelled out in Chapter 3 and the examples included show that, with respect to accuracy and implementation, they are numerically equivalent.

Chapter 4 presents both the patching and overlapping methods for Poisson's equation on a rectangle. In this chapter, only two subdomains are used. Because the Sinc-Galerkin and sinc-collocation methods are the same for Poisson's equation with an appropriate choice of weights, only the Sinc-Galerkin method is discussed. Again, both the patching and overlapping methods perform equally well, as seen in the examples.

Chapter 5 addresses the solution of Poisson's equation on an el-shaped domain. This is the final tool needed for solving such equations on more complex domains. The method used for the development of the discrete system mandates that the subdomains must not overlap, and at least three subdomains must be used for this domain. Thus the patching method is developed for use with three subdomains, though multiple subdomains are possible. The numerical results are quite good for this method.



## CHAPTER 2

### SINC METHODS FOR DIFFERENTIAL EQUATIONS

#### Introduction

Sinc methods for differential equations have been well-studied since their introduction in [25]. They have been applied to a variety of differential equations such as two-point boundary-value problems, Poisson's equation, the wave equation, the heat equation, the advection-diffusion equation, and Burgers' equation. Both the Sinc-Galerkin and sinc-collocation methods are well-suited for problems with boundary singularities. They also both converge exponentially, even in the presence of such boundary singularities. For an overview of sinc methods for differential equations see [17], [26], and [27].

The second section presents an introduction to sinc interpolation and quadrature methods. These are necessary tools for deriving the Sinc-Galerkin and the sinc-collocation methods for solving differential equations. For problems with constant coefficients, the Sinc-Galerkin method might well be the method of choice. For problems with variable coefficients, the sinc-collocation method is especially convenient because the coefficients are more efficiently handled. In order to leave a clear path for future work, both methods are presented here. The Sinc-Galerkin method is developed in the third section and the sinc-collocation method is derived in the fourth section.

## Sinc Interpolation and Quadrature Methods

The following sinc interpolation and quadrature results are presented in detail in [17], [26], and [27]. For  $h > 0$  and any integer  $j$ , define the sinc translates on the real line by

$$S(j, h)(x) \equiv \operatorname{sinc}\left(\frac{x - jh}{h}\right)$$

where for  $z \in \mathbb{C}$

$$\operatorname{sinc}(z) = \begin{cases} \frac{\sin(\pi z)}{\pi z}, & z \neq 0 \\ 1, & z = 0 \end{cases}.$$

Three examples of these translates are shown in Figure 1.

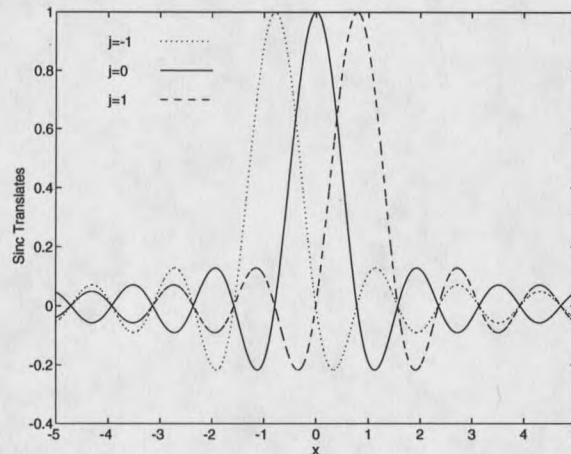


Figure 1: The sinc translates  $S(j, h)(x)$  for  $h = \pi/4$  shown on  $[-5, 5]$

In order to present the interpolation and quadrature results on the real line, the Paley-Wiener class of functions is defined as follows.

**Definition 2.1** *Let  $h$  be a positive constant. The Paley-Wiener class of functions  $B(h)$  is the family of entire functions  $f$  such that on the real line  $f \in L^2(\mathbb{R})$  and in the complex plane  $f$  is of exponential type  $\pi/h$ , i.e., there exists  $K > 0$  so that*

$$|f(z)| \leq K \exp(\pi|z|/h)$$

for all  $z \in \mathbb{C}$ .

The sinc interpolation and quadrature methods are exact on members of the Paley-Wiener class of functions, as seen in Theorem 2.1.

**Theorem 2.1** *If  $f \in B(h)$ , then for all  $z \in \mathbb{C}$ ,*

$$f(z) = \sum_{k=-\infty}^{\infty} f(kh)S(k, h)(z) .$$

*Furthermore, if  $f \in L^1(\mathbb{R})$ ,*

$$\int_{-\infty}^{\infty} f(u) du = h \sum_{k=-\infty}^{\infty} f(kh) .$$

See [22] and [24] for the proof of the first and second parts, respectively, of this theorem.

The Paley-Wiener class of functions is quite restrictive. For practical applications, a larger class of functions on which these methods perform well is desirable. Define the infinite strip  $D_S$  by

$$D_S \equiv \left\{ w = u + iv : |v| < d \leq \frac{\pi}{2} \right\} . \quad (2.1)$$

$D_S$  is shown in Figure 2.

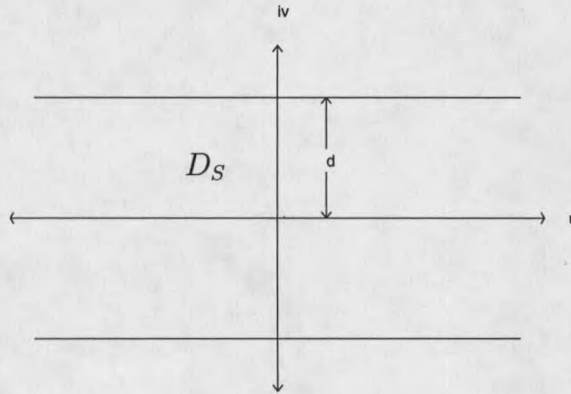


Figure 2: The domain  $D_S$

In order to extend the sinc interpolation and quadrature results, define the class of functions  $B^p(D_S)$ .

**Definition 2.2** Let  $B^p(D_S)$  be the set of functions analytic in  $D_S$  that satisfy

$$\int_{-d}^d |f(u + iv)| dv = \mathcal{O}(|u|^a), \quad u \rightarrow \pm\infty, \quad 0 \leq a < 1$$

and

$$\begin{aligned} N^p(f, D_S) &\equiv \lim_{v \rightarrow d^-} \left\{ \left( \int_{-\infty}^{\infty} |f(u + iv)|^p du \right)^{1/p} \right. \\ &\quad \left. + \left( \int_{-\infty}^{\infty} |f(u - iv)|^p du \right)^{1/p} \right\} \\ &< \infty. \end{aligned}$$

As seen in Theorem 2.2, proven in [17], the sinc interpolation and quadrature methods perform quite well on this class of functions.

**Theorem 2.2** If  $f \in B^p(D_S)$ ,  $p = 1$  or  $2$  then there exists a positive constant  $K_1$  such that

$$\left\| f - \sum_{k=-\infty}^{\infty} f(kh)S(k, h) \right\|_{\infty} \leq K_1 \exp(-\pi d/h).$$

Additionally, if  $p = 1$  then there exists a positive constant  $K_2$  such that

$$\left| \int_{-\infty}^{\infty} f(u) du - h \sum_{k=-\infty}^{\infty} f(kh) \right| \leq K_2 \exp(-2\pi d/h).$$

In practice, only finite sums can be calculated. The effect of truncation can be minimized by assuming appropriate growth conditions on  $f$ , as summarized in the following theorem proved in [24].

**Theorem 2.3** Assume  $f \in B^p(D_S)$  for  $p = 1$  or  $2$  and that there are positive constants  $\alpha$ ,  $\beta$ , and  $C$  so that

$$|f(u)| \leq C \begin{cases} \exp(-\alpha|u|), & u \in (-\infty, 0) \\ \exp(-\beta u), & u \in [0, \infty) \end{cases}$$

Make the selections

$$N = \left\lceil \left\lceil \frac{\alpha}{\beta} M + 1 \right\rceil \right\rceil$$

and

$$h = \left( \frac{\pi d}{\alpha M} \right)^{1/2},$$

where  $[\cdot]$  denotes the greatest integer function. Then there exists  $K_3 > 0$ , independent of  $M$ , so that

$$\left\| f - \sum_{k=-M}^N f(kh)S(k, h) \right\|_{\infty} \leq K_3 \exp(-(\pi d \alpha M)^{1/2}).$$

Additionally, if  $p = 1$  and

$$h = \left( \frac{2\pi d}{\alpha M} \right)^{1/2},$$

then there exists  $K_4 > 0$  independent of  $M$  so that

$$\left| \int_{-\infty}^{\infty} f(x)dx - h \sum_{k=-M}^N f(kh) \right| \leq K_4 \exp(-2\pi d \alpha M)^{1/2}.$$

To solve problems on the finite interval  $(a, b)$ , use the conformal map

$$\phi(z) = \ln \left( \frac{z-a}{b-z} \right). \quad (2.2)$$

This map carries the eye-shaped region

$$D_E \equiv \left\{ z = x + iy : \left| \arg \left( \frac{z-a}{b-z} \right) \right| < d \leq \frac{\pi}{2} \right\} \quad (2.3)$$

onto the infinite strip  $D_S$  in (2.1). An example of  $D_E$  is shown in Figure 3.

To describe the sinc quadrature and interpolation rules, the function space  $B(D_E)$  is defined as follows.

**Definition 2.3** Let  $D_E$  be the domain described in (2.3) in the  $z = x + iy$  plane with boundary points  $a \neq b$  on the real line. Let  $w = \phi(z)$  be the conformal map of  $D_E$  onto the infinite strip  $D_S$  given in (2.2). Denote by  $z = \tau(w)$  the inverse of the mapping  $\phi$  and let

$$\Gamma \equiv \{z \in \mathbb{C} : z = \tau(u), u \in \mathbb{R}\} = \tau(\mathbb{R}) = (a, b).$$

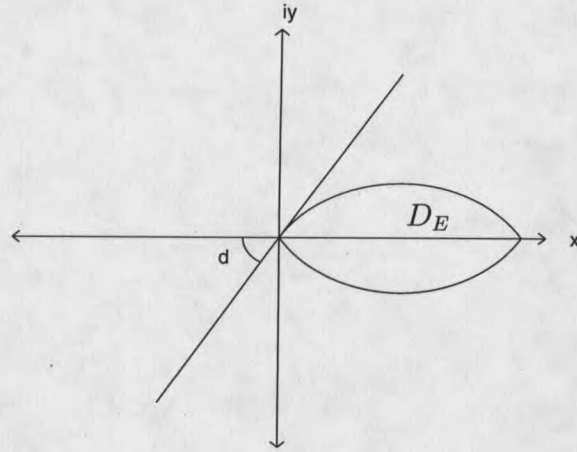


Figure 3: The domain  $D_E$  for  $d = \pi/3$ ,  $a = 0$ , and  $b = 1$

Then  $B(D_E)$  is defined as the class of functions analytic in  $D_E$  which satisfy for some constant  $a$  with  $0 \leq a < 1$ ,

$$\int_{\tau(u+L)} |F(z)dz| = \mathcal{O}(|u|^a), \quad u \rightarrow \pm\infty$$

where  $L = \{iv : |v| < d\}$  and for  $\gamma$  a simple closed contour in  $D_E$

$$N(F, D_E) \equiv \lim_{\gamma \rightarrow \partial D_E} \int_{\gamma} |F(z)dz| < \infty.$$

Further, for  $h > 0$  and  $k = 0, \pm 1, \pm 2, \dots$ , denote the nodes

$$x_k \equiv \tau(kh) = \phi^{-1}(kh) = \frac{a + be^{kh}}{e^{kh} + 1}. \quad (2.4)$$

The sinc quadrature rule on  $(a, b)$  is contained in the following result.

**Theorem 2.4** Let  $F \in B(D_E)$  with  $h > 0$ , let  $\phi$  be the one-to-one conformal map given in (2.2), and let  $x_k = \phi^{-1}(kh)$  be as given in (2.4). Assume that there are positive constants  $\alpha$ ,  $\beta$ , and  $C$ , so that

$$|F(x)| \leq C \begin{cases} (x-a)^{\alpha-1}, & x \in (a, (a+b)/2) \\ (b-x)^{\beta-1}, & x \in [(a+b)/2, b) \end{cases}.$$

Make the selections

$$N = \left\lceil \left\lfloor \frac{\alpha}{\beta} M + 1 \right\rfloor \right\rceil$$

and

$$h = \left( \frac{2\pi d}{\alpha M} \right)^{1/2}$$

Then there exists a constant  $K_5 > 0$ , independent of  $M$ , so that

$$\left| \int_a^b F(x) dx - h \sum_{k=-M}^N \frac{F(x_k)}{\phi'(x_k)} \right| \leq K_5 \exp \left( -(2\pi d \alpha M)^{1/2} \right).$$

### Two-Point Boundary-Value Problem

The linear two-point boundary-value problem with homogeneous Dirichlet boundary conditions on the finite interval  $(a, b)$  is given by

$$\begin{aligned} \mathcal{L}u(x) &\equiv -u''(x) + p(x)u'(x) + q(x)u(x) \\ &= f(x), \quad a < x < b \\ u(a) &= u(b) = 0. \end{aligned} \tag{2.5}$$

Sinc methods for problems of this type are discussed in detail in [16], [17], [25], and [26].

The basis functions used in solving (2.5) are defined by

$$S_j(x) = S(j, h) \circ \phi(x), \tag{2.6}$$

where  $\phi$  is given in (2.2). A graph of these basis functions can be seen in Figure 4.

The approximate solution is given by

$$u_m(x) = \sum_{k=-M}^N u_k S_k(x), \quad m = M + N + 1, \tag{2.7}$$

where  $h > 0$  is fixed.

### The Sinc-Galerkin Method

The Sinc-Galerkin method for (2.5) is clearly developed in [16], [17], [25], and [26]. The Galerkin method requires orthogonalizing the residual  $\mathcal{L}u_m - f$  against each

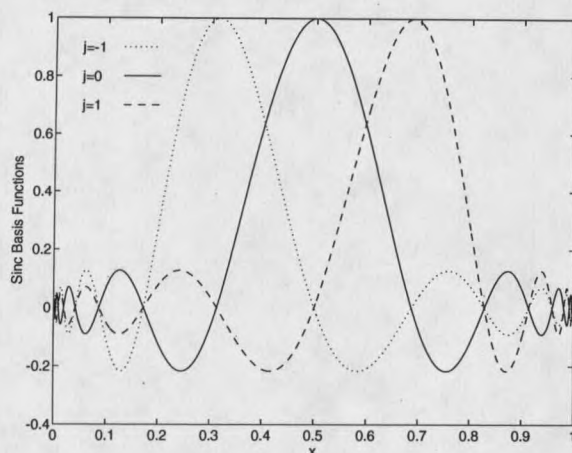


Figure 4: The sinc basis functions  $S_j(x)$  for  $h = \pi/4$  shown on  $(0, 1)$

basis function using a weighted  $L^2$  inner product. To simplify notation, orthogonalize  $\mathcal{L}u - f$  against each basis function yielding

$$(\mathcal{L}u - f, S_j) = 0, \quad -M \leq j \leq N,$$

where

$$(f, g) = \int_a^b f(x)g(x)\omega(x)dx$$

and

$$\omega(x) = (\phi'(x))^{-r}, \quad r \geq 0.$$

Thus for  $-M \leq j \leq N$ ,

$$\int_a^b (-u''(x) + p(x)u'(x) + q(x)u(x) - f(x)) S_j(x)\omega(x)dx = 0.$$

Integrating by parts to remove all derivatives from  $u$  yields

$$\begin{aligned} \int_a^b f(x)S_j(x)\omega(x)dx &= - \int_a^b u(x) (S_j\omega)''(x)dx \\ &- \int_a^b u(x) (pS_j\omega)'(x)dx + \int_a^b u(x)q(x)S_j(x)\omega(x)dx + BT \end{aligned} \tag{2.8}$$

where

$$BT = (upS_j\omega)(x) \Big|_a^b - (u'S_j\omega)(x) \Big|_a^b + (u(S_j\omega)')(x) \Big|_a^b.$$



The exponent  $r$  in the weight function  $\omega$  can be chosen so that the boundary term  $BT$  disappears.

The following notation will be necessary for writing down the discrete Sinc-Galerkin and sinc-collocation systems. For a function  $\gamma$  defined on  $(a, b)$  denote by  $\vec{\gamma}$  the  $m \times 1$  vector

$$\vec{\gamma} = [\gamma(x_{-M}) \dots \gamma(x_N)]^T, \quad (2.9)$$

and let  $\mathcal{D}(\gamma)$  be the  $m \times m$  diagonal matrix

$$\mathcal{D}(\gamma) = \begin{bmatrix} \gamma(x_{-M}) & & \\ & \dots & \\ & & \gamma(x_N) \end{bmatrix}.$$

Also define

$$\delta_{jk}^{(0)} \equiv [S(j, h) \circ \phi(x)] \Big|_{x=x_k} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases},$$

$$\delta_{jk}^{(1)} \equiv h \frac{d}{d\phi} [S(j, h) \circ \phi(x)] \Big|_{x=x_k} = \begin{cases} 0, & k = j \\ \frac{(-1)^{k-j}}{k-j}, & k \neq j \end{cases},$$

$$\delta_{jk}^{(2)} \equiv h^2 \frac{d^2}{d\phi^2} [S(j, h) \circ \phi(x)] \Big|_{x=x_k} = \begin{cases} \frac{-\pi^2}{3}, & k = j \\ \frac{-2(-1)^{k-j}}{(k-j)^2}, & k \neq j \end{cases},$$

and define the  $m \times m$  matrices  $I^{(p)}$  for  $p = 0, 1, 2$  by

$$I^{(p)} = [\delta_{jk}^{(p)}], \quad j, k = -M, \dots, N.$$

Applying the sinc quadrature rule to (2.8) and using this notation yields the following set of  $m$  equations where  $-M \leq j \leq N$

$$\begin{aligned} 0 &= h \sum_{k=-M}^N u(x_k) \left[ \frac{\delta_{jk}^{(2)}}{h^2} (\phi' \omega)(x_k) + \frac{\delta_{jk}^{(1)}}{h} \left( \frac{\phi''}{\phi'} \omega + 2\omega' \right) (x_k) \right] \\ &+ h \left( \frac{\omega'' u}{\phi'} \right) (x_j) + h \sum_{k=-M}^N (u p \omega)(x_k) \frac{\delta_{jk}^{(1)}}{h} \\ &+ h \left( \frac{u(p\omega)'}{\phi'} \right) (x_j) - h \left( \frac{q\omega u}{\phi'} \right) (x_j) + h \left( \frac{f\omega}{\phi'} \right) (x_j) \\ &+ \mathcal{O} \left( M \exp \left( -(\pi d \alpha M)^{1/2} \right) \right). \end{aligned} \quad (2.10)$$

Now replace  $u$  by  $u_m$  as required for the Galerkin method and drop the error term.

Notice that  $u_m(x_k) = u_k$  and  $\omega = (\phi')^{-r}$ . Then (2.10) becomes

$$\begin{aligned}
0 &= h \sum_{k=-M}^N u_k \left[ \frac{\delta_{jk}^{(2)}}{h^2} (\phi')^{1-r}(x_k) + \frac{\delta_{jk}^{(1)}}{h} \left( \frac{\phi''}{(\phi')^{1+r}} + 2((\phi')^{-r})' \right) (x_k) \right] \\
&+ u_j h \left( \frac{((\phi')^{-r})''}{\phi'} \right) (x_j) + h \sum_{k=-M}^N u_k (p(\phi')^{-r})(x_k) \frac{\delta_{jk}^{(1)}}{h} \\
&+ u_j h \left( \frac{(p(\phi')^{-r})'}{\phi'} \right) (x_j) - u_j h \left( \frac{q}{(\phi')^{1+r}} \right) (x_j) + h \left( \frac{f}{(\phi')^{1+r}} \right) (x_j).
\end{aligned} \tag{2.11}$$

Using the notation given above to write (2.11) as a matrix equation yields the discrete Sinc-Galerkin system, whose coefficient matrix is

$$\begin{aligned}
\Gamma_r &= \left\{ A \left( \frac{1}{(\phi')^r} \right) - \frac{1}{h} I^{(1)} \mathcal{D} \left( \frac{p}{\phi'} \right) + \mathcal{D} \left( \frac{-1}{(\phi')^{2-r}} \left( \frac{p}{(\phi')^r} \right)' \right) \right. \\
&\quad \left. + \mathcal{D} \left( \frac{q}{(\phi')^2} \right) \right\} \mathcal{D} \left( \frac{1}{(\phi')^{r-1}} \right).
\end{aligned} \tag{2.12}$$

The second derivative matrix in (2.12) is given by

$$\begin{aligned}
A \left( \frac{1}{(\phi')^r} \right) &= \frac{-1}{h^2} I^{(2)} - \frac{1}{h} I^{(1)} \mathcal{D} \left( (1-2r) \frac{\phi''}{(\phi')^2} \right) \\
&+ \mathcal{D} \left( \left( \frac{-1}{(\phi')^{2-r}} \right) \left( \frac{1}{(\phi')^r} \right)'' \right).
\end{aligned}$$

The discrete Sinc-Galerkin system for (2.5) is then given by

$$\Gamma_r \vec{u} = \mathcal{D} \left( \frac{1}{(\phi')^{r+1}} \right) \vec{f} \tag{2.13}$$

where

$$\vec{u} = [u_{-M} \dots u_N]^T \tag{2.14}$$

and  $\vec{f}$  is defined in (2.9) with  $\gamma$  replaced by  $f$ . The following theorem for the convergence of this method in the case  $p(x) \equiv 0$  is proven in [25].

**Theorem 2.5** Let the numbers  $u_k$  ( $k = -M, \dots, N$ ) be determined by (2.13), and let  $u_m(x)$  be as defined in (2.7). Then assume  $f/\sqrt{\phi'}$ ,  $u(\phi')^2$ ,  $uq/\sqrt{\phi'}$   $\in B(D_E)$  and that there exist positive constants  $\alpha$ ,  $\beta$ ; and  $C$  so that

$$|u(x)| \leq C \begin{cases} (x-a)^{\alpha+1/2}, & x \in (a, (a+b)/2) \\ (b-x)^{\beta+1/2}, & x \in [(a+b)/2, b) \end{cases}$$

Choose  $r = 1/2$ ,  $h = (\pi d/(\alpha M))^{1/2}$ , and

$$N = \left\lceil \left\lfloor \frac{\alpha}{\beta} M + 1 \right\rfloor \right\rceil$$

Then

$$\|u_m - u\|_{\infty} \leq CM^2 e^{-\sqrt{\pi d \alpha M}}$$

where  $u$  is the solution of (2.5) with  $p(x) \equiv 0$ .

### The Sinc-Collocation Method

The sinc-collocation method for the problem (2.5) is discussed in detail in [1], [7], [11], [17], and [19]. Collocation requires orthogonalizing the residual against dirac delta functions centered at the sinc nodes

$$\delta_j(x) \equiv \delta(x - x_j).$$

Again for simplification of notation, first apply this method to  $\mathcal{L}u - f$  to get

$$\langle \mathcal{L}u - f, \delta_j \rangle = 0.$$

Using the  $L^2$  inner product

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

yields the equation

$$\int_a^b (\mathcal{L}u - f)(x)\delta_j(x)dx = 0.$$

Recalling properties of the dirac delta function found in [13] leaves the equations

$$(\mathcal{L}u - f)(x_j) = 0.$$

The resulting scheme is defined by the  $m$  equations ( $m = M + N + 1$ )

$$\mathcal{L}u_m(x_j) = f(x_j), \quad j = -M, \dots, N. \quad (2.15)$$

Because the approximation (2.7) used in the Sinc-Galerkin method is not differentiable at  $x = a$  and at  $x = b$ , a weighted approximation is needed. This new approximation is given by

$$u_m(x) = \sum_{k=-M}^N u_k (\phi'(x_k))^l \frac{S_k(x)}{(\phi'(x))^l}. \quad (2.16)$$

Substituting into (2.15) yields

$$\sum_{k=-M}^N u_k (\phi'(x_k))^l \left[ - \left( \frac{S_k}{(\phi')^l} \right)'' (x_j) + \left( p \left( \frac{S_k}{(\phi')^l} \right)' \right) (x_j) + \left( q \left( \frac{S_k}{(\phi')^l} \right) \right) (x_j) \right] = f(x_j).$$

Expanding the derivative terms yields

$$\begin{aligned} & \sum_{k=-M}^N u_k (\phi'(x_k))^l \left[ - \frac{\delta_{kj}^{(2)}}{h^2} (\phi')^{(2-l)}(x_j) - \frac{\delta_{kj}^{(1)}}{h} (1 - 2l) \left( \frac{\phi''}{(\phi')^l} \right) (x_j) \right. \\ & - \delta_{kj}^{(0)} \left( \frac{1}{(\phi')^l} \right)'' (x_j) + \frac{\delta_{kj}^{(1)}}{h} (p (\phi')^{1-l})(x_j) + \delta_{kj}^{(0)} \left( \frac{-lp\phi''}{(\phi')^{1+l}} \right) (x_j) \\ & \left. + (q (\phi')^{-l})(x_j) \right] \\ & = f(x_j). \end{aligned}$$

The fundamental matrices for the discrete sinc-collocation system are

$$\begin{aligned} C(l) = & \left\{ \frac{-1}{h^2} I^{(2)} - \frac{1}{h} \mathcal{D} \left( (2l - 1) \frac{\phi''}{(\phi')^2} + \frac{p}{\phi'} \right) I^{(1)} \right. \\ & \left. + \mathcal{D} \left( \left( \frac{-1}{(\phi')^{2-l}} \right) \left( \frac{1}{(\phi')^l} \right)'' - \frac{l\phi''p}{(\phi')^3} + \frac{q}{(\phi')^2} \right) \right\} \end{aligned}$$

and

$$\mathcal{K}_l = C(l) \mathcal{D}((\phi')^l). \quad (2.17)$$

The resulting discrete sinc-collocation system is given by

$$\mathcal{K}_l \vec{u} = \mathcal{D} \left( \frac{1}{(\phi')^{2-l}} \right) \vec{f} \quad (2.18)$$

where  $\vec{u}$  is given in (2.14) and  $\vec{f}$  is given by (2.9) with  $\gamma$  replaced by  $f$ . Notice that if  $p(x) \equiv 0$  and  $l = 1/2$  then this system is the same as the Sinc-Galerkin system (2.13) for  $r = 1/2$ , i.e.,  $\Gamma_{1/2} = \mathcal{K}_{1/2}$ . Also notice that the coefficients  $p$  and  $q$  are not differentiated in the sinc-collocation method, but they are in the Sinc-Galerkin method. For this reason the sinc-collocation method is advantageous for problems with variable coefficients. The following theorem concerning the convergence of this method for the case  $p(x) \equiv 0$  and  $q(x) \geq \delta > 0$  is found in [1].

**Theorem 2.6** *Let the numbers  $u_k$  ( $k = -M, \dots, N$ ) be determined by (2.18) and let  $u_m(x)$  be as defined in (2.16). Then assume  $u(\phi')^{3/2} \in B(D_E)$  and that there exist positive constants  $\alpha, \beta$ , and  $C$  so that*

$$|u(x)| \leq C \begin{cases} (x-a)^{\alpha+1/2}, & x \in (a, (a+b)/2) \\ (b-x)^{\beta+1/2}, & x \in [(a+b)/2, b) \end{cases}$$

Choose  $l = 1/2$ ,  $h = (\pi d / (\alpha M))^{1/2}$ , and

$$N = \left\lceil \left\lfloor \frac{\alpha}{\beta} M + 1 \right\rfloor \right\rceil.$$

Then

$$\|u_m - u\|_{\infty} \leq CM^{3/2} e^{-\sqrt{\pi d \alpha M}}$$

where  $u$  is the solution of (2.5) with  $p(x) \equiv 0$  and  $q(x) \geq \delta > 0$ .

### Poisson's Equation on a Rectangle

Let  $\Omega$  be the rectangular region  $\{(x, y) : a < x < b, s < y < t\}$ . Let  $\partial\Omega$  be the boundary of  $\Omega$ . Poisson's equation with homogeneous Dirichlet boundary conditions is given by

$$\begin{aligned}
-\nabla^2 u(x, y) &\equiv -\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega \\
u(x, y) &= 0, \quad (x, y) \in \partial\Omega.
\end{aligned} \tag{2.19}$$

The sinc methods use a product of bases for the basis in two dimensions. The following discussion only addresses the Sinc-Galerkin method. A similar approach works for the sinc-collocation method. The complete development can be found in [2], [17], [18], and [20]. There are several ways to achieve the Sinc-Galerkin and the sinc-collocation systems for (2.19). Following the traditional Galerkin development, assume an approximation of the form

$$u_{m_x, m_y}(x, y) = \sum_{j=-M_x}^{N_x} \sum_{k=-M_y}^{N_y} u_{jk} S_{jk}(x, y) \tag{2.20}$$

where

$$S_{jk}(x, y) = S_j(x) S_k(y).$$

Here  $m_x = M_x + N_x + 1$ ,  $m_y = M_y + N_y + 1$ ,  $S_j(x) = S(j, h) \circ \phi(x)$ , as given in (2.6), and

$$S_k(y) = S(k, h) \circ \psi(y). \tag{2.21}$$

The conformal map  $\phi$  is given in (2.2), and the map  $\psi$  is given by

$$\psi(z) = \ln \left( \frac{z-s}{t-z} \right). \tag{2.22}$$

Using a weighted  $L^2$  inner product

$$(f, g) = \int_s^t \int_a^b f(x, y) g(x, y) (\phi'(x) \psi'(y))^{-r} dx dy,$$

orthogonalize the residual (again using  $u$  to simplify the notation) against each basis function

$$(\mathcal{L}u - f, S_{jk}) = 0,$$

for  $-M_x \leq j \leq N_x$  and  $-M_y \leq k \leq N_y$ . Perform integration by parts to remove all derivatives from the  $u$  terms. Then apply the sinc quadrature rule as necessary and replace  $u$  by  $u_{m_x, m_y}$ . See [17] for the painfully lengthy details.

The above approach is simple but quite messy. An approach found in [17] which is useful in deriving the domain decomposition method is given below. This method is more straightforward than the traditional approach and the resulting systems are identical.

Fix  $x = x_p$ . Along this line, Poisson's equation implies that

$$-u_{xx}(x_p, y) - u_{yy}(x_p, y) = f(x_p, y).$$

Since  $x$  is fixed, rewrite the equation as

$$-u_{yy}(x_p, y) = f(x_p, y) + u_{xx}(x_p, y).$$

Notice that the boundary conditions imply

$$u(x_p, s) = u(x_p, t) = 0.$$

This is now a two-point boundary-value problem like those discussed in the second section. Thus the resulting system for each  $x_p$  looks like

$$\Gamma_r \vec{u}(x_p, \vec{y}) = \mathcal{D} \left( \frac{1}{(\psi')^{r+1}} \right) \left( \vec{f}(x_p, \vec{y}) + \vec{u}_{xx}(x_p, \vec{y}) \right).$$

Allowing  $p$  to vary, the overall system is given by

$$\Gamma_r U^T = \mathcal{D} \left( \frac{1}{(\psi')^{r+1}} \right) (F^T + U_{xx}^T).$$

Transposing each side and multiplying on the right by the diagonal matrix yields the equation

$$U \Gamma_r^T \mathcal{D} \left( (\psi')^{r+1} \right) = (F + U_{xx}). \quad (2.23)$$

Here  $U = [u(x_j, y_k)]$  and  $F = [f(x_j, y_k)]$  are the  $m_x \times m_y$  matrices formed from point evaluation of  $u$  and  $f$ , respectively. The  $m_x \times m_y$  matrix  $U_{xx} = [u_{xx}(x_j, y_k)]$  is the point evaluation of the second partial derivative of  $u$  with respect to  $x$ , and the diagonal matrix is of size  $m_y \times m_y$ .

Similarly, fix  $y = y_q$ . The resulting two-point boundary-value problem is

$$-u_{xx}(x, y_q) = f(x, y_q) + u_{yy}(x, y_q)$$

with boundary conditions

$$u(a, y_q) = u(b, y_q) = 0.$$

Apply the Sinc-Galerkin method to this new boundary-value problem and let  $q$  vary to get the matrix equation

$$\Gamma_r U \mathcal{D} \left( (\psi')^{r+1} \right) = (F + U_{yy}). \quad (2.24)$$

Adding (2.23) and (2.24) yields the Sylvester equation

$$\begin{aligned} \Gamma_r U \mathcal{D} \left( (\psi')^{r+1} \right) + U \Gamma_r^T \mathcal{D} \left( (\psi')^{r+1} \right) &= (F + U_{yy} + F + U_{xx}) \\ &= F. \end{aligned} \quad (2.25)$$

There are many approaches to solving equations of this type. Because of the work to come in domain decomposition, the approach taken will be to concatenate each side of (2.25). The following two definitions will be helpful in simplifying the discrete Sinc-Galerkin system. More details can be found in [14] and [17].

**Definition 2.4** For a matrix  $B = (b_{ij})$ ,  $1 \leq i \leq m$ ,  $1 \leq j \leq n$ , the concatenation of  $B$  is the  $mn \times 1$  vector

$$\text{co}(B) \equiv \begin{bmatrix} \vec{b}_{i1} \\ \vec{b}_{i2} \\ \vdots \\ \vec{b}_{in} \end{bmatrix}$$

where  $\vec{b}_{ik}$  is the  $k$ th column of  $B$ .



**Definition 2.5** Let  $A$  be an  $m \times n$  matrix and  $B$  be a  $p \times q$  matrix. The Kronecker or tensor product of  $A$  and  $B$  is the  $mp \times nq$  matrix

$$A \otimes B \equiv \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & & & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}$$

A useful property of concatenation is given in Theorem 2.7.

**Theorem 2.7** Let  $A$  be  $m \times m$ ,  $X$  be  $m \times n$ , and  $B$  be  $n \times n$ . Then

$$co(AXB) = (B^T \otimes A)co(X).$$

Proof of this theorem is given in [14] (with the notation  $vecA$  instead of  $co(A)$ ) and [17]. Concatenating each side of (2.25) yields the system

$$\left[ \mathcal{D}((\psi')^{r+1}) \otimes \Gamma_r + \left( \mathcal{D}((\psi')^{r+1}) \Gamma_r \right) \otimes I \right] co(U) = co(F), \quad (2.26)$$

which is then solved directly.

To achieve the collocation system for (2.19), proceed as above using the collocation method to write the system for each ordinary differential equation. Because the  $p$  and  $q$  terms are zero here, the system with  $l = 1/2$  is identical to the one given in (2.26) when  $r = 1/2$ .

## CHAPTER 3

DOMAIN DECOMPOSITION FOR ORDINARY DIFFERENTIAL  
EQUATIONSIntroduction

The two traditional methods of domain decomposition, overlapping and patching, have similar approaches. For the two-point boundary-value problem

$$\begin{aligned}\mathcal{L}u(x) &\equiv -u''(x) + p(x)u'(x) + q(x)u(x) \\ &= f(x), \quad a < x < b \\ u(a) &= u(b) = 0,\end{aligned}\tag{3.1}$$

the domain  $\Omega = (a, b)$  is split into two subintervals,  $\Omega^1$  and  $\Omega^2$ . If the subintervals overlap, the differential equation is solved on each subdomain and the solutions are matched at the new endpoints. Nice overviews are found in [4] and [10]. For the patching method,  $\Omega^1$  and  $\Omega^2$  have only a common endpoint. In this case, the problem is solved on the subdomains, and the solutions and their derivatives are matched at this point of intersection. This method is described in [4] and [15].

Patching domain decomposition methods are traditionally used for solving elliptic problems such as Poisson's equation. Certain characteristics of the sinc methods make an overlapping approach desirable. Because of this, both methods are followed through in this preliminary work on the two-point boundary-value problem. These methods have been shown to be related, and in some cases, identical. For more details see [5] and [6]. For a recent survey of the literature on domain decomposition techniques see [8].

The second section presents the overlapping method of domain decomposition. Both the Sinc-Galerkin and sinc-collocation overlapping methods are given. The third section presents the patching method of domain decomposition. Again, both the Sinc-Galerkin and sinc-collocation techniques are applied. Numerical examples are given in each section.

### The Overlapping Method of Domain Decomposition

This general method is outlined in [4]. Split the domain  $\Omega$  into two subdomains,  $\Omega^1 = (a, \xi_2)$  and  $\Omega^2 = (\xi_1, b)$ ,  $a < \xi_1 < \xi_2 < b$ . Then solve the problems

$$\begin{aligned} -u''(x) + p(x)u'(x) + q(x)u(x) &= f(x), \quad x \in \Omega^1 \\ u(a) &= 0, \end{aligned} \tag{3.2}$$

$$\begin{aligned} -v''(x) + p(x)v'(x) + q(x)v(x) &= f(x), \quad x \in \Omega^2 \\ v(b) &= 0, \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} u(\xi_1) &= v(\xi_1) \\ u(\xi_2) &= v(\xi_2). \end{aligned} \tag{3.4}$$

Whether using the Sinc-Galerkin or sinc-collocation overlapping methods, the following conventions are used. Let  $\phi^j$  denote the conformal map given in (2.2) on the subdomain  $\Omega^j$ , and let  $S_k^j$  denote the respective basis functions given in (2.6) on  $\Omega^j$ . Notice that the non-homogeneous boundary conditions (3.4) introduced by the decomposition are not satisfied by the  $S_k^j$ . Therefore we must introduce the extra basis functions

$$\omega_1(x) = (x - a)^3 \left( \frac{-3}{(\xi_2 - a)^4} x + \frac{4\xi_2 - a}{(\xi_2 - a)^4} \right) \tag{3.5}$$

and

$$\omega_2(x) = (x - b)^3 \left( \frac{-3}{(\xi_1 - b)^4} x + \frac{4\xi_1 - b}{(\xi_1 - b)^4} \right). \quad (3.6)$$

The basis function  $\omega_1(x)$  is chosen so that  $\omega_1(a) = \omega_1'(a) = \omega_1''(a) = 0$ ,  $\omega_1(\xi_2) = 1$ , and  $\omega_1'(\xi_2) = 0$ . Similarly,  $\omega_2(x)$  satisfies  $\omega_2(b) = \omega_2'(b) = \omega_2''(b) = 0$ ,  $\omega_2(\xi_1) = 1$ , and  $\omega_2'(\xi_1) = 0$ . See Figure 5 and Figure 6 for a view of these boundary basis functions when  $a = -1$ ,  $b = 4$ ,  $\xi_1 = .9$ , and  $\xi_2 = 1$ .

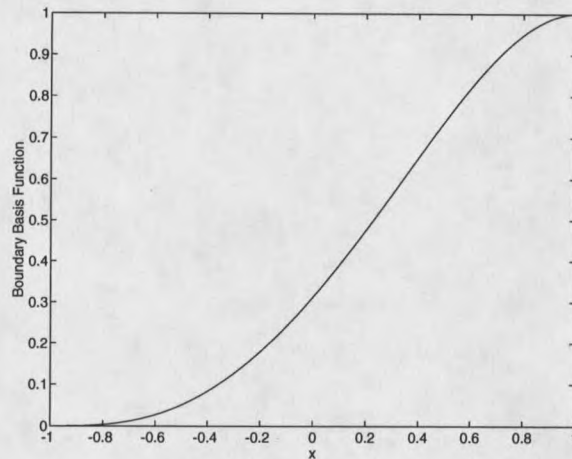


Figure 5: The boundary basis function  $\omega_1$  on the interval  $\Omega^1 = (-1, 1)$

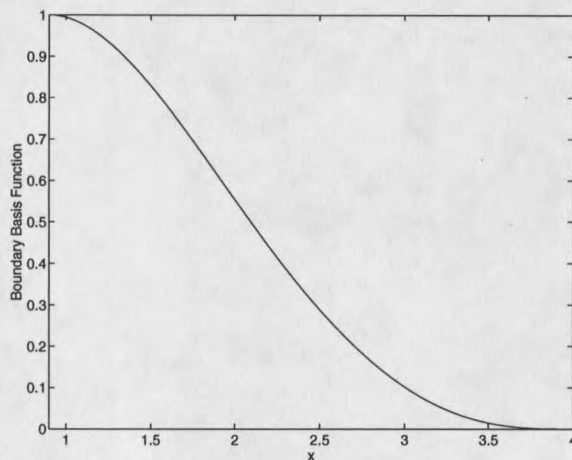


Figure 6: The boundary basis function  $\omega_2$  on the interval  $\Omega^2 = (.9, 4)$

In the region of overlap, which will be quite small, the approximate solution may be taken to be the solution in  $\Omega^1$  or  $\Omega^2$  since they are, within the order of the method, equal.

### The Sinc-Galerkin Overlapping Method

The approximate solutions used for (3.2) and (3.3) are

$$u_{m^1}(x) = \sum_{k=-M^1}^{N^1} u_k S_k^1(x) + u_{N^1+1} \omega_1(x), \quad m^1 = M^1 + N^1 + 1 \quad (3.7)$$

and

$$v_{m^2}(x) = v_{-M^2-1} \omega_2(x) + \sum_{j=-M^2}^{N^2} v_j S_j^2(x), \quad m^2 = M^2 + N^2 + 1, \quad (3.8)$$

respectively. Thus the superscript denotes either subdomain  $\Omega^1$  or  $\Omega^2$ . From the boundary condition at  $\xi_1$

$$\begin{aligned} \sum_{k=-M^1}^{N^1} u_k S_k^1(\xi_1) + u_{N^1+1} \omega_1(\xi_1) &= v_{-M^2-1} \omega_2(\xi_1) + \sum_{j=-M^2}^{N^2} v_j S_j^2(\xi_1) \\ &= v_{-M^2-1}. \end{aligned} \quad (3.9)$$

Similarly, the boundary condition at  $\xi_2$  yields

$$\begin{aligned} v_{-M^2-1} \omega_2(\xi_2) + \sum_{j=-M^2}^{N^2} v_j S_j^2(\xi_2) &= \sum_{k=-M^1}^{N^1} u_k S_k^1(\xi_2) + u_{N^1+1} \omega_1(\xi_2) \\ &= u_{N^1+1}. \end{aligned} \quad (3.10)$$

Let

$$\vec{b}_1 = [ S_{-M^1}^1(\xi_1) \quad \dots \quad S_{N^1}^1(\xi_1) \quad \omega_1(\xi_1) ]$$

and

$$\vec{b}_2 = [ \omega_2(\xi_2) \quad S_{-M^2}^2(\xi_2) \quad \dots \quad S_{N^2}^2(\xi_2) ]$$

and let  $\vec{e}_j$  denote the appropriately sized row vector with 1 in the  $j$ th position and 0 elsewhere. Orthogonalizing the residual with respect to the  $m^1$  or  $m^2$  sinc basis functions in  $\Omega^1$  or  $\Omega^2$ , respectively, yields the under-determined bordered systems

$$\mathcal{G}_r^1 \vec{u} = \vec{f}^1 \quad (3.11)$$

and

$$\mathcal{G}_r^1 \vec{v} = \vec{f}^2 \quad (3.12)$$

where

$$\mathcal{G}_r^1 = \left[ \begin{array}{c|c} \Gamma_r^1 & \frac{\mathcal{L}(\omega_1)}{((\phi^1)')^{1+r}} \end{array} \right]_{m^1 \times (m^1+1)} \quad (3.13)$$

and

$$\mathcal{G}_r^2 = \left[ \begin{array}{c|c} \frac{\mathcal{L}(\omega_2)}{((\phi^2)')^{1+r}} & \Gamma_r^2 \end{array} \right]_{m^2 \times (m^2+1)} \quad (3.14)$$

The notation  $\mathcal{L}(\omega_1)/((\phi^1)')^{1+r}$  represents the  $m^1 \times 1$  matrix whose entries are given by  $\mathcal{L}(\omega_1(x_k^1))/((\phi^1)'(x_k^1))^{1+r}$ , and similarly for  $\mathcal{L}(\omega_2)/((\phi^2)')^{1+r}$ . The nodes  $x_k^1$  and  $x_k^2$  are defined in (2.4) for the subdomains  $\Omega^1$  and  $\Omega^2$ , respectively, where  $h^j$  is given by the formula

$$h^j = \left( \frac{\pi d}{\alpha M^j} \right)^{1/2}$$

for  $j = 1, 2$ . The matrices  $\Gamma_r^1$  and  $\Gamma_r^2$  are defined in (2.12) where  $\phi$  is replaced by  $\phi^1$  and  $\phi^2$ , respectively. The resulting coupled system for (3.1) with block coefficient matrix is then

$$\begin{bmatrix} \mathcal{G}_r^1 & \Theta^2 \\ B_o^1 & B_o^2 \\ \Theta^1 & \mathcal{G}_r^2 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{f}^1 \\ \vec{\theta} \\ \vec{f}^2 \end{bmatrix} \quad (3.15)$$

where the  $m^1 \times 1$  vector  $\vec{f}^1$  and the  $m^2 \times 1$  vector  $\vec{f}^2$  are defined as

$$\vec{f}^j = \mathcal{D} \left( \frac{1}{((\phi^j)')^{r+1}} \right) \vec{f}, \quad j = 1, 2, \quad (3.16)$$

$$\vec{u} = [u_{-M^1} \dots u_{N^1+1}]^T, \quad (3.17)$$

$$\vec{v} = [v_{-M^2-1} \dots v_{N^2}]^T,$$

$$B_o^1 = \begin{bmatrix} \vec{b}_1 \\ -\vec{e}_{m^1+1} \end{bmatrix}, \quad (3.18)$$

$$B_o^2 = \begin{bmatrix} -\vec{e}_1 \\ \vec{b}_2 \end{bmatrix}, \quad (3.19)$$

and  $\Theta^j$  is an appropriately sized zero matrix (either  $m^1 \times (m^1 + 1)$  or  $m^2 \times (m^2 + 1)$ ), for  $j = 1$  or  $2$ , respectively. The matrices  $B_o^j$  are  $2 \times (m^j + 1)$  for  $j = 1, 2$  and the vector  $\vec{\theta}$  is a  $2 \times 1$  zero vector.

Each of the three sample problems presented in this chapter is posed on the interval  $(-1, 4)$ . For the overlapping domain decomposition method this interval is broken into the two subintervals  $\Omega^1 = (-1, 1)$  and  $\Omega^2 = (.9, 4)$ . The solution in the region of overlap is taken to be the approximate solution in  $\Omega^1$ . The problem was checked for sensitivity to the amount of overlap. Extremely small overlaps did decrease the overall accuracy. The degradation in error is obvious for overlaps smaller than .001. For each problem the coefficients chosen are  $p(x) \equiv 1$  and  $q(x) \equiv 1$ , and the weight function exponents used are  $r = l = 1/2$ . The choices of  $p(x)$  and  $q(x)$  are made with the expectation that this will be appropriate for elliptic partial differential equations. The choice of  $d$  was  $\pi/2$ , and for consistency  $\alpha$  was chosen to be 1 in each example. Even in this more general setting, and without balancing errors with a tuned choice of  $\alpha$ , the error predicted in Theorem 2.5 is nearly attained. More finely tuned choices of  $\alpha$  can further improve these results.

Let

$$\|E_S\| = \max_{x \in S} |u(x) - u_A(x)|$$

where  $S = \{x_k^1 : -M^1 \leq k \leq N^1\} \cup \{x_j^2 : -M^2 \leq j \leq N^2\}$  is the set of all grid points generated from the Sinc-Galerkin method and  $u_A$  is given by

$$u_A(x) = \begin{cases} u_{m^1}(x), & x \in \Omega^1 \\ v_{m^2}(x), & x \in \Omega^2 \setminus \Omega^1 \end{cases}$$

Here the true solution is given by  $u(x)$ . A uniform error is found by letting

$$\|E_U\| = \max_{y \in U} |u(y) - u_A(y)|$$

where  $U = \{y_j = -1 + 5j/100 : 0 \leq j \leq 100\}$  is a uniform grid of mesh size 0.05. Note also that in each example  $N^j = M^j$  for  $j = 1, 2$ . The problems are all run using MATLAB, Version 4.1, which provides sixteen digits of precision on a DECstation 5000/200.

**Example 3.1** Consider the test problem

$$-u''(x) + u'(x) + u(x) = \left(\frac{4}{25}\right)^2 (x^4 - 2x^3 - 29x^2 + 69x + 38)$$

$$u(-1) = u(4) = 0$$

which has the analytic true solution given by

$$u(x) = \left(\frac{4}{25}\right)^2 (x+1)^2(x-4)^2.$$

As is expected, the method performs very well on this problem. Define  $M \equiv M^1 = M^2$  and hence the mesh sizes satisfy  $h^1 = h^2 \equiv h$ . Thus  $h = \pi/\sqrt{2M}$ . Note that in Figure 7 the true and approximate solutions are plotted only on the interval  $[-.9, 2]$  for  $M = 2, 4$ , and 8. This was done to better illustrate the convergence near the region of overlap. The errors reported in Table 1 illustrate nearly exponential convergence.



$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$2.2015e - 02$	$3.4013e - 02$
4	$1.1107e + 00$	$9.4811e - 03$	$9.9684e - 03$
8	$7.8540e - 01$	$2.6798e - 03$	$2.5105e - 03$
16	$5.5536e - 01$	$2.5562e - 04$	$2.3860e - 04$
32	$3.9270e - 01$	$7.3522e - 06$	$6.8913e - 06$
64	$2.7768e - 01$	$4.4802e - 08$	$4.2140e - 08$

Table 1: Error in the approximation of  $u$  for Example 3.1 using the Sinc-Galerkin overlapping method

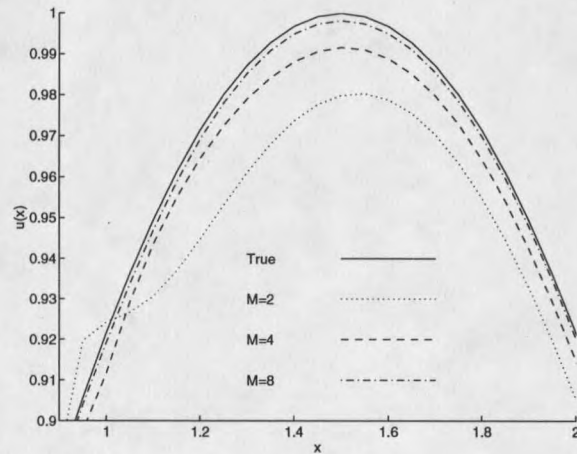


Figure 7: Sinc-Galerkin overlapping solution for Example 3.1

**Example 3.2** Next consider the test problem

$$-u''(x) + u'(x) + u(x) = \left(\frac{2}{5}\right) \frac{4x^4 - 20x^3 - 14x^2 + 98x + 113}{4((4-x)(1+x))^{3/2}}$$

$$u(-1) = u(4) = 0$$

which has the singular true solution given by

$$u(x) = \frac{2\sqrt{(x+1)(4-x)}}{5}$$

Again the method performs well on this problem, though the presence of boundary singularities ( $u'$  and  $u''$  are not defined at the endpoints) causes larger errors than in Example 3.1. The errors are reported in Table 2, with  $M \equiv M^1 = M^2$  and hence  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ .

The column labelled  $\|E_S\|_G$  is a prediction of the error calculated from the error at the previous step using the Sinc-Galerkin convergence rate,

$$\|u_m - u\|_\infty \leq CM^2 e^{-\sqrt{\pi d \alpha} M}$$

Hence,

$$\begin{aligned} \|u_{2m} - u\|_\infty &\leq C(2M)^2 e^{-\sqrt{\pi d \alpha} 2M} \\ &= C \left( M^2 e^{-\sqrt{\pi d \alpha} M} \right)^{\sqrt{2}} (2M)^2 M^{-2\sqrt{2}} \end{aligned}$$

Thus a prediction of the error at the next step can be made via the formula

$$\|E_S(2M)\|_G \approx \|E_S(M)\|^{\sqrt{2}} (2M)^2 M^{-2\sqrt{2}}. \quad (3.20)$$

Notice the nearly exponential convergence exhibited by a comparison of the adjacent columns  $\|E_S\|$  and  $\|E_S\|_G$  in Table 2.

$M$	$h$	$\ E_S\ $	$\ E_S\ _G$	$\ E_U\ $
2	1.5708e + 00	1.5175e - 01	—	1.5736e - 01
4	1.1107e + 00	1.0394e - 01	1.5653e - 01	1.0077e - 01
8	7.8540e - 01	4.8658e - 02	5.1620e - 02	4.4876e - 02
16	5.5536e - 01	1.4629e - 02	9.9374e - 03	1.3458e - 02
32	3.9270e - 01	2.4870e - 03	1.0227e - 03	2.2984e - 03
64	2.7768e - 01	1.9342e - 04	4.6998e - 05	1.7875e - 04

Table 2: Error in the approximation of  $u$  for Example 3.2 using the Sinc-Galerkin overlapping method

**Example 3.3** Consider the test problem

$$-u''(x) + u'(x) + u(x) = \frac{4x^4 - 14x^3 - 41x^2 + 56x + 104}{64(x+1)^{1.5}}$$

$$u(-1) = u(4) = 0$$

which has singular true solution given by

$$u(x) = \frac{\sqrt{x+1}(x-4)^2}{16}$$

This problem is the only one of the given examples which can truly benefit from domain decomposition. There is a boundary singularity at the left-hand endpoint. Thus it is advantageous to concentrate more nodes on the left-hand subdomain  $\Omega^1$ , and decrease the amount of work necessary to achieve a desired accuracy. The numerical error for domain decomposition with  $M^1 = M^2 \equiv M$  is given in Table 3, and the true and approximate solutions are shown in Figure 8 for  $M = 2, 4$ , and 8. In this case  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . In Table 4, the number of nodes in the left-hand subdomain  $\Omega^1$  is fixed at 129, corresponding to  $M^1 = 64$ , and the number

of nodes in the right-hand subdomain  $\Omega^2$  is allowed to increase as indicated. Thus  $h^1 = .27768$  and  $h^2 = \pi/\sqrt{2M^2}$ . The results show that one can use  $M^1 = 64$  on the left and  $M^2 = 32$  on the right to achieve the same error as that obtained with  $M^1 = M^2 = 64$ . In addition, with  $M^1 = 64$  and  $M^2 = 16$ , the error is of the same magnitude as for the cases  $M^2 = 32$  and  $M^2 = 64$ . Thus the size of the systems can be dramatically decreased, while the same error is achieved.

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$2.2852e - 01$	$2.3394e - 01$
4	$1.1107e + 00$	$1.4880e - 01$	$1.4258e - 01$
8	$7.8540e - 01$	$6.8459e - 02$	$6.6033e - 02$
16	$5.5536e - 01$	$2.0531e - 02$	$1.9821e - 02$
32	$3.9270e - 01$	$3.4897e - 03$	$3.3660e - 03$
64	$2.7768e - 01$	$2.7141e - 04$	$2.6173e - 04$

Table 3: Error in the approximation of  $u$  for Example 3.3 using the Sinc-Galerkin overlapping method

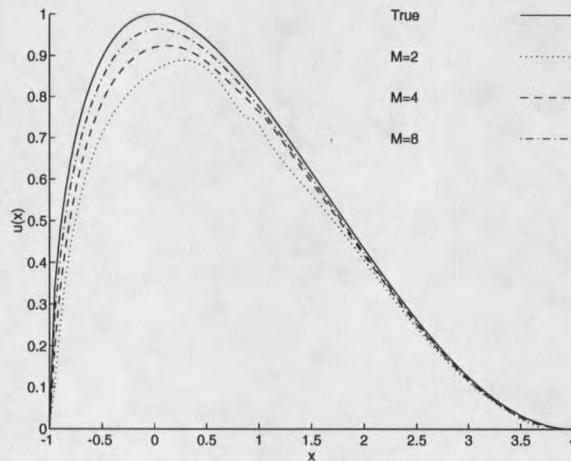


Figure 8: Sinc-Galerkin overlapping solution for Example 3.3

$M^2$	$h^2$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	9.6309e - 02	9.6309e - 02
4	1.1107e + 00	1.9016e - 02	1.9205e - 02
8	7.8540e - 01	4.9133e - 03	4.7255e - 03
16	5.5536e - 01	3.2919e - 04	3.0967e - 04
32	3.9270e - 01	2.7141e - 04	2.6168e - 04
64	2.7768e - 01	2.7141e - 04	2.6173e - 04

Table 4: Error in the Sinc-Galerkin overlapping approximation of  $u$  for Example 3.3 with  $M^1 = 64$  and  $h^1 = .27768$  on the left-hand subinterval

### The Sinc-Collocation Overlapping Method

The approximations used for (3.2) and (3.3) are

$$\begin{aligned}
 u_{m^1}(x) &= \sum_{k=-M^1}^{N^1} u_k ((\phi^1)'(x_k^1))^l \frac{S_k^1(x)}{((\phi^1)'(x))^l} + u_{N^1+1} \omega_1(x) \\
 &\approx \sum_{k=-M^1}^{N^1} u_k S_k^1(x) + u_{N^1+1} \omega_1(x)
 \end{aligned} \tag{3.21}$$

and

$$\begin{aligned}
 v_{m^2}(x) &= v_{-M^2-1} \omega_2(x) + \sum_{j=-M^2}^{N^2} v_j ((\phi^2)'(x_j^2))^l \frac{S_j^2(x)}{((\phi^2)'(x))^l} \\
 &\approx v_{-M^2-1} \omega_2(x) + \sum_{j=-M^2}^{N^2} v_j S_j^2(x),
 \end{aligned} \tag{3.22}$$

respectively, where  $\omega_1(x)$  and  $\omega_2(x)$  are given in equations (3.5) and (3.6). The approximate sums in (3.21) and (3.22) reflect the fact that these sums are, within the order of the method, equal. The latter sums are easier to calculate in practice, and perform equally well. Applying the boundary condition at  $\xi_1$  and  $\xi_2$  using the latter approximations given in (3.21) and (3.22) yields the same equations given in

(3.9) and (3.10), respectively. Let  $B_o^1$ ,  $B_o^2$ ,  $\vec{e}_j$ ,  $\vec{u}$ , and  $\vec{v}$  be as defined in the previous section. Collocating the system yields the under-determined, bordered systems

$$C_i^1 \vec{u} = \vec{f}^1$$

and

$$C_i^2 \vec{v} = \vec{f}^2$$

where

$$C_i^1 = \left[ \begin{array}{c|c} \mathcal{K}_i^1 & \frac{\mathcal{L}(\omega_1)}{((\phi^1)')^{2-l}} \end{array} \right]_{m^1 \times (m^1+1)} \quad (3.23)$$

and

$$C_i^2 = \left[ \begin{array}{c|c} \frac{\mathcal{L}(\omega_2)}{((\phi^2)')^{2-l}} & \mathcal{K}_i^2 \end{array} \right]_{m^2 \times (m^2+1)} \quad (3.24)$$

The matrices  $\mathcal{K}_i^1$  and  $\mathcal{K}_i^2$  are defined in (2.17) where  $\phi$  is replaced by  $\phi^1$  and  $\phi^2$ , respectively. The resulting coupled system for (3.1) with block coefficient matrix is then

$$\begin{bmatrix} C_i^1 & \Theta^2 \\ B_o^1 & B_o^2 \\ \Theta^1 & C_i^2 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{f}^1 \\ \vec{\theta} \\ \vec{f}^2 \end{bmatrix} \quad (3.25)$$

where for  $j = 1, 2$ ,  $\vec{f}^j$  is an  $m^j \times 1$  vector given by

$$\vec{f}^j = \mathcal{D} \left( \frac{1}{((\phi^j)')^{2-l}} \right) \vec{f}, \quad (3.26)$$

and  $\Theta^j$  is an appropriately sized zero matrix. Note that (3.25) is structurally the same as (3.15) where the Galerkin matrices  $\mathcal{G}_r^1$  and  $\mathcal{G}_r^2$  have been replaced by the collocation matrices  $C_i^1$  and  $C_i^2$ , respectively. The collocation matrices are found in (3.23) and (3.24).

As seen in the following three examples, the errors achieved via the sinc-collocation overlapping method are almost identical to those achieved with the Sinc-Galerkin overlapping method. The error predicted in Theorem 2.6 is nearly attained.

**Example 3.4** This problem, the same is in Example 3.1, has analytic solution

$$u(x) = \left(\frac{4}{25}\right)^2 (x+1)^2(x-4)^2.$$

As expected, the method works well on this problem. As seen in Figure 9 and Table 5 the sinc-collocation overlapping method performs well, with  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The errors exhibit almost exponential convergence and the uniform errors  $\|E_U\|$  are nearly identical to the errors on the sinc grid  $S$ ,  $\|E_S\|$ . Comparing the results in Table 1 to those in Table 5, and comparing Figure 7 to Figure 9, shows that the Sinc-Galerkin and sinc-collocation overlapping methods perform almost identically on this analytic problem.

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$4.0066e - 02$	$5.3626e - 02$
4	$1.1107e + 00$	$7.6096e - 03$	$8.1290e - 03$
8	$7.8540e - 01$	$2.7713e - 03$	$2.6082e - 03$
16	$5.5536e - 01$	$2.5819e - 04$	$2.4157e - 04$
32	$3.9270e - 01$	$7.3083e - 06$	$6.9339e - 06$
64	$2.7768e - 01$	$4.6774e - 08$	$4.4299e - 08$

Table 5: Error in the approximation of  $u$  for Example 3.4 using the sinc-collocation overlapping method

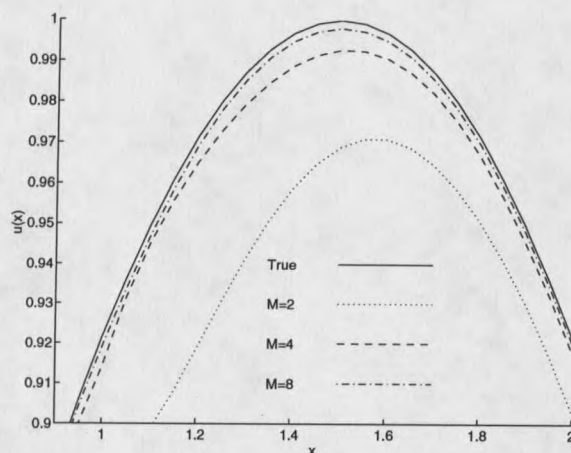


Figure 9: Sinc-collocation overlapping solution for Example 3.4

**Example 3.5** This problem, the same as in Example 3.2, has solution

$$u = \frac{2\sqrt{(x+1)(4-x)}}{5}.$$

The boundary singularities make this a much harder problem than that in Example 3.4. In spite of the singularities, the method performs well, as indicated in Table 6. Here again  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The column labelled  $\|E_S\|_C$  is a prediction of the error calculated from the error at the previous step using the sinc-collocation convergence rate. The formula used for this is derived like (3.20) and is given by

$$\|E_S(2M)\|_C \approx \|E_S(M)\|^{\sqrt{2}} (2M)^{3/2} M^{-3/\sqrt{2}}. \quad (3.27)$$

Notice the nearly exponential convergence exhibited by a comparison of the adjacent columns  $\|E_S\|$  and  $\|E_S\|_C$ .



$M$	$h$	$\ E_S\ $	$\ E_S\ _C$	$\ E_U\ $
2	1.5708e + 00	1.4893e - 01	—	1.5644e - 01
4	1.1107e + 00	1.0367e - 01	1.2443e - 01	1.0071e - 01
8	7.8540e - 01	4.8654e - 02	4.8461e - 02	4.4942e - 02
16	5.5536e - 01	1.4630e - 02	1.0808e - 02	1.3482e - 02
32	3.9270e - 01	2.4870e - 03	1.2842e - 03	2.2907e - 03
64	2.7768e - 01	1.9342e - 04	6.8123e - 05	1.7781e - 04

Table 6: Error in the approximation of  $u$  for Example 3.5 using the sinc-collocation overlapping method

**Example 3.6** This problem, the same as in Example 3.3, has solution

$$u(x) = \frac{\sqrt{x+1}(x-4)^2}{16}.$$

This example has a boundary singularity at one endpoint. Again the method performs well, as seen in Figure 10 and Table 7. Here  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . Comparing these to Figure 8 and Table 3 shows that the Sinc-Galerkin and sinc-collocation overlapping methods perform identically on this singular problem. Because the singularity is only at the left-hand endpoint, one can fix the number of nodes in the left-hand subdomain  $\Omega^1$  at 129, corresponding to  $M^1 = 64$  and let the number of nodes in  $\Omega^2$ , the right-hand subdomain, vary. As seen in Table 8, one can achieve the same amount of accuracy using fewer nodes in  $\Omega^2$ .

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$2.3058e - 01$	$2.3474e - 01$
4	$1.1107e + 00$	$1.4891e - 01$	$1.4242e - 01$
8	$7.8540e - 01$	$6.8460e - 02$	$6.6117e - 02$
16	$5.5536e - 01$	$2.0531e - 02$	$1.9848e - 02$
32	$3.9270e - 01$	$3.4897e - 03$	$3.3592e - 03$
64	$2.7768e - 01$	$2.7141e - 04$	$2.6059e - 04$

Table 7: Error in the approximation of  $u$  for Example 3.6 using the sinc-collocation overlapping method

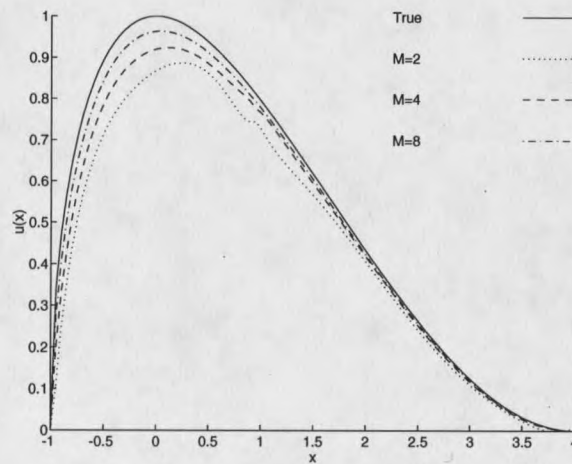


Figure 10: Sinc-collocation overlapping solution for Example 3.6

$M^2$	$h^2$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	7.4345e - 02	7.4345e - 02
4	1.1107e + 00	1.9799e - 02	1.9779e - 02
8	7.8540e - 01	4.8137e - 03	4.6152e - 03
16	5.5536e - 01	3.1914e - 04	3.0017e - 04
32	3.9270e - 01	2.7141e - 04	2.6054e - 04
64	2.7768e - 01	2.7141e - 04	2.6059e - 04

Table 8: Error in the sinc-collocation overlapping approximation of  $u$  for Example 3.6 with  $M^1 = 64$  and  $h^1 = .27768$  on the left-hand subinterval

### Patching Domain Decomposition Methods

The patching domain decomposition method is described in [4] and proceeds as follows. Split the domain into two non-overlapping subdomains,  $\Omega^1 = (a, \xi)$  and  $\Omega^2 = (\xi, b)$ ,  $a < \xi < b$ . Then solve the problems

$$-u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad x \in \Omega^1$$

$$u(a) = 0,$$

$$-v''(x) + p(x)v(x)' + q(x)v(x) = f(x), \quad x \in \Omega^2$$

$$v(b) = 0,$$

and

$$u(\xi) = v(\xi)$$

$$u'(\xi) = v'(\xi).$$

(3.28)

The Sinc-Galerkin and sinc-collocation methods both use the following conventions. As was done for the overlapping method, let  $\phi^j$  denote the conformal map given in (2.2) on the subdomain  $\Omega^j$ , and let  $S_k^j$  denote the respective basis functions given in (2.6) on  $\Omega^j$ . Recall that the subdomains  $\Omega^j$  have changed from that in the overlapping techniques, and thus so have the functions  $\phi^j$  and  $S_k^j$ . Notice that the non-homogeneous boundary conditions introduced by the decomposition are not satisfied by the  $S_k^j$ . Therefore we must again introduce extra basis functions similar to those in (3.5) and (3.6). These are given by

$$\omega_1(x) = (x - a)^3 \left( \frac{-3}{(\xi - a)^4} x + \frac{4\xi - a}{(\xi - a)^4} \right) \quad (3.29)$$

and

$$\omega_2(x) = (x - b)^3 \left( \frac{-3}{(\xi - b)^4} x + \frac{4\xi - b}{(\xi - b)^4} \right) \quad (3.30)$$

The choice of  $\omega_1(x)$  satisfies  $\omega_1(a) = \omega_1'(a) = \omega_1''(a) = 0$ ,  $\omega_1(\xi) = 1$ , and  $\omega_1'(\xi) = 0$ . Similarly  $\omega_2(b) = \omega_2'(b) = \omega_2''(b) = 0$ ,  $\omega_2(\xi) = 1$ , and  $\omega_2'(\xi) = 0$ .

### The Sinc-Galerkin Patching Method

For this method the same approximate solutions as in (3.7) and (3.8) are used. From the first boundary condition on  $u$  and  $v$  at  $\xi$ , comes the relation

$$\sum_{k=-M^1}^{N^1} u_k S_k^1(\xi) + u_{N^1+1} \omega_1(\xi) = v_{-M^2-1} \omega_2(\xi) + \sum_{j=-M^2}^{N^2} v_j S_j^2(\xi)$$

and so

$$u_{N^1+1} = v_{-M^2-1} \quad (3.31)$$

Unfortunately, the derivative boundary condition at  $\xi$  creates more difficulty. The nature of the sinc basis and its derivatives leaves either a system which gives the same condition as the first boundary condition, or a condition which is unsolvable with regards to the first boundary condition. In either case, another way to approximate

the derivative conditions must be found. The alternative chosen from among several tested was a three-point approximation to the derivatives, [3]. Let  $\delta > 0$  and then

$$u'(\xi) = \frac{1}{2\delta}[u(\xi - 2\delta) - 4u(\xi - \delta) + 3u(\xi)] + \mathcal{O}(\delta^3)$$

and

$$v'(\xi) = \frac{-1}{2\delta}[v(\xi + 2\delta) - 4v(\xi + \delta) + 3v(\xi)] + \mathcal{O}(\delta^3).$$

The boundary condition on  $u'$  and  $v'$  in (3.28) yields

$$\begin{aligned} & \sum_{j=-M^1}^{N^1} u_j (S_j^1(\xi - 2\delta) - 4S_j^1(\xi - \delta)) + u_{N^1+1} (\omega_1(\xi - 2\delta) - 4\omega_1(\xi - \delta) + 3) \\ & = -v_{-M^2-1} (\omega_2(\xi + 2\delta) - 4\omega_2(\xi + \delta) + 3) - \sum_{k=-M^2}^{N^2} v_k (S_k^2(\xi + 2\delta) - 4S_k^2(\xi + \delta)). \end{aligned} \quad (3.32)$$

Let

$$\begin{aligned} \vec{b}_3 = & [ S_{-M^1}^1(\xi - 2\delta) - 4S_{-M^1}^1(\xi - \delta) \quad \dots \quad S_{N^1}^1(\xi - 2\delta) - 4S_{N^1}^1(\xi - \delta) \\ & \omega_1(\xi - 2\delta) - 4\omega_1(\xi - \delta) + 3 ] \end{aligned} \quad (3.33)$$

and

$$\begin{aligned} \vec{b}_4 = & [ \omega_2(\xi + 2\delta) - 4\omega_2(\xi + \delta) + 3 \quad S_{-M^2}^2(\xi + 2\delta) - 4S_{-M^2}^2(\xi + \delta) \quad \dots \\ & S_{N^2}^2(\xi + 2\delta) - 4S_{N^2}^2(\xi + \delta) ] \end{aligned} \quad (3.34)$$

Orthogonalizing the residual with respect to the  $m^1$  or  $m^2$  sinc basis functions in  $\Omega^1$  or  $\Omega^2$ , respectively, yields the under-determined bordered systems given in (3.11) and (3.12). Again  $\Gamma_r^1$  and  $\Gamma_r^2$  are defined in (2.12) with  $\phi$  replaced by  $\phi^1$  and  $\phi^2$ , respectively, and  $\mathcal{G}_r^1$  and  $\mathcal{G}_r^2$  are given in (3.13) and (3.14), respectively. Notice that

the regions  $\Omega^1$  and  $\Omega^2$  differ here since we are patching. The resulting coupled system for (3.1) with block coefficient matrix is then

$$\begin{bmatrix} \mathcal{G}_r^1 & \Theta^2 \\ B_p^1 & B_p^2 \\ \Theta^1 & \mathcal{G}_r^2 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{f}^1 \\ \vec{\theta} \\ \vec{f}^2 \end{bmatrix} \quad (3.35)$$

where  $\vec{f}^j$  for  $j = 1, 2$ , is given in (3.16),

$$B_p^1 = \begin{bmatrix} \vec{b}_3 \\ -\vec{e}_{m^1+1} \end{bmatrix}, \quad (3.36)$$

and

$$B_p^2 = \begin{bmatrix} \vec{b}_4 \\ \vec{e}_1 \end{bmatrix}. \quad (3.37)$$

The vectors  $\vec{u}$  and  $\vec{v}$  are defined in (3.17),  $B_p^j$  is  $2 \times (m^j + 1)$ ,  $\Theta^j$  is an  $m^j \times (m^j + 1)$  zero matrix, and  $\vec{\theta}$  is a  $2 \times 1$  zero vector. The same problems as in the overlapping domain decomposition method are presented in this section. In each problem the interval  $(-1, 4)$  is broken into the two non-overlapping subintervals  $(-1, 1)$  and  $(1, 4)$ . The problem was checked for sensitivity to the size of  $\delta$ . The choice  $\delta = .01$  was used in the following examples, and seemed to produce the best results. Each problem was run with coefficients  $p(x) \equiv 1$  and  $q(x) \equiv 1$ , and the weight function exponents used were  $r = l = .5$ . The choices for  $d$  and  $\alpha$  were  $\pi/2$  and 1, respectively. As before, more carefully tuned choices of  $\alpha$  could improve the numerical results. As in the previous examples, the problems were run with  $M^1 = N^1$  and  $M^2 = N^2$ .

**Example 3.7** Consider the problem

$$-u''(x) + u'(x) + u(x) = \left(\frac{4}{25}\right)^2 (x^4 - 2x^3 - 29x^2 + 69x + 38)$$

$$u(-1) = u(4) = 0,$$

which has true analytic solution

$$u(x) = \left(\frac{4}{25}\right)^2 (x+1)^2(x-4)^2.$$

As predicted, the method works well on this problem. As seen in Figure 11 and Table 9 the patching method performs well, with  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The errors exhibit almost exponential convergence and the uniform errors  $\|E_U\|$  are nearly identical to the errors on the sinc grid  $S$ ,  $\|E_S\|$ .

It is interesting to compare results from the Sinc-Galerkin patching method in Figure 11 to the results from the Sinc-Galerkin overlapping method in Figure 7. While the patching method starts out much more poorly than the overlapping method, the errors are very close at  $M = 64$ , as seen in a comparison of Table 9 and Table 1. The same phenomena will hold in the next examples.

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	1.6299e + 00	1.6323e + 00
4	1.1107e + 00	3.8467e - 01	3.8427e - 01
8	7.8540e - 01	3.3391e - 02	3.2594e - 02
16	5.5536e - 01	1.9320e - 03	1.8742e - 03
32	3.9270e - 01	5.3821e - 05	5.2164e - 05
64	2.7768e - 01	4.6443e - 07	4.5055e - 07

Table 9: Error in the approximation of  $u$  for Example 3.7 using the Sinc-Galerkin patching method

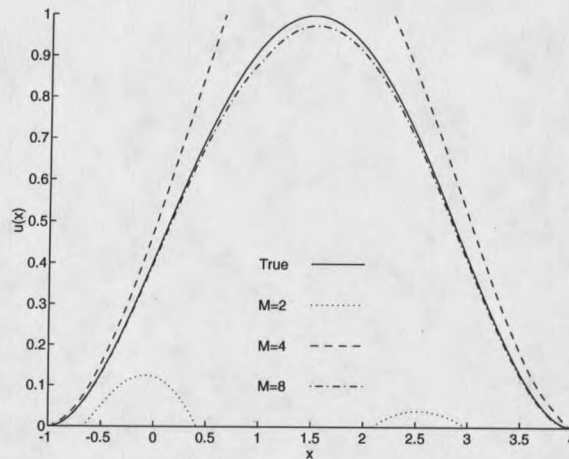


Figure 11: Sinc-Galerkin patching solution for Example 3.7

**Example 3.8** Next consider the test problem

$$-u''(x) + u'(x) + u(x) = \left(\frac{2}{5}\right) \frac{4x^4 - 20x^3 - 14x^2 + 98x + 113}{4((4-x)(1+x))^{3/2}}$$

$$u(-1) = u(4) = 0,$$

which has the singular true solution

$$u = \frac{2\sqrt{(x+1)(4-x)}}{5}.$$

The steepness of the solution due to the boundary singularity makes this a more interesting problem than Example 3.7. In spite of the singularities, the method performs well, as indicated in Table 10. Here again  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The formula for the predicted error column,  $\|E_S\|_G$  is given in (3.20). It is again reported for each appropriate case.



$M$	$h$	$\ E_S\ $	$\ E_S\ _G$	$\ E_U\ $
2	1.5708e + 00	2.0067e + 00	—	2.0061e + 00
4	1.1107e + 00	2.9090e - 01	—	2.9055e - 01
8	7.8540e - 01	4.7885e - 02	2.2127e - 01	4.4338e - 02
16	5.5536e - 01	1.4392e - 02	9.7148e - 03	1.3248e - 02
32	3.9270e - 01	2.4466e - 03	9.9936e - 04	2.2622e - 02
64	2.7768e - 01	1.9028e - 04	4.5922e - 05	1.7615e - 04

Table 10: Error in the approximation of  $u$  for Example 3.8 using the Sinc-Galerkin patching method

**Example 3.9** Consider the test problem

$$-u''(x) + u'(x) + u(x) = \frac{4x^4 - 14x^3 - 41x^2 + 56x + 104}{64(x+1)^{1.5}}$$

$$u(-1) = u(4) = 0,$$

which has the singular true solution

$$u(x) = \frac{\sqrt{x+1}(x-4)^2}{16}.$$

This problem also has a boundary singularity at one endpoint. Again the method performs well, as seen in Figure 12 and Table 11. Here  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . Because the singularity is only at the left-hand endpoint, one can fix the number of nodes in the left-hand subdomain at 129, corresponding to  $M^1 = 64$  and let the number of nodes in  $\Omega^2$ , the right-hand subdomain vary. As seen in Table 12, one can achieve the same amount of accuracy using fewer nodes in  $\Omega^2$ .

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$1.2811e + 00$	$1.2803e + 00$
4	$1.1107e + 00$	$1.4888e - 01$	$1.4278e - 01$
8	$7.8540e - 01$	$6.8455e - 02$	$6.5930e - 02$
16	$5.5536e - 01$	$2.0531e - 02$	$1.9811e - 02$
32	$3.9270e - 01$	$3.4897e - 03$	$3.3652e - 03$
64	$2.7768e - 01$	$2.7141e - 04$	$2.6159e - 04$

Table 11: Error in the approximation of  $u$  for Example 3.9 using the Sinc-Galerkin patching method

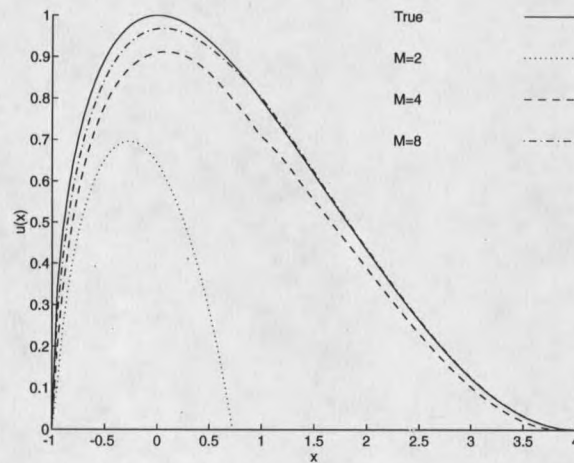


Figure 12: Sinc-Galerkin patching solution for Example 3.9

$M^2$	$h^2$	$\ E_S^G\ $	$\ E_U\ $
2	1.5708e + 00	1.5056e - 01	1.5106e - 01
4	1.1107e + 00	1.9831e - 01	1.9615e - 01
8	7.8540e - 01	6.0257e - 02	5.9407e - 02
16	5.5536e - 01	5.7655e - 03	5.6809e - 03
32	3.9270e - 01	2.7141e - 04	2.6086e - 04
64	2.7768e - 01	2.7141e - 04	2.6159e - 04

Table 12: Error in the approximation of  $u$  for Example 3.9 using the Sinc-Galerkin patching method with  $M^1 = 64$  and  $h^1 = .27768$

### The Sinc-Collocation Patching Method

Here the same approximations as in (3.21) and (3.22) are used, and  $\omega_1$  and  $\omega_2$  are given in (3.29) and (3.30), respectively. The discretization of the boundary conditions on  $u$  and  $v$  at  $\xi$  is given in (3.31). Similarly, the discretization of the boundary conditions on  $u'$  and  $v'$  at  $\xi$  is given in (3.32). The block coefficient matrix is structurally the same as in (3.35) where the Galerkin matrices  $\mathcal{G}_r^1$  and  $\mathcal{G}_r^2$  are replaced by the collocation matrices  $C_l^1$  and  $C_l^2$ , respectively, as seen below

$$\begin{bmatrix} C_l^1 & \Theta^2 \\ B_p^1 & B_p^2 \\ \Theta^1 & C_l^2 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{f}^1 \\ \vec{\theta} \\ \vec{f}^2 \end{bmatrix} \quad (3.38)$$

where  $\vec{f}^j$  for  $j = 1, 2$ , is given in (3.26),

$$B_p^1 = \begin{bmatrix} \vec{b}_3 \\ -\vec{e}_{m^1+1} \end{bmatrix},$$

and

$$B_p^2 = \begin{bmatrix} \vec{b}_4 \\ \vec{e}_1 \end{bmatrix},$$

and the vectors  $\vec{b}_3$  and  $\vec{b}_4$  are given in (3.33) and (3.34), respectively.

The same problems as in previous examples are presented here. In each problem the interval  $(-1,4)$  is broken into the two non-overlapping subintervals  $(-1,1)$  and  $(1,4)$ . The problem was checked for sensitivity to the size of  $\delta$ . The choice  $\delta = .01$  was used in the following examples, and seemed to produce the best results. Each problem was run with coefficients  $p(x) \equiv 1$  and  $q(x) \equiv 1$ , and the weight function exponents used were  $r = l = .5$ . The choices for  $d$  and  $\alpha$  were  $\pi/2$  and 1, respectively. As in the earlier examples, the problems were run with  $M^1 = N^1$  and  $M^2 = N^2$ .

**Example 3.10** This problem, as given in Example 3.7, has analytic solution

$$u(x) = \left(\frac{4}{25}\right)^2 (x+1)^2(x-4)^2.$$

As expected, the method works well on this problem. As seen in Figure 13 and Table 13 the patching method performs well, with  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The errors exhibit almost exponential convergence and the uniform errors are nearly identical to the errors on the sinc grid  $S$ .

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	1.7109e + 00	1.7111e + 00
4	1.1107e + 00	2.7094e - 01	2.7056e - 01
8	7.8540e - 01	3.1750e - 02	3.0943e - 02
16	5.5536e - 01	1.9591e - 03	1.9000e - 03
32	3.9270e - 01	5.4150e - 05	5.2495e - 05
64	2.7768e - 01	4.7482e - 07	4.6092e - 07

Table 13: Error in the approximation of  $u$  for Example 3.10 using the sinc-collocation patching method

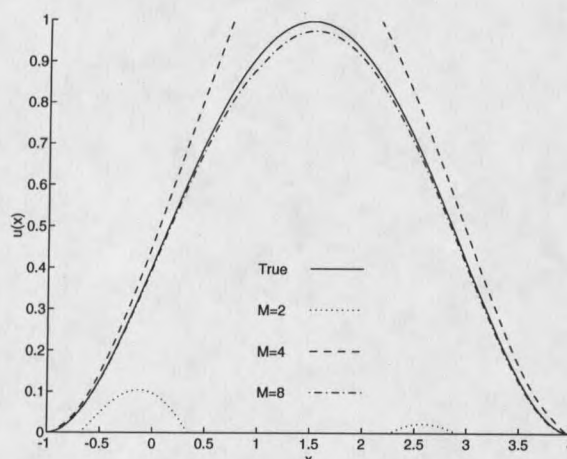


Figure 13: Sinc-collocation patching solution for Example 3.10

**Example 3.11** This problem, as given in Example 3.8, has solution

$$u = \frac{2\sqrt{(x+1)(4-x)}}{5}.$$

The boundary singularities make this a much harder problem than that in Example 3.10. In spite of the singularities, the method performs well, as indicated in Table 14. Here again  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . The formula for the predicted errors column,  $\|E_S\|_C$ , is given in (3.27). It is again reported for each appropriate case.

**Example 3.12** This problem, as given in Example 3.9, has solution

$$u(x) = \frac{\sqrt{x+1}(x-4)^2}{16}.$$

This problem also has a boundary singularity at one endpoint. Again the method performs well, as seen in Figure 14 and Table 15. Here  $M \equiv M^1 = M^2$  and  $h \equiv h^1 = h^2 = \pi/\sqrt{2M}$ . Because the singularity is only at the left-hand endpoint, one can fix the number of nodes in the left-hand subdomain at 129, corresponding to  $M^1 = 64$  and let the number of nodes in  $\Omega^2$ , the right-hand subdomain, vary. As seen in Table 16, one can achieve the same amount of accuracy using fewer nodes in  $\Omega^2$ .

$M$	$h$	$\ E_S\ $	$\ E_S\ _C$	$\ E_U\ $
2	$1.5708e + 00$	$2.0211e + 00$	—	$2.0208e + 00$
4	$1.1107e + 00$	$1.6052e - 01$	—	$1.6020e - 01$
8	$7.8540e - 01$	$4.7889e - 02$	$3.1796e - 02$	$4.4530e - 02$
16	$5.5536e - 01$	$1.4392e - 02$	$1.0568e - 02$	$1.3235e - 02$
32	$3.9270e - 01$	$2.4466e - 03$	$1.2548e - 03$	$2.2480e - 03$
64	$2.7768e - 01$	$1.9028e - 04$	$6.6563e - 05$	$1.7277e - 04$

Table 14: Error in the approximation of  $u$  for Example 3.11 using the sinc-collocation patching method

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$1.4456e + 00$	$1.4448e + 00$
4	$1.1107e + 00$	$1.4905e - 01$	$1.4278e - 01$
8	$7.8540e - 01$	$6.8457e - 02$	$6.6034e - 02$
16	$5.5536e - 01$	$2.0531e - 02$	$1.9830e - 02$
32	$3.9270e - 01$	$3.4897e - 03$	$3.3572e - 03$
64	$2.7768e - 01$	$2.7141e - 04$	$2.6020e - 04$

Table 15: Error in the approximation of  $u$  for Example 3.12 using the sinc-collocation patching method

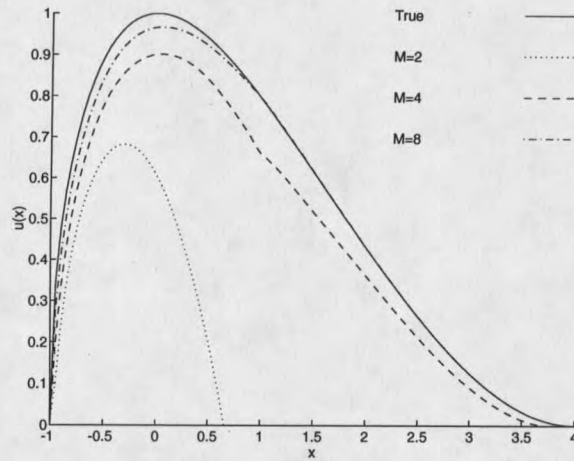


Figure 14: Sinc-collocation patching solution for Example 3.12

$M^2$	$h^2$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$1.1066e - 01$	$1.1115e - 01$
4	$1.1107e + 00$	$2.2433e - 01$	$2.2279e - 01$
8	$7.8540e - 01$	$6.1302e - 02$	$6.0458e - 02$
16	$5.5536e - 01$	$5.8142e - 03$	$5.7296e - 03$
32	$3.9270e - 01$	$2.7141e - 04$	$2.5946e - 04$
64	$2.7768e - 01$	$2.7141e - 04$	$2.6020e - 04$

Table 16: Error in the approximation of  $u$  for Example 3.12 using the sinc-collocation patching method with  $M^1 = 64$  and  $h^1 = .27768$

## CHAPTER 4

## DOMAIN DECOMPOSITION FOR POISSON'S EQUATION

Introduction

Poisson's equation on a rectangle with homogeneous Dirichlet boundary conditions is given by

$$\begin{aligned} -\nabla^2 u(x, y) &\equiv -\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega \\ u(x, y) &= 0, \quad (x, y) \in \partial\Omega. \end{aligned} \tag{4.1}$$

Consider splitting the domain

$$\Omega = \{(x, y) : a < x < b, s < y < t\}$$

into one or more subdomains. In this chapter,  $\Omega$  is split into two subdomains. Choose  $a < \xi_1 \leq \xi_2 < b$  and let

$$\Omega^1 = \{(x, y) : a < x < \xi_2, s < y < t\}$$

and

$$\Omega^2 = \{(x, y) : \xi_1 < x < b, s < y < t\}.$$

Let  $\Gamma^1 = \{(\xi_2, y) : s < y < t\}$  and  $\Gamma^2 = \{(\xi_1, y) : s < y < t\}$ .

The classical approach to domain decomposition for (4.1) requires the solution of the systems

$$\begin{aligned} -\Delta u(x, y) &= f(x, y), \quad (x, y) \in \Omega^1 \\ u(x, y) &= 0, \quad (x, y) \in \partial\Omega^1 \setminus \Gamma^1 \end{aligned}$$



and

$$\begin{aligned} -\Delta v(x, y) &= f(x, y), \quad (x, y) \in \Omega^2 \\ v(x, y) &= 0, \quad (x, y) \in \partial\Omega^2 \setminus \Gamma^2. \end{aligned}$$

For the overlapping method the matching conditions require

$$\begin{aligned} u(\xi_1, y) &= v(\xi_1, y), \quad (\xi_1, y) \in \Gamma^2 \\ u(\xi_2, y) &= v(\xi_2, y), \quad (\xi_2, y) \in \Gamma^1. \end{aligned}$$

For the patching method note that  $\xi_1 = \xi_2 \equiv \xi$  and thus  $\Gamma^1 = \Gamma^2 \equiv \Gamma$ . The matching conditions require that

$$\begin{aligned} u(\xi, y) &= v(\xi, y), \quad (\xi, y) \in \Gamma \\ \frac{\partial u}{\partial \nu}(\xi, y) &= -\frac{\partial v}{\partial \nu}(\xi, y), \quad (\xi, y) \in \Gamma, \end{aligned}$$

where  $\nu$  is the outward unit normal for the appropriate subdomain. These techniques for elliptic problems are described in [6]. In the second section overlapping domain decomposition is developed for Poisson's equation using the Sinc-Galerkin method. The sinc-collocation method is equivalent for this problem with  $r = l = 1/2$ . In the third section patching domain decomposition is developed. Numerical results are presented in each of these sections.

### Overlapping Domain Decomposition for Poisson's Equation

An easier approach for using the sinc methods is to proceed as in the development of the discrete sinc system for Poisson's equation in the last section of Chapter 2. The approximations given for Poisson's equation in (2.20) do not meet the required conditions. Thus the approximations

$$u_{m_x^1, m_y}(x, y) = \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y}^{N_y} u_{jk} \Psi_j^1(x) S_k(y) \quad (4.2)$$

and

$$u_{m_x^2, m_y}(x, y) = \sum_{j=-M_x^2-1}^{N_x^2} \sum_{k=-M_y}^{N_y} v_{jk} \Psi_j^2(x) S_k(y) \quad (4.3)$$

are used where

$$\Psi_j^1(x) = \begin{cases} S_j^1(x), & -M_x^1 \leq j \leq N_x^1 \\ \omega_1(x), & j = N_x^1 + 1 \end{cases}$$

and

$$\Psi_j^2(x) = \begin{cases} \omega_2(x), & j = -M_x^2 - 1 \\ S_j^2(x), & -M_x^2 \leq j \leq N_x^2 \end{cases}$$

Here,  $\omega_1(x)$  and  $\omega_2(x)$  are the boundary basis functions given in (3.5) and (3.6), respectively. The sinc basis functions  $S_j^1(x)$  and  $S_j^2(x)$  are as defined in (2.6) with conformal maps

$$\phi^1(x) = \ln \left( \frac{x-a}{\xi_2-x} \right)$$

and

$$\phi^2(x) = \ln \left( \frac{x-\xi_1}{b-x} \right).$$

The sinc basis functions  $S_k(y)$  are defined in (2.21) and the conformal map  $\psi(y)$  is given in (2.22).

Fix  $x = x_p^1 \in \Omega^1$ ,  $-M_x^1 \leq p \leq N_x^1$ . Along this line the approximation given in (4.2) has the form

$$\begin{aligned} u_{m_x^1, m_y}(x_p^1, y) &= \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y}^{N_y} u_{jk} \Psi_j^1(x_p^1) S_k(y) \\ &= \sum_{k=-M_y}^{N_y} \left( u_{pk} + u_{N_x^1+1, k} \omega_1(x_p^1) \right) S_k(y) \\ &\equiv \sum_{k=-M_y}^{N_y} c_{pk} S_k(y). \end{aligned}$$

Along this line, the problem being solved is

$$-u_{xx}(x_p^1, y) - u_{yy}(x_p^1, y) = f(x_p^1, y), \quad s < y < t$$

$$u(x_p^1, s) = u(x_p^1, t) = 0$$

or

$$\begin{aligned} -u_{yy}(x_p^1, y) &= f(x_p^1, y) + u_{xx}(x_p^1, y), \quad s < y < t \\ u(x_p^1, s) &= u(x_p^1, t) = 0. \end{aligned}$$

This is a second-order two-point boundary-value problem with homogeneous Dirichlet boundary conditions. Using either the Sinc-Galerkin method or the sinc-collocation method (they are the same for this problem when  $r = l = 1/2$ ) the system to be solved is given for  $-M_x^1 \leq p \leq N_x^1$  by

$$\Gamma_{1/2}^1 \vec{c}_{p,\vec{y}} = \mathcal{D} \left( (\psi')^{-3/2} \right) \left( \vec{f}(x_p^1, \vec{y}) + \vec{u}_{xx}(x_p^1, \vec{y}) \right). \quad (4.4)$$

Here  $\vec{c}_{p,\vec{y}}$  is a vector with entries  $c_{p,k}$ ,  $\vec{f}(x_p^1, \vec{y})$  is a vector with entries  $f(x_p^1, y_k)$ , and  $\vec{u}_{xx}(x_p^1, \vec{y})$  is a vector with entries  $u_{xx}(x_p^1, y_k)$  for  $-M_y \leq k \leq N_y$ . From (2.12) the coefficient matrix for  $r = 1/2$  has the form

$$\Gamma_{1/2}^j = \frac{-1}{h^2} I^{(2)} + \mathcal{D} \left( \frac{1}{4} \right)$$

for  $j = 1, 2$ .

Allowing  $p$  to vary yields the equation

$$\Gamma_{1/2}^1 C^T = \mathcal{D} \left( (\psi')^{-3/2} \right) \left( F^1 + U_{xx} \right)^T. \quad (4.5)$$

Transposing each side of (4.5) yields

$$C \Gamma_{1/2}^1 = \left( F^1 + U_{xx} \right) \mathcal{D} \left( (\psi')^{-3/2} \right) \quad (4.6)$$

or

$$C \Gamma_{1/2}^1 \mathcal{D} \left( (\psi')^{3/2} \right) = F^1 + U_{xx}. \quad (4.7)$$

The matrices  $C = [c_{jk}]$ ,  $F^1 = [f(x_j^1, y_k)]$ , and  $U_{xx} = [u_{xx}(x_j^1, y_k)]$  are of size  $m_x^1 \times m_y$ .

In order to write this in terms of a system that can be solved for  $U$ , notice that

$$c_{pk} = u_{pk} + u_{N_x^1+1,k} \omega_1(x_p^1).$$

Let  $E_1$  be the  $m_x^1 \times (m_x^1 + 1)$  partitioned matrix

$$E_1 \equiv \left[ I \quad \vdots \quad \vec{\omega}_1 \right], \quad (4.8)$$

where  $\vec{\omega}_1$  is the  $m_x^1 \times 1$  vector with entries  $\omega_1(x_j^1)$ . Then

$$C = E_1 U.$$

Substituting this into the system (4.7) yields

$$E_1 U \left( \Gamma_{1/2}^1 \right)^T \mathcal{D} \left( (\psi')^{3/2} \right) = F^1 + U_{xx}. \quad (4.9)$$

Similarly, let  $x = x_p^2 \in \Omega^2$ ,  $-M_x^2 \leq p \leq N_x^2$ . Along this line, the differential equation takes the form

$$\begin{aligned} -v_{yy}(x_p^2, y) &= f(x_p^2, y) + v_{xx}(x_p^2, y), \quad s < y < t \\ v(x_p^2, s) &= v(x_p^2, t) = 0. \end{aligned}$$

Notice that  $v_{m_x^2, m_y}$  has the form

$$\begin{aligned} v_{m_x^2, m_y}(x_p^2, y) &= \sum_{j=-M_x^2-1}^{N_x^2} \sum_{k=-M_y}^{N_y} v_{jk} \Psi_j^2(x_p^2) S_k(y) \\ &= \sum_{k=-M_y}^{N_y} (v_{pk} + v_{-M_x^2-1, k} \omega_2(x_p^2)) S_k(y). \end{aligned}$$

Let  $E_2$  be the  $m_x^2 \times (m_x^2 + 1)$  partitioned matrix

$$E_2 \equiv \left[ \vec{\omega}_2 \quad \vdots \quad I \right], \quad (4.10)$$

where  $\vec{\omega}_2$  is the  $m_x^2 \times 1$  vector whose entries are given by  $\omega_2(x_j^2)$ . Proceeding as above yields the system

$$E_2 V \left( \Gamma_{1/2}^2 \right)^T \mathcal{D} \left( (\psi')^{3/2} \right) = F^2 + V_{xx}, \quad (4.11)$$

where  $V = [v_{jk}]$ ,  $F^2 = [f(x_j^2, y_k)]$ , and  $V_{xx} = [v_{xx}(x_j^2, y_k)]$  are  $m_x^2 \times m_y$  matrices.

Now fix  $y = y_q$ ,  $-M_y \leq q \leq N_y$ . The problem to be solved is now the one-dimensional domain decomposition problem

$$-u_{xx}(x, y_q) = f(x, y_q) + u_{yy}(x, y_q), \quad x \in \Omega^1$$

$$u(a, y_q) = 0$$

$$-v_{xx}(x, y_q) = f(x, y_q) + v_{yy}(x, y_q), \quad x \in \Omega^2$$

$$v(b, y_q) = 0$$

with matching conditions given by

$$u(\xi_1, y_q) = v(\xi_1, y_q)$$

$$u(\xi_2, y_q) = v(\xi_2, y_q).$$

There are  $m_y$  of these problems so letting  $q$  vary yields the three coupled equations

$$\mathcal{G}_{1/2}^1 U = \mathcal{D} \left( ((\phi^1)')^{-3/2} \right) (F^1 + U_{yy}), \quad (4.12)$$

$$B_o^1 U + B_o^2 V = \Theta, \quad (4.13)$$

and

$$\mathcal{G}_{1/2}^2 V = \mathcal{D} \left( ((\phi^2)')^{-3/2} \right) (F^2 + V_{yy}). \quad (4.14)$$

Here  $\mathcal{G}_{1/2}^1$  is given by

$$\mathcal{G}_{1/2}^1 = \left[ \begin{array}{c|c} \Gamma_{1/2}^1 & \frac{-\vec{\omega}_1''}{((\phi^1)')^{3/2}} \end{array} \right]_{m^1 \times (m^1+1)}, \quad (4.15)$$

$\mathcal{G}_{1/2}^2$  is given by

$$\mathcal{G}_{1/2}^2 = \left[ \begin{array}{c|c} \frac{-\vec{\omega}_2''}{((\phi^2)')^{3/2}} & \Gamma_{1/2}^2 \end{array} \right]_{m^2 \times (m^2+1)}, \quad (4.16)$$

$B_o^1$  and  $B_o^2$  are the boundary matrices found in (3.18) and (3.19), respectively, and  $\Theta$  is a  $2 \times m_y$  zero matrix. From (4.12) comes the equation

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1 U = F^1 + U_{yy}. \quad (4.17)$$

Adding (4.9) and (4.17) yields the system

$$\begin{aligned} E_1 U (\Gamma_{1/2}^1)^T \mathcal{D}((\psi')^{3/2}) + \mathcal{D}(((\phi^1)')^{3/2}) \mathcal{G}_{1/2}^1 U &= F^1 + U_{xx} + U_{yy} + F^1 \\ &= F^1. \end{aligned} \quad (4.18)$$

For simplification of the notation, rewrite (4.18) as

$$A_1 U C_1 + A_2 U C_2 = F^1, \quad (4.19)$$

where

$$A_1 = E_1, \quad (4.20)$$

$$A_2 = \mathcal{D}(((\phi^1)')^{3/2}) \mathcal{G}_{1/2}^1, \quad (4.21)$$

$$C_1 = (\Gamma_{1/2}^1)^T \mathcal{D}((\psi')^{3/2}), \quad (4.22)$$

and

$$C_2 = I \quad (4.23)$$

is an  $m_y \times m_y$  identity matrix. Concatenating each side of (4.19) and using Theorem 2.7 yields the under-determined system

$$[C_1^T \otimes A_1 + C_2^T \otimes A_2] \text{co}(U) = \text{co}(F^1)$$

or

$$\mathcal{P} \text{co}(U) = \text{co}(F^1), \quad (4.24)$$

where

$$\mathcal{P} = C_1^T \otimes A_1 + C_2^T \otimes A_2.$$

Similarly from (4.14) comes the system

$$\mathcal{D}(((\phi^2)')^{3/2}) \mathcal{G}_{1/2}^2 V = F^2 + V_{yy}. \quad (4.25)$$

Adding (4.25) and (4.11) yields the system

$$\begin{aligned} E_2 V (\Gamma_{1/2}^2)^T \mathcal{D} ((\psi')^{3/2}) + \mathcal{D} (((\phi^2)')^{3/2}) \mathcal{G}_{1/2}^2 V &= F^2 + V_{xx} + V_{yy} + F^2 \\ &= F^2 . \end{aligned} \quad (4.26)$$

Again for simplicity rewrite (4.26) as

$$A_3 V C_3 + A_4 V C_4 = F^2 , \quad (4.27)$$

where

$$A_3 = E_2 , \quad (4.28)$$

$$A_4 = \mathcal{D} (((\phi^2)')^{3/2}) \mathcal{G}_{1/2}^2 , \quad (4.29)$$

$$C_3 = (\Gamma_{1/2}^2)^T \mathcal{D} ((\psi')^{3/2}) , \quad (4.30)$$

and

$$C_4 = I \quad (4.31)$$

is an  $m_y \times m_y$  identity matrix. Concatenating each side and again using Theorem 2.7 yields the under-determined system

$$[C_3^T \otimes A_3 + C_4^T \otimes A_4] \text{co}(V) = \text{co}(F^2) \quad (4.32)$$

or

$$Q \text{co}(V) = \text{co}(F^2) \quad (4.33)$$

with

$$Q = C_3^T \otimes A_3 + C_4^T \otimes A_4 .$$

Finally, the boundary terms represented in (4.13) can be written

$$B_o^1 UI + B_o^2 VI = \Theta . \quad (4.34)$$

Again concatenating each side of (4.34) and applying Theorem 2.7 yields the system

$$(I \otimes B_o^1) co(U) + (I \otimes B_o^2) co(V) = co(\Theta) \quad (4.35)$$

or

$$\mathcal{R}co(U) + \mathcal{S}co(V) = \vec{\theta}, \quad (4.36)$$

where

$$\mathcal{R} = I \otimes B_o^1, \quad (4.37)$$

$$\mathcal{S} = I \otimes B_o^2, \quad (4.38)$$

and  $\vec{\theta}$  is a  $2m_y \times 1$  zero vector. Combining the systems (4.24), (4.33), and (4.36) yields the following block matrix system

$$\begin{bmatrix} \mathcal{P} & \Theta^2 \\ \mathcal{R} & \mathcal{S} \\ \Theta^1 & \mathcal{Q} \end{bmatrix} \begin{bmatrix} co(U) \\ co(V) \end{bmatrix} = \begin{bmatrix} co(F^1) \\ \vec{\theta} \\ co(F^2) \end{bmatrix},$$

where the zero matrix  $\Theta^1$  is of size  $m_y m_x^2 \times m_y(m_x^1 + 1)$  and the zero matrix  $\Theta^2$  is of size  $m_y m_x^1 \times m_y(m_x^2 + 1)$ .

In each of the following three sample problems

$$\Omega = \{(x, y) : -1 < x < 4, 0 < y < 1\}.$$

Thus in this section,  $\Omega^1$  and  $\Omega^2$  are chosen so that

$$\Omega^1 = \{(x, y) : -1 < x < 1, 0 < y < 1\}$$

and

$$\Omega^2 = \{(x, y) : 0.9 < x < 4, 0 < y < 1\}.$$

As before,  $d$  is chosen to be  $\pi/2$  and  $\alpha$  is chosen to be 1. Let the sinc error be defined by

$$\|E_S\| = \max_{(x,y) \in S} |u(x, y) - u_A(x, y)| \quad (4.39)$$



where

$$S = \{x_k^1 : -M_x^1 \leq k \leq N_x^1\} \cup \{x_j^2 : -M_x^2 \leq j \leq N_x^2\} \times \{y_j : -M_y \leq j \leq N_y\}$$

is the set of all grid points generated from the Sinc-Galerkin method and  $u_A$  is given by

$$u_A(x, y) = \begin{cases} u_{m_x^1, m_y}(x, y), & (x, y) \in \Omega^1 \\ v_{m_x^2, m_y}(x, y), & (x, y) \in \Omega^2 \setminus \Omega^1 \end{cases} \quad (4.40)$$

Without loss of generality, the approximate solution in the region of overlap can be chosen to be  $u_{m_x^1, m_y}(x, y)$ . The errors obtained from  $u_{m_x^1, m_y}(x, y)$  and  $v_{m_x^2, m_y}(x, y)$  were both checked, and the higher of the two numbers is reported in the tables. Similarly define a uniform error by

$$\|E_U\| = \max_{(x, y) \in U} |u(x, y) - u_A(x, y)| \quad (4.41)$$

where

$$U = \{(-1 + 5j/100, k/100) : 0 \leq j \leq 100, 0 \leq k \leq 100\}$$

is a uniform grid over  $\Omega$ .

**Example 4.1** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x+1)^2(x-4)^2y^2(1-y)^2}{3.1596}$$

This problem is an analogue of that given in Example 3.1. The choices  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$  are made. The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . A mesh plot of the approximate solution with

$M = 8$  is shown in Figure 15, and a contour plot which displays the convergence of the approximations for increasing  $M$  is shown in Figure 16. The contour levels are decreasing from the center out to the boundary. As seen in Table 17, the method performs well on this problem and the results are consistent with those of Example 3.1.

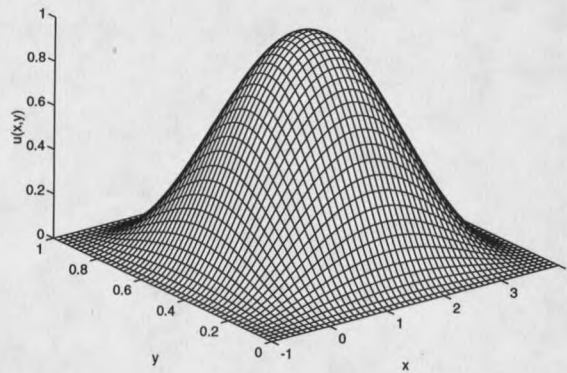


Figure 15: Sinc-Galerkin overlapping solution for Example 4.1 with  $M = 8$

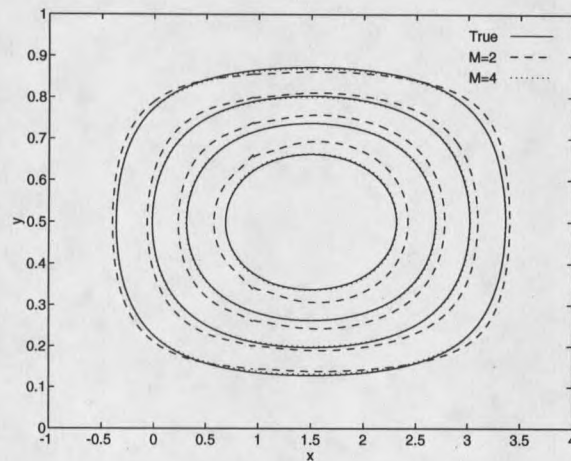


Figure 16: Contour plots from the Sinc-Galerkin overlapping solution for Example 4.1 with contour lines at  $u = .8, .6, .4,$  and  $.2$

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$6.1074e - 02$	$8.2025e - 02$
4	$1.1107e + 00$	$3.6344e - 03$	$1.0926e - 02$
6	$9.0690e - 01$	$3.5574e - 03$	$3.5572e - 03$
8	$7.8540e - 01$	$1.2540e - 03$	$1.1371e - 03$
10	$7.0248e - 01$	$5.8874e - 04$	$5.0834e - 04$

Table 17: Error in the Sinc-Galerkin overlapping method for Example 4.1

**Example 4.2** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{\sqrt{(x+1)(x-4)y(1-y)}}{3.1877}$$

This example is analogous to Example 3.2. Here  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$ . The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . This problem has boundary singularities, and thus is a harder problem than the one given in Example 4.1. A mesh plot of the approximate solution for  $M = 8$  is shown in Figure 17 and contour plots from the approximate and true solutions are shown in Figure 18. Notice that the contour levels in this picture are different than the ones used in Figure 16. The highest contours are in the center of the figure. The method performs well, although the boundary singularities cause larger errors, as seen in Table 18. Again the results are consistent with those in Example 3.2.

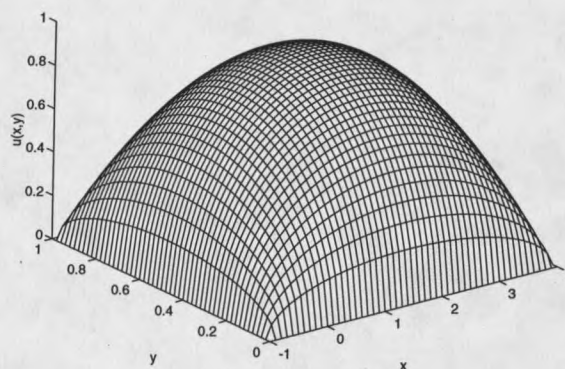


Figure 17: Sinc-Galerkin overlapping solution for Example 4.2 with  $M = 8$

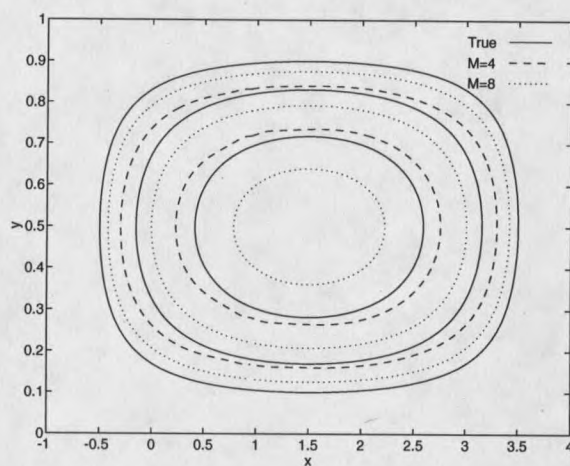


Figure 18: Contour plots from the Sinc-Galerkin overlapping solution for Example 4.2 with contour lines at  $u = .9, .75,$  and  $.6$

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$2.2048e - 01$	$2.3170e - 01$
4	$1.1107e + 00$	$1.3576e - 01$	$1.3705e - 01$
6	$9.0690e - 01$	$8.9681e - 02$	$8.9632e - 02$
8	$7.8540e - 01$	$6.2029e - 02$	$6.1673e - 02$
10	$7.0248e - 01$	$4.4326e - 02$	$4.4104e - 02$

Table 18: Error in the Sinc-Galerkin overlapping method for Example 4.2

**Example 4.3** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{\sqrt{(x+1)y(x-4)^2(1-y)^2}}{5.4371}.$$

This is an analogue of Example 3.3. In this example  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$ . The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . This problem has boundary singularities, and thus is a harder problem than the one given in Example 4.1. The method performs well, although the boundary singularities cause larger errors, as seen in Table 19. Figures 19 and 20 show the steepness of the solution. Notice that the singularities are located along the lines  $y = 0$  and  $x = -1$ . The solution is very steep near this singularity, causing slightly larger errors than in Example 4.2. This could be corrected by tuning the choices of  $N_x^1$ ,  $N_y^1$ ,  $h_x^1$  and  $h_y$ . It is also advantageous to fix the number of nodes in the  $x$  direction as  $M_x^2 = N_x^2 = 2$  in  $\Omega^2$  and let  $M_y = N_y = M_x^1 = N_x^1$  vary. Thus  $h_x^2 = 1.5708$  and  $h_x^1 = h_y$  are varying. As seen in Table 20, nearly identical errors are obtained for  $M_x^1 = 10$ , with a much smaller system size. Additionally, the reduction in system size allows larger cases such as  $M_x^1 = 12$  and  $M_x^1 = 14$  to be run.

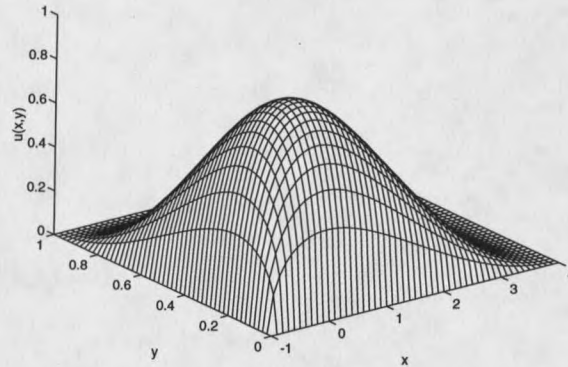


Figure 19: Sinc-Galerkin overlapping solution for Example 4.3 with  $M = 8$

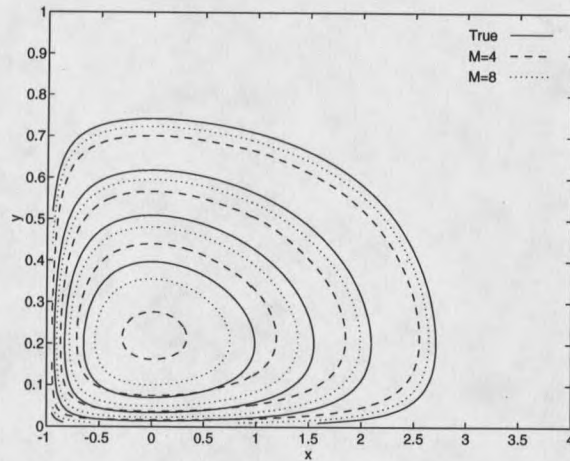


Figure 20: Contour plots from the Sinc-Galerkin overlapping solution for Example 4.3 with contour lines at  $u = .8, .6, .4,$  and  $.2$

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$3.5832e - 01$	$3.7729e - 01$
4	$1.1107e + 00$	$2.3420e - 01$	$2.3657e - 01$
6	$9.0690e - 01$	$1.5612e - 01$	$1.5294e - 01$
8	$7.8540e - 01$	$1.0813e - 01$	$1.0619e - 01$
10	$7.0248e - 01$	$7.7340e - 02$	$7.6184e - 02$

Table 19: Error in the Sinc-Galerkin overlapping method for Example 4.3

$M_x^1 = M_y$	$h_x^1 = h_y$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	3.5832e - 01	3.7729e - 01
4	1.1107e + 00	2.3418e - 01	2.3656e - 01
6	9.0690e - 01	1.5611e - 01	1.5290e - 01
8	7.8540e - 01	1.0812e - 01	1.0617e - 01
10	7.0248e - 01	7.7339e - 02	7.6156e - 02
12	6.4127e - 01	5.6768e - 02	5.5569e - 02
14	5.9371e - 01	4.2554e - 02	4.1750e - 02

Table 20: Error in the Sinc-Galerkin overlapping method for Example 4.3 with  $M_x^2 = 2$ .

### Patching Domain Decomposition for Poisson's Equation

For the patching method the approximations

$$u_{m_x^1, m_y} = \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y}^{N_y} u_{jk} \Psi_j^1(x) S_k(y) \quad (4.42)$$

and

$$v_{m_x^2, m_y} = \sum_{j=-M_x^2-1}^{N_x^2} \sum_{k=-M_y}^{N_y} v_{jk} \Psi_j^2(x) S_k(y) \quad (4.43)$$

are used where

$$\Psi_j^1(x) = \begin{cases} S_j^1(x), & -M_x^1 \leq j \leq N_x^1 \\ \omega_1(x), & j = N_x^1 + 1 \end{cases}$$

and

$$\Psi_j^2(x) = \begin{cases} \omega_2(x), & j = -M_x^2 - 1 \\ S_j^2(x), & -M_x^2 \leq j \leq N_x^2 \end{cases}$$

Here the boundary basis functions  $\omega_1(x)$  and  $\omega_2(x)$  are given in (3.29) and (3.30), respectively.  $S_j^1(x)$  and  $S_j^2(x)$  are given in (2.6) with conformal maps  $\phi^1(x)$  and  $\phi^2(x)$ ,

respectively, and  $S_k(y)$  is defined in (2.21). Fix  $x = x_p^1 \in \Omega^1$  for  $-M_x^1 \leq p \leq N_x^1$ .

Along this line the approximation given in (4.42) has the form

$$\begin{aligned} u_{m_x^1, m_y}(x_p^1, y) &= \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y}^{N_y} u_{jk} \Psi_j^1(x_p^1) S_k(y) \\ &= \sum_{k=-M_y}^{N_y} \left( u_{pk} + u_{N_x^1+1, k} \omega_1(x_p^1) \right) S_k(y) \\ &\equiv \sum_{k=-M_y}^{N_y} c_{pk} S_k(y). \end{aligned}$$

The problem being solved is

$$\begin{aligned} -u_{xx}(x_p^1, y) - u_{yy}(x_p^1, y) &= f(x_p^1, y), \quad s < y < t \\ u(x_p^1, s) = u(x_p^1, t) &= 0 \end{aligned}$$

or

$$\begin{aligned} -u_{yy}(x_p^1, y) &= f(x_p^1, y) + u_{xx}(x_p^1, y), \quad s < y < t \\ u(x_p^1, s) = u(x_p^1, t) &= 0. \end{aligned}$$

This is a second-order two-point boundary-value problem with homogeneous Dirichlet boundary conditions. Using either the Sinc-Galerkin method or the sinc-collocation method (they are the same for this problem when  $r = l = 1/2$ .) the system to be solved is given for  $-M_x^1 \leq p \leq N_x^1$  by

$$\Gamma_{1/2}^1 \vec{C}_{p, \vec{y}} = \mathcal{D} \left( (\psi')^{-3/2} \right) \left( \vec{f}(x_p^1, \vec{y}) + \vec{u}_{xx}(x_p^1, \vec{y}) \right).$$

Here the same notation as in (4.4) is used. Allowing  $p$  to vary yields the equation

$$\Gamma_{1/2}^1 C^T = \mathcal{D} \left( (\psi')^{-3/2} \right) \left( F^1 + U_{xx} \right)^T. \quad (4.44)$$

Again,  $C$ ,  $F^1$ , and  $U_{xx}$  are as used in (4.5). Transposing each side of (4.44) yields

$$C \Gamma_{1/2}^1 = \left( F^1 + U_{xx} \right) \mathcal{D} \left( (\psi')^{-3/2} \right)$$



or

$$C\Gamma_{1/2}^1 \mathcal{D} \left( (\psi')^{3/2} \right) = F^1 + U_{xx} . \quad (4.45)$$

In order to write this in terms of a system that can be solved for  $U$ , notice that

$$c_{pk} = u_{pk} + u_{N_x^1+1,k} \omega_1(x_p^1) .$$

Let  $E_1$  be the  $m_x^1 \times (m_x^1 + 1)$  partitioned matrix of (4.8),

$$E_1 \equiv \left[ I \quad \vdots \quad \vec{\omega}_1 \right] ,$$

where  $\vec{\omega}_1$  is the  $m_x^1 \times 1$  vector with entries  $\omega_1(x_j^2)$ . Then

$$C = E_1 U .$$

Substituting this into the system (4.45) yields

$$E_1 U \left( \Gamma_{1/2}^1 \right)^T \mathcal{D} \left( (\psi')^{3/2} \right) = F^1 + U_{xx} . \quad (4.46)$$

Similarly, let  $x = x_p^2 \in \Omega^2$ , where  $-M_x^2 \leq p \leq N_x^2$ . Let  $E_2$  be the  $m_x^2 \times (m_x^2 + 1)$  partitioned matrix of (4.10),

$$E_2 \equiv \left[ \vec{\omega}_2 \quad \vdots \quad I \right] ,$$

where  $\vec{\omega}_2$  is the  $m_x^2 \times 1$  vector with entries  $\omega_2(x_j^2)$ . Proceeding as above yields the system

$$E_2 V \left( \Gamma_{1/2}^2 \right)^T \mathcal{D} \left( (\psi')^{3/2} \right) = F^2 + V_{xx} . \quad (4.47)$$

Here  $V$ ,  $F^2$ , and  $V_{xx}$  are the same matrices used in (4.11). Now fix  $y = y_q$ ,  $-M_y \leq q \leq N_y$ . The problem to be solved is now the one-dimensional domain decomposition problem

$$-u_{xx}(x, y_q) = f(x, y_q) + u_{yy}(x, y_q) , \quad x \in \Omega^1$$

$$u(a, y_q) = 0$$

$$-v_{xx}(x, y_q) = f(x, y_q) + v_{yy}(x, y_q), \quad x \in \Omega^2$$

$$v(b, y_q) = 0$$

with matching conditions given by

$$u(\xi, y_q) = v(\xi, y_q)$$

$$u_x(\xi, y_q) = v_x(\xi, y_q)$$

for  $q = -M_y, \dots, N_y$ . There are  $m_y$  of these problems so letting  $q$  vary yields the three coupled equations

$$\mathcal{G}_{1/2}^1 U = \mathcal{D} \left( ((\phi^1)')^{-3/2} \right) (F^1 + U_{yy}), \quad (4.48)$$

$$B_p^1 U + B_p^2 V = \Theta, \quad (4.49)$$

and

$$\mathcal{G}_{1/2}^2 V = \mathcal{D} \left( ((\phi^2)')^{-3/2} \right) (F^2 + V_{yy}). \quad (4.50)$$

Here  $\mathcal{G}_{1/2}^1$  and  $\mathcal{G}_{1/2}^2$  are given in (4.15) and (4.16), respectively.  $B_p^1$  and  $B_p^2$  are the boundary matrices given in (3.36) and (3.37), respectively. The zero matrix  $\Theta$  is of size  $2 \times m_y$ .

From (4.48) comes the equation

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1 U = F^1 + U_{yy}. \quad (4.51)$$

Adding (4.46) and (4.51) yields the system

$$\begin{aligned} E_1 U (\Gamma_{1/2}^1)^T \mathcal{D} \left( (\psi')^{3/2} \right) + \mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1 U &= F^1 + U_{xx} + U_{yy} + F^1 \\ &= F^1. \end{aligned} \quad (4.52)$$

For simplification of the notation, rewrite (4.52) as

$$A_1 U C_1 + A_2 U C_2 = F^1, \quad (4.53)$$

where  $A_1$ ,  $A_2$ ,  $C_1$ , and  $C_2$  are defined in (4.20), (4.21), (4.22), and (4.23), respectively. Concatenating each side of (4.53) and applying Theorem 2.7 yields the under-determined system

$$\left[ C_1^T \otimes A_1 + C_2^T \otimes A_2 \right] co(U) = co(F^1)$$

or

$$\mathcal{P}co(U) = co(F^1). \quad (4.54)$$

Again

$$\mathcal{P} = C_1^T \otimes A_1 + C_2^T \otimes A_2.$$

Similarly from (4.50) comes the system

$$\mathcal{D} \left( ((\phi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2 V = F^2 + V_{yy}. \quad (4.55)$$

Adding (4.55) and (4.47) yields the system

$$\begin{aligned} E_2 V (\Gamma_{1/2}^2)^T \mathcal{D} \left( (\psi')^{3/2} \right) + \mathcal{D} \left( ((\phi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2 V &= F^2 + V_{xx} + V_{yy} + F^2 \\ &= F^2. \end{aligned} \quad (4.56)$$

Again for simplicity rewrite (4.56) as

$$A_3 V C_3 + A_4 V C_4 = F^2, \quad (4.57)$$

where  $A_3$ ,  $A_4$ ,  $C_3$ , and  $C_4$  are given in (4.28), (4.29), (4.30), and (4.31), respectively. Concatenating each side of (4.57) and applying Theorem 2.7 yields the under-determined system

$$\left[ C_3^T \otimes A_3 + C_4^T \otimes A_4 \right] co(V) = co(F^2)$$

or

$$\mathcal{Q}co(V) = co(F^2), \quad (4.58)$$

with

$$Q = C_3^T \otimes A_3 + C_4^T \otimes A_4 .$$

Finally, the boundary terms represented in (4.49) can be written

$$B_p^1 UI + B_p^2 VI = \Theta . \quad (4.59)$$

Concatenating each side of (4.59) and applying Theorem 2.7 yields the system

$$(I \otimes B_p^1) co(U) + (I \otimes B_p^2) co(V) = co(\Theta)$$

or

$$\mathcal{R}co(U) + \mathcal{S}co(V) = \vec{\theta} , \quad (4.60)$$

where  $\mathcal{R}$  and  $\mathcal{S}$  are given in (4.37) and (4.38), respectively, with  $B_o^j$  replaced by  $B_p^j$ ,  $j = 1, 2$ , and  $\vec{\theta}$  is a  $2m_y \times 1$  zero vector. Combining the systems (4.54), (4.58), and (4.60) yields the following block matrix system

$$\begin{bmatrix} \mathcal{P} & \Theta^2 \\ \mathcal{R} & \mathcal{S} \\ \Theta^1 & Q \end{bmatrix} \begin{bmatrix} co(U) \\ co(V) \end{bmatrix} = \begin{bmatrix} co(F^1) \\ \vec{\theta} \\ co(F^2) \end{bmatrix} .$$

Here  $\Theta^1$  is a  $m_y m_x^2 \times m_y(m_x^1 + 1)$  zero matrix and  $\Theta^2$  is a  $m_y m_x^1 \times m_y(m_x^2 + 1)$  zero matrix.

In each of the following three sample problems

$$\Omega = \{(x, y) : -1 < x < 4 , 0 < y < 1\} .$$

In this section,  $\Omega^1$  and  $\Omega^2$  are chosen so that

$$\Omega^1 = \{(x, y) : -1 < x < 1 , 0 < y < 1\}$$

and

$$\Omega^2 = \{(x, y) : 1 < x < 4 , 0 < y < 1\} .$$

Again  $d$  is chosen to be  $\pi/2$ , and  $\alpha$  is chosen to be 1. The sinc error  $\|E_S\|$  and the uniform error  $\|E_U\|$  are defined in (4.39) and (4.41), respectively, where the approximate solution is given by

$$u_A(x, y) = \begin{cases} u_{m_x^1, m_y}(x, y), & (x, y) \in \Omega^1 \\ v_{m_x^2, m_y}(x, y), & (x, y) \in \Omega^2 \end{cases}$$

The solutions  $u_{m_x^1, m_y}$  and  $v_{m_x^2, m_y}$  are defined in (4.42) and (4.43), respectively.

These examples are the same ones reported in the previous section on the overlapping method. This provides an opportunity to compare the results of both methods. Mesh plots of the approximate solution are not shown, for they are nearly identical to those in the previous section.

**Example 4.4** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x+1)^2(x-4)^2y^2(1-y)^2}{3.1596}$$

This same problem was used in Example 4.1 with the overlapping method. In this example  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$ . The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . Contour plots of the approximate solutions are shown in Figure 21, where the contour levels decrease away from the center of the domain. As seen in Table 21, the method performs well on this problem.

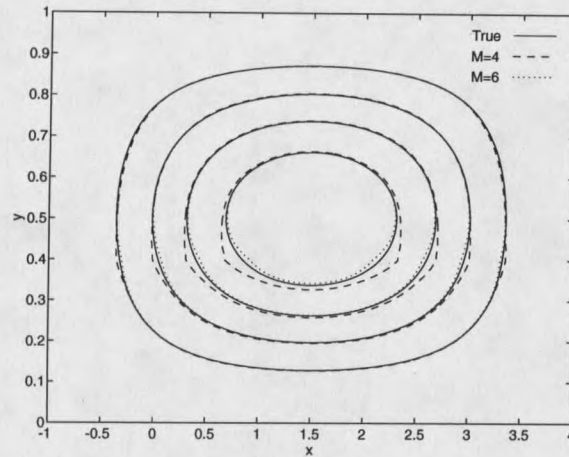


Figure 21: Contour plots from the Sinc-Galerkin patching solution for Example 4.4 with contour lines at  $u = 8, .6, .4,$  and  $.2$

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$9.5648e - 01$	$1.1284e + 00$
4	$1.1107e + 00$	$7.8851e - 02$	$7.0616e - 02$
6	$9.0690e - 01$	$2.8389e - 02$	$2.4483e - 02$
8	$7.8540e - 01$	$1.1334e - 02$	$9.7983e - 03$
10	$7.0248e - 01$	$5.1896e - 03$	$4.4390e - 03$

Table 21: Error in the Sinc-Galerkin patching method for Example 4.4

**Example 4.5** From Example 4.2 comes the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{\sqrt{(x+1)(x-4)y(1-y)}}{3.1877}.$$

In this example  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$ . The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . This problem has boundary singularities, and thus is a harder problem than the one given in Example 4.4. The method performs well, although the boundary singularities cause larger errors, as seen in Table 22 and as illustrated in the contour plots of Figure 22.

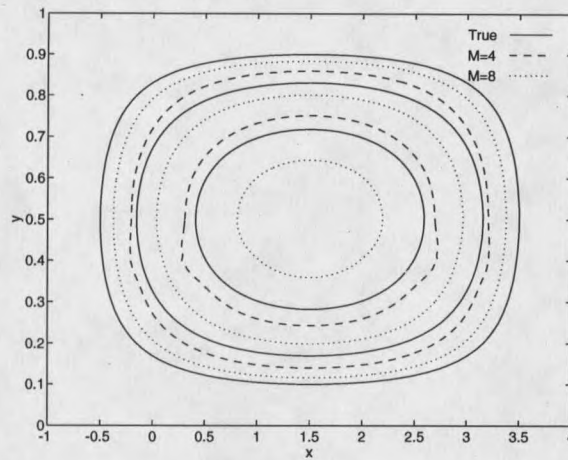


Figure 22: Contour plots from the Sinc-Galerkin patching solution for Example 4.5 with contour lines at  $u = .9, .75,$  and  $.6$

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$1.2717e + 00$	$1.4591e + 00$
4	$1.1107e + 00$	$1.3536e - 01$	$1.3698e - 01$
6	$9.0690e - 01$	$8.9824e - 02$	$8.9559e - 02$
8	$7.8540e - 01$	$6.2544e - 02$	$6.2401e - 02$
10	$7.0248e - 01$	$4.4742e - 02$	$4.4618e - 02$

Table 22: Error in the Sinc-Galerkin patching method for Example 4.5

**Example 4.6** From Example 4.3 comes the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (-1, 4) \times (0, 1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{\sqrt{(x+1)y(x-4)^2(1-y)^2}}{5.4371}.$$

In this example  $M \equiv M_x^1 = M_x^2 = M_y$  and  $N \equiv N_x^1 = N_x^2 = N_y$ . The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y = \pi/\sqrt{2M}$ . This problem has boundary singularities, and thus is a harder problem than the one given in Example 4.4. The method performs well, although the boundary singularities cause larger errors, as seen in Table 23. Figure 23 shows the steepness of the solution via contour plots. As in Example 4.3, the singularities suggest that it might be advantageous to fix  $M_x^2$  and allow  $M_x^1 = M_y$  to vary. As seen in Table 24, nearly identical results for  $M_x^1 = 10$  are obtained with  $M_x^2 = 6$ . Thus  $h_x^2 = .90690$  and  $h_x^1 = h_y$  are varying. The reduction in system size allows larger cases of  $M_x^1$  to be run.

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	6.4327e - 01	7.5272e - 01
4	1.1107e + 00	2.3423e - 01	2.3659e - 01
6	9.0690e - 01	1.5610e - 01	1.5282e - 01
8	7.8540e - 01	1.0813e - 01	1.0619e - 01
10	7.0248e - 01	7.7340e - 02	7.6181e - 02

Table 23: Error in the Sinc-Galerkin patching method for Example 4.6



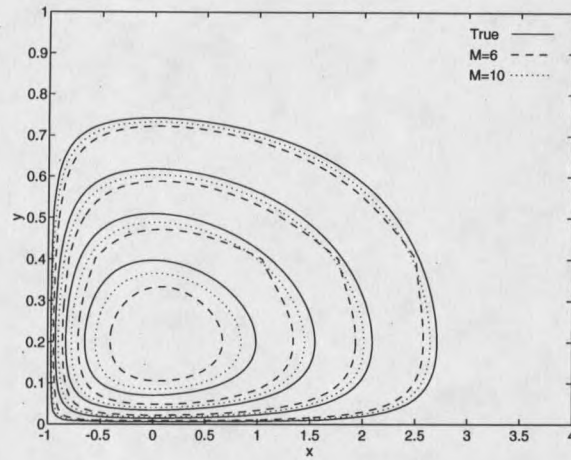


Figure 23: Contour plots from the Sinc-Galerkin patching solution for Example 4.6 with contour lines at  $u = .8, .6, .4,$  and  $.2$

$M_x^1 = M_y$	$h_x^1 = h_y$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$3.7726e - 01$	$3.4774e - 01$
4	$1.1107e + 00$	$2.3426e - 01$	$2.3662e - 01$
6	$9.0690e - 01$	$1.5610e - 01$	$1.5282e - 01$
8	$7.8540e - 01$	$1.0812e - 01$	$1.0610e - 01$
10	$7.0248e - 01$	$7.7336e - 02$	$7.6092e - 02$
12	$6.4127e - 01$	$5.6767e - 02$	$5.5499e - 02$
14	$5.9371e - 01$	$5.7712e - 02$	$4.9378e - 02$

Table 24: Error in the Sinc-Galerkin patching method for Example 4.6 with  $M_x^2 = 6$ .

## CHAPTER 5

## POISSON'S EQUATION ON AN EL-SHAPED DOMAIN

Introduction

The goal in developing the sinc methods for domain decomposition was to increase the complexity of domains on which problems can be solved via sinc methods. The sinc methods can easily handle partial differential equations posed on a rectangle, a semi-infinite strip, and an infinite strip. In fact, a second-order partial differential equation posed on a domain that can be mapped, either conformally or via a twice continuously differentiable change of variables with a nonzero Jacobian, to a rectangle can be solved using sinc methods. This chapter is dedicated to solving Poisson's equation on an el-shaped domain. Because the Sinc-Galerkin and sinc-collocation systems for Poisson's equation are identical for this problem (with  $r = l = 1/2$ ), as seen in (2.13) and (2.18), only the Sinc-Galerkin method will be discussed here.

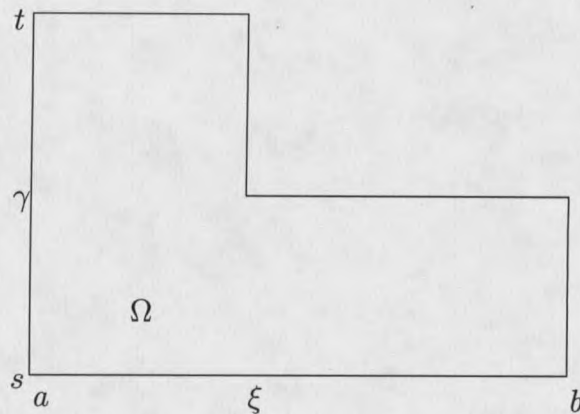
Poisson's equation on a general domain  $\Omega$  with homogeneous Dirichlet boundary conditions is given by

$$\begin{aligned} -\Delta u(x, y) &= f(x, y), \quad (x, y) \in \Omega \\ u(x, y) &= 0, \quad (x, y) \in \partial\Omega. \end{aligned} \tag{5.1}$$

In this chapter, for  $a < \xi < b$  and  $s < \gamma < t$ , the domain  $\Omega$  is given by

$$\Omega = \{(x, y) : a < x < \xi, s < y < t\} \cup \{(x, y) : a < x < b, s < y < \gamma\}.$$

See Figure 24 for a picture of  $\Omega$ .

Figure 24: The domain  $\Omega$ 

In order to use the methods of Chapter 4 to build the system, some concessions must be made. The development of both the patching and overlapping methods requires the cancellation of terms from neighboring subdomains. For this reason, the neighboring subdomains should share the same nodes along the line of overlap. For instance, in the examples in Chapter 4, the same  $y$  nodes were used in each subdomain in order to ensure this cancellation. These considerations imply that for either method, at least three subdomains should be used in the development of the method for the el-shaped domain. Of course, many more subdomains could be used here, but the method is most easily described by limiting to three subdomains.

The overlapping method is more complex in this case than the patching method. The requirement of sharing nodes mandates the placement of the regions of overlap. Unfortunately, this placement creates a new (but much smaller) el-shaped domain. Thus, no progress is made in this decomposition. Subdomain iteration could eliminate this restriction but this technique is not considered here. For this reason, the patching method is the one used.

### Sinc-Galerkin Patching Method on an El-Shaped Domain

Consider splitting the domain  $\Omega$  into three non-overlapping subdomains as follows. Let

$$\Omega^1 = \{(x, y) : a < x < \xi, \gamma < y < t\},$$

$$\Omega^2 = \{(x, y) : a < x < \xi, s < y < \gamma\},$$

and

$$\Omega^3 = \{(x, y) : \xi < x < b, s < y < \gamma\}.$$

Let  $\Gamma^1 = \{(x, \gamma) : a < x < \xi\}$  and  $\Gamma^2 = \{(\xi, y) : s < y < \gamma\}$ . This decomposition is seen in Figure 25.

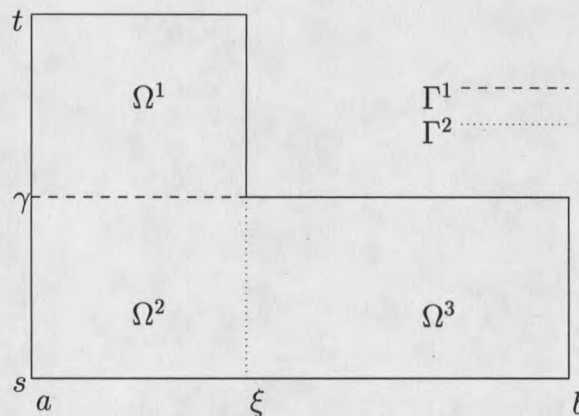


Figure 25: The subdomains for  $\Omega$

From (5.1) comes the new problems

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega^1$$

$$u(a, y) = u(\xi, y) = 0, \quad \gamma \leq y \leq t$$

$$u(x, t) = 0, \quad a \leq x \leq \xi,$$

$$-\Delta v(x, y) = f(x, y), \quad (x, y) \in \Omega^2$$

$$v(a, y) = 0, \quad s \leq y \leq \gamma$$

$$v(x, s) = 0, \quad a \leq x \leq \xi,$$

and

$$-\Delta w(x, y) = f(x, y), \quad (x, y) \in \Omega^3$$

$$w(x, s) = w(x, \gamma) = 0, \quad \xi \leq x \leq b$$

$$w(b, y) = 0, \quad s \leq y \leq \gamma.$$

The boundary conditions enforced are then

$$u(x, y) = v(x, y), \quad (x, y) \in \Gamma^1$$

$$v(x, y) = w(x, y), \quad (x, y) \in \Gamma^2$$

and

$$\frac{\partial u}{\partial \nu}(x, y) = -\frac{\partial v}{\partial \nu}(x, y), \quad (x, y) \in \Gamma^1$$

$$\frac{\partial v}{\partial \nu}(x, y) = -\frac{\partial w}{\partial \nu}(x, y), \quad (x, y) \in \Gamma^2,$$

where  $\nu$  is the appropriate outward unit normal. Additionally, there is a compatibility condition

$$u(\xi, \gamma) = v(\xi, \gamma) = w(\xi, \gamma).$$

We make the approximations

$$\begin{aligned} u_{m_x^1, m_y^1}(x, y) &= \sum_{j=-M_x^1}^{N_x^1} \sum_{k=-M_y^1-1}^{N_y^1} u_{jk} \Psi_j^1(x) \Upsilon_k^1(y) \\ u_{m_x^1, m_y^2}(x, y) &= \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y^2}^{N_y^2+1} v_{jk} \Psi_j^1(x) \Upsilon_k^2(y) \\ w_{m_x^2, m_y^2}(x, y) &= \sum_{j=-M_x^2-1}^{N_x^2} \sum_{k=-M_y^2}^{N_y^2} w_{jk} \Psi_j^2(x) \Upsilon_k^2(y). \end{aligned} \tag{5.2}$$

Here, the basis functions are given by

$$\Psi_j^1(x) = \begin{cases} S_j^1(x), & -M_x^1 \leq j \leq N_x^1 \\ \omega_1(x), & j = N_x^1 + 1 \end{cases},$$

$$\Psi_j^2(x) = \begin{cases} \omega_2(x), & j = -M_x^2 - 1 \\ S_j^2(x), & -M_x^2 \leq j \leq N_x^2 \end{cases},$$

$$\Upsilon_k^1(y) = \begin{cases} \eta_1(y), & k = -M_y^1 - 1 \\ S_k^1(y), & -M_y^1 \leq k \leq N_y^1 \end{cases},$$

and

$$\Upsilon_k^2(x) = \begin{cases} S_k^2(y), & -M_y^2 \leq k \leq N_y^2 \\ \eta_2(y), & k = N_y^2 + 1 \end{cases}.$$

The boundary basis functions are given by

$$\omega_1(x) = (x - a)^3 \left( \frac{-3}{(\xi - a)^4} x + \frac{4\xi - a}{(\xi - a)^4} \right),$$

$$\omega_2(x) = (x - b)^3 \left( \frac{-3}{(\xi - b)^4} x + \frac{4\xi - b}{(\xi - b)^4} \right),$$

$$\eta_1(y) = (y - t)^3 \left( \frac{-3}{(\gamma - t)^4} y + \frac{4\gamma - t}{(\gamma - t)^4} \right),$$

and

$$\eta_2(y) = (y - s)^3 \left( \frac{-3}{(\gamma - s)^4} y + \frac{4\gamma - s}{(\gamma - s)^4} \right).$$

The conformal maps used are

$$\phi^1(x) = \ln \left( \frac{x - a}{\xi - x} \right),$$

$$\phi^2(x) = \ln \left( \frac{x - \xi}{b - x} \right),$$

$$\psi^1(y) = \ln \left( \frac{y - \gamma}{t - y} \right)$$

and

$$\psi^2(y) = \ln \left( \frac{y-s}{\gamma-y} \right).$$

The following notation will be necessary for developing the discrete Sinc-Galerkin system. Let

$$E_1 = \left[ \vec{\eta}_1 : I \right]^T,$$

$$E_3 = \left[ \vec{\omega}_2 : I \right],$$

$$E_2^L = \left[ I : \vec{\omega}_1 \right],$$

and

$$E_2^R = \left[ I : \vec{\eta}_2 \right]^T.$$

Here  $\vec{\eta}_1$  is the  $m_y^1 \times 1$  vector with entries  $\eta_1(y_k^1)$  for  $-M_y^1 \leq k \leq N_y^1$ ,  $\vec{\eta}_2$  is the  $m_y^2 \times 1$  vector with entries  $\eta_2(y_k^2)$  for  $-M_y^2 \leq k \leq N_y^2$ ,  $\vec{\omega}_1$  is the  $m_x^1 \times 1$  vector with entries  $\omega_1(x_j^1)$  for  $-M_x^1 \leq j \leq N_x^1$ , and  $\vec{\omega}_2$  is the  $m_x^2 \times 1$  vector with entries  $\omega_2(x_j^2)$  for  $-M_x^2 \leq j \leq N_x^2$ .

Notice that these matrices can be used for point evaluations of the approximations as follows. Let  $U = [u_{jk}]$  be the  $m_x^1 \times (m_y^1 + 1)$  matrix of coefficients from  $\Omega^1$  in (5.2). Similarly, let  $V = [v_{jk}]$  be the  $(m_x^2 + 1) \times (m_y^2 + 1)$  matrix of coefficients from  $\Omega^2$  and let  $W = [w_{jk}]$  be the  $(m_x^2 + 1) \times m_y^2$  matrix of coefficients from  $\Omega^3$ . Then

$$\left[ u_{m_x^1, m_y^1}(x_p^1, y_q^1) \right] = U E_1,$$

$$\left[ v_{m_x^2, m_y^2}(x_p^2, y_q^2) \right] = E_2^L V E_2^R,$$

and

$$\left[ w_{m_x^2, m_y^2}(x_p^2, y_q^2) \right] = E_3 W.$$

Fix  $y = y_q^1 \in \Omega^1$ ,  $-M_y^1 \leq q \leq N_y^1$ . Along this line, the equation being solved

is

$$\begin{aligned}
 -u_{xx}(x, y_q^1) &= f(x, y_q^1) + u_{yy}(x, y_q^1), \quad (x, y_q^1) \in \Omega^1 \\
 u(a, y_q^1) &= u(\xi, y_q^1) = 0.
 \end{aligned}$$

This is a second-order boundary-value problem with homogeneous Dirichlet boundary conditions. The discrete system for this problem is

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \Gamma_{1/2}^1(x) U E_1 = F^1 + U_{yy}. \quad (5.3)$$

Here the matrices  $F^1 = [f(x_j^1, y_k^1)]$  and  $U_{yy} = [u_{yy}(x_j^1, y_k^1)]$  are of size  $m_x^1 \times m_y^1$ .  $\Gamma_{1/2}^1(x)$  is the  $m_x^1 \times m_x^1$  matrix  $\Gamma_r^1$  given in (2.12) with  $r = 1/2$  where the  $x$  is used to indicate the size of the matrix  $\Gamma_{1/2}^1$ . This matrix has the form

$$\Gamma_{1/2}^1(x) = \frac{-1}{(h_x^1)^2} I^{(2)} + \mathcal{D} \left( \frac{1}{4} \right). \quad (5.4)$$

Now fix  $x = x_p^2 \in \Omega^3$ ,  $-M_x^2 \leq p \leq N_x^2$ . Along this line, the equation being solved is

$$\begin{aligned}
 -w_{yy}(x_p^2, y) &= f(x_p^2, y) + w_{xx}(x_p^2, y), \quad (x_p^2, y) \in \Omega^3 \\
 w(x_p^2, s) &= w(x_p^2, \gamma) = 0.
 \end{aligned}$$

This is another second-order boundary-value problem with homogeneous Dirichlet boundary conditions. The discrete system for this problem is given by

$$E_3 W \left( \Gamma_{1/2}^2(y) \right)^T \mathcal{D} \left( ((\psi^2)')^{3/2} \right) = F^3 + W_{xx}. \quad (5.5)$$

Here  $\Gamma_{1/2}^2(y)$  is the same matrix given in (5.4) but it is of size  $m_y^2 \times m_y^2$ . The matrices  $F^3 = [f(x_j^2, y_k^2)]$  and  $W_{xx} = [w_{xx}(x_j^2, y_k^2)]$  are of size  $m_x^2 \times m_y^2$ .

Next, fix  $y = y_q^2 \in \Omega^2 \cup \Omega^3$ ,  $-M_y^2 \leq q \leq N_y^2$ . Along this line, the problem being solved is a second-order domain decomposition problem in the  $x$  direction. The problem is given by the equations



$$-v_{xx}(x, y_q^2) = f(x, y_q^2) + v_{yy}(x, y_q^2), \quad (x, y_q^2) \in \Omega^2$$

$$v(a, y_q^2) = 0$$

and

$$-w_{xx}(x, y_q^2) = f(x, y_q^2) + w_{yy}(x, y_q^2), \quad (x, y_q^2) \in \Omega^3$$

$$w(b, y_q^2) = 0$$

with matching conditions on the boundary given by

$$v(\xi, y_q^2) = w(\xi, y_q^2)$$

$$v_x(\xi, y_q^2) = w_x(\xi, y_q^2).$$

Notice that the evaluation of  $v$  at the point  $(x, y_q^2)$  is given by

$$\begin{aligned} v_{m_x^1, m_y^2}(x, y_q^2) &= \sum_{j=-M_x^1}^{N_x^1+1} \sum_{k=-M_y^2}^{N_y^2+1} v_{jk} \Psi_j^1(x) \Upsilon_k^2(y_q^2) \\ &= \sum_{j=-M_x^1}^{N_x^1+1} v_{jq} \Psi_j^1(x) + v_{j, N_y^2+1} \Psi_j^1(x) \Upsilon_{N_y^2+1}^2(y_q^2) \\ &= \sum_{j=-M_x^1}^{N_x^1+1} \Psi_j^1(x) [v_{jq} + v_{j, N_y^2+1} \eta_2(y_q^2)]. \end{aligned}$$

The  $(m_x^1 + 1) \times m_y^2$  matrix of these coefficients is given by  $VE_2^R$ . From (4.51), (4.59), and (4.55) come the matrix formulations arising from this problem,

$$\mathcal{D} \left( ((\phi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2(x) W = F^3 + W_{yy}, \quad (5.6)$$

$$B_p^1(x) V E_2^R + B_p^2(x) W = \Theta, \quad (5.7)$$

and

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1(x) V E_2^R = F^2 + V_{yy}, \quad (5.8)$$

respectively. Here the matrices  $F^3 = [f(x_j^2, y_k^2)]$  and  $W_{yy} = [w_{yy}(x_j^2, y_k^2)]$  are of size  $m_x^2 \times m_y^2$ ,  $F^2 = [f(x_j^1, y_k^2)]$  and  $V_{yy} = [v_{yy}(x_j^1, y_k^2)]$  are of size  $m_x^1 \times m_y^2$ , and  $\Theta$  is a zero

matrix of size  $2 \times m_y^2$ . The  $2 \times (m_x^j + 1)$  boundary matrices  $B_p^j(x)$  are given by (3.36) and (3.37), respectively. The matrix  $\mathcal{G}_{1/2}^1(x)$  is given by

$$\mathcal{G}_{1/2}^1(x) = \left[ \begin{array}{c|c} \Gamma_{1/2}^1(x) & \frac{-\bar{\omega}_1''}{((\phi^1)')^{3/2}} \end{array} \right]_{m_x^1 \times (m_x^1 + 1)},$$

and  $\mathcal{G}_{1/2}^2(x)$  is given by

$$\mathcal{G}_{1/2}^2(x) = \left[ \begin{array}{c|c} \frac{-\bar{\omega}_2''}{((\phi^2)')^{3/2}} & \Gamma_{1/2}^2(x) \end{array} \right]_{m_x^2 \times (m_x^2 + 1)}.$$

The vectors  $\bar{\omega}_1''$  and  $\bar{\omega}_2''$  have entries  $\omega_1''(x_j^1)$  and  $\omega_2''(x_j^2)$  and are of size  $m_x^1 \times 1$  and  $m_x^2 \times 1$ , respectively.

Finally, fix  $x = x_p^1 \in \Omega^1 \cup \Omega^2$  for  $-M_x^1 \leq p \leq N_x^1$ . This is a second-order domain decomposition problem in the  $y$  direction. The equations are given by

$$-u_{yy}(x_p^1, y) = f(x_p^1, y) + u_{xx}(x_p^1, y), \quad (x_p^1, y) \in \Omega^1$$

$$u(x_p^1, t) = 0,$$

$$-v_{yy}(x_p^1, y) = f(x_p^1, y) + v_{xx}(x_p^1, y), \quad (x_p^1, y) \in \Omega^2$$

$$v(x_p^1, s) = 0,$$

with matching boundary conditions

$$v(x_p^1, \gamma) = u(x_p^1, \gamma)$$

$$v_y(x_p^1, \gamma) = u_y(x_p^1, \gamma).$$

As before, we need the coefficients used in evaluation of  $v$  along the line  $x = x_p^1$ . They are given by the  $m_x^1 \times (m_y^2 + 1)$  matrix  $E_2^L V$ . The matrix equations for the system are given by

$$U \left( \mathcal{G}_{1/2}^1(y) \right)^T \mathcal{D} \left( ((\psi^1)')^{3/2} \right) = F^1 + U_{xx}, \quad (5.9)$$

$$U (B_p^1(y))^T + E_2^L V (B_p^2(y))^T = \Theta , \quad (5.10)$$

and

$$E_2^L V (\mathcal{G}_{1/2}^2(y))^T \mathcal{D} (((\psi^2)')^{3/2}) = F^2 + V_{xx} . \quad (5.11)$$

Here the  $2 \times (m_y^j + 1)$  boundary matrices  $B_p^j(y)$  are given by

$$B_p^1(y) = \begin{bmatrix} \vec{b}_6 \\ \vec{e}_1 \end{bmatrix} ,$$

and

$$B_p^2(y) = \begin{bmatrix} \vec{b}_5 \\ -\vec{e}_{m_y^2+1} \end{bmatrix} ,$$

respectively, where

$$\vec{b}_5 = [ S_{-M^2}^2(\gamma - 2\delta) - 4S_{-M^2}^2(\gamma - \delta) \quad \dots \quad S_{N^2}^2(\gamma - 2\delta) - 4S_{N^2}^2(\gamma - \delta)$$

$$\eta_2(\gamma - 2\delta) - 4\eta_2(\gamma - \delta) + 3 ]$$

and

$$\vec{b}_6 = [ \eta_1(\gamma + 2\delta) - 4\eta_1(\gamma + \delta) + 3 \quad S_{-M^1}^1(\gamma + 2\delta) - 4S_{-M^1}^1(\gamma + \delta) \quad \dots$$

$$S_{N^1}^1(\gamma + 2\delta) - 4S_{N^1}^1(\gamma + \delta) ] .$$

The matrices  $F^1 = [f(x_j^1, y_k^1)]$  and  $U_{xx} = [u_{xx}(x_j^1, y_k^1)]$  are of size  $m_x^1 \times m_y^1$ , and the matrices  $F^2 = [f(x_j^1, y_k^2)]$  and  $V_{yy} = [v_{yy}(x_j^1, y_k^2)]$  are of size  $m_x^1 \times m_y^2$ . The zero matrix  $\Theta$  is of size  $m_x^1 \times 2$ . The matrices  $\mathcal{G}_{1/2}^j(y)$  for  $j = 1, 2$  are given by

$$\mathcal{G}_{1/2}^1(y) = \left[ \begin{array}{c|c} \frac{-\vec{\eta}_1''}{((\psi^1)')^{3/2}} & \Gamma_{1/2}^1(y) \end{array} \right]_{m_y^1 \times (m_y^1 + 1)}$$

and

$$\mathcal{G}_{1/2}^2(y) = \left[ \begin{array}{c|c} \Gamma_{1/2}^2(y) & \frac{-\vec{\eta}_2''}{((\psi^2)')^{3/2}} \end{array} \right]_{m_y^2 \times (m_y^2 + 1)} ,$$

The vectors  $\vec{\eta}_1''$  and  $\vec{\eta}_2''$  have entries  $\eta_1''(y_k^1)$  and  $\eta_2''(y_k^2)$  and are of size  $m_y^1 \times 1$  and  $m_y^2 \times 1$ , respectively.

Adding (5.3) and (5.9) and cancelling the appropriate terms (since  $F^1 + U_{xx} + U_{yy} = 0$ ) the resulting equation is

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \Gamma_{1/2}^1(x) U E_1 + U \left( \mathcal{G}_{1/2}^1(y) \right)^T \mathcal{D} \left( ((\psi^1)')^{3/2} \right) = F^1. \quad (5.12)$$

Adding (5.8) and (5.11) yields

$$\mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1(x) V E_2^R + E_2^L V \left( \mathcal{G}_{1/2}^2(y) \right)^T \mathcal{D} \left( ((\psi^2)')^{3/2} \right) = F^2. \quad (5.13)$$

Similarly, adding (5.5) and (5.6) yields the equation

$$E_3 W \left( \Gamma_{1/2}^2(y) \right)^T \mathcal{D} \left( ((\psi^2)')^{3/2} \right) + \mathcal{D} \left( ((\phi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2(x) W = F^3. \quad (5.14)$$

Finally concatenate (5.12), (5.13), (5.14), (5.7), and (5.10) and apply Theorem 2.7 to achieve the following set of five equations

$$Aco(U) = co(F^1)$$

$$Bco(V) = co(F^2)$$

$$Cco(W) = co(F^3)$$

$$Dco(V) + Eco(W) = \vec{\theta}^2$$

$$Gco(U) + Hco(V) = \vec{\theta}^1.$$

Here  $A$  is of size  $m_x^1 m_y^1 \times m_x^1 (m_y^1 + 1)$  and is given by

$$A = E_1^T \otimes \mathcal{D} \left( ((\phi^1)')^{3/2} \right) \Gamma_{1/2}^1(x) + \mathcal{D} \left( ((\psi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1(y) \otimes I.$$

The matrix  $B$  is of size  $m_x^1 m_y^2 \times (m_x^1 + 1)(m_y^2 + 1)$  and is given by

$$B = \left( E_2^R \right)^T \otimes \mathcal{D} \left( ((\phi^1)')^{3/2} \right) \mathcal{G}_{1/2}^1(x) + \mathcal{D} \left( ((\psi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2(y) \otimes E_2^L.$$

The matrix  $C$  is of size  $m_x^2 m_y^2 \times (m_x^2 + 1)m_y^2$  and is given by

$$C = \mathcal{D} \left( ((\psi^2)')^{3/2} \right) \Gamma_{1/2}^2(y) \otimes E_3 + I \otimes \mathcal{D} \left( ((\phi^2)')^{3/2} \right) \mathcal{G}_{1/2}^2(x).$$

The boundary matrices are  $D$ , which is of size  $2m_y^2 \times (m_x^1 + 1)(m_y^2 + 1)$ ;  $E$ , which is of size  $2m_y^2 \times (m_y^2)(m_x^2 + 1)$ ;  $G$ , which is of size  $2m_x^1 \times m_x^1(m_y^1 + 1)$ ; and  $H$ , which is of size  $2m_x^1 \times (m_x^1 + 1)(m_y^2 + 1)$ . They are given by

$$D = \left( E_2^R \right)^T \otimes B_p^1(x),$$

$$E = I \otimes B_p^2(x),$$

$$G = B_p^1(y) \otimes I,$$

and

$$H = B_p^2(y) \otimes E_2^L.$$

Lastly, the vector  $\bar{\theta}^1$  is of size  $2m_x^1 \times 1$  and the vector  $\bar{\theta}^2$  is of size  $2m_y^2 \times 1$ . The matrix system formed from these equations is under-determined. This can be fixed by noticing that one condition never addressed is  $v_{m_x^1, m_y^2}(\xi, \gamma) = 0$ .

In each of the following four sample problems

$$\Omega = \{(x, y) : -1 < x < 1, 0 < y < 2\} \cup \{(x, y) : -1 < x < 4, 0 < y < 1\}.$$

Thus in this section,  $\Omega^1$ ,  $\Omega^2$ , and  $\Omega^3$  are chosen to be

$$\Omega^1 = \{(x, y) : -1 < x < 1, 1 < y < 2\},$$

$$\Omega^2 = \{(x, y) : -1 < x < 1, 0 < y < 1\}.$$

and

$$\Omega^3 = \{(x, y) : 1 < x < 4, 0 < y < 1\}.$$

As before,  $d$  is chosen to be  $\pi/2$ , and  $\alpha$  is chosen to be 1. Let the sinc error be defined by

$$\|E_S\| = \max_{(x,y) \in S} |u(x,y) - u_A(x,y)|$$

where

$$\begin{aligned} S &= \{x_j^1 : -M_x^1 \leq j \leq N_x^1\} \times \{y_k^1 : -M_y^1 \leq k \leq N_y^1\} \\ &\cup \{x_j^1 : -M_x^1 \leq j \leq N_x^1\} \times \{y_k^2 : -M_y^2 \leq k \leq N_y^2\} \\ &\cup \{x_j^2 : -M_x^2 \leq j \leq N_x^2\} \times \{y_k^2 : -M_y^2 \leq k \leq N_y^2\} \end{aligned}$$

is the set of all grid points generated from the Sinc-Galerkin method and  $u_A$  is given by

$$u_A(x,y) = \begin{cases} u_{m_x^1, m_y^1}(x,y), & (x,y) \in \Omega^1 \\ v_{m_x^1, m_y^2}(x,y), & (x,y) \in \Omega^2 \\ w_{m_x^2, m_y^2}(x,y), & (x,y) \in \Omega^3 \end{cases}$$

Similarly define a uniform error by

$$\|E_U\| = \max_{(x,y) \in U} |u(x,y) - u_A(x,y)| \quad (5.15)$$

where

$$\begin{aligned} U &= \{(-1 + 5j/100, k/50) : 0 \leq j \leq 100, 0 \leq k \leq 50\} \\ &\cup \{(-1 + 5j/100, 1 + k/50) : 0 \leq j \leq 40, 0 \leq k \leq 50\} \end{aligned}$$

is a uniform grid over  $\Omega$ . Notice that because of the homogeneous Dirichlet boundary conditions on  $\Omega$ , these test problems have solutions that are necessarily zero along the interior boundaries,  $\Gamma^1$  and  $\Gamma^2$ . Introducing non-homogeneous Dirichlet boundary conditions is the simplest way to accommodate nonzero solutions along  $\Gamma^1$  and  $\Gamma^2$ . This is then easily taken care of by adding boundary basis functions similar to those used along the introduced boundaries. See [17] for details. In order to more easily

and clearly define the method, only homogeneous Dirichlet boundary conditions are presented here.

**Example 5.1** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x-1)(x+1)(4-x)y(y-1)(y-2)}{3.1596}.$$

The choices  $M \equiv M_x^1 = M_x^2 = M_y^1 = M_y^2$  and  $N \equiv N_x^1 = N_x^2 = N_y^1 = N_y^2$  are made.

The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y^1 = h_y^2 = \pi/\sqrt{2M}$ . In

order to present the best view of the mesh plots, the domain  $\Omega$ , seen in Figure 25,

was rotated. This orientation in  $\mathbb{R}^3$  can be seen in Figure 26. A mesh plot of the

approximate solution with  $M = 8$  is shown in Figure 27, and a contour plot which

displays the convergence of the approximations for increasing  $M$  is shown in Figure

28. As seen in Table 25, the method performs well on this problem.

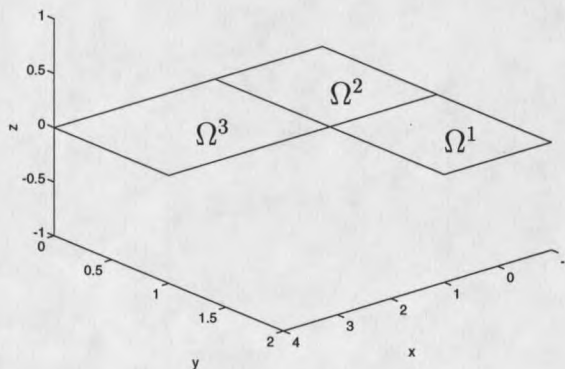


Figure 26: The rotated domain  $\Omega$

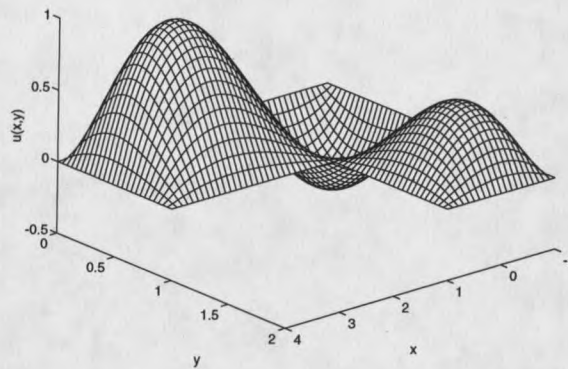
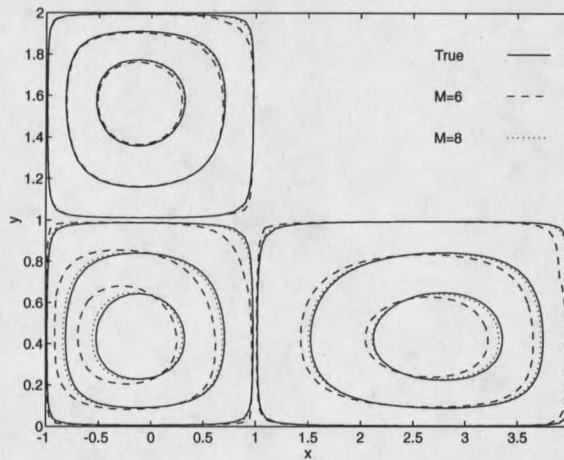
Figure 27: Sinc-Galerkin patching solution for Example 5.1 with  $M = 8$ 

Figure 28: Contour plots from the Sinc-Galerkin patching solution for Example 5.1

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$1.9171e - 01$	$2.7847e - 01$
4	$1.1107e + 00$	$1.9381e - 01$	$2.0510e - 01$
6	$9.0690e - 01$	$1.2459e - 01$	$1.3278e - 01$
8	$7.8540e - 01$	$2.8859e - 02$	$3.1669e - 02$
10	$7.0248e - 01$	$1.1538e - 02$	$1.2919e - 02$

Table 25: Error in the Sinc-Galerkin patching method for Example 5.1



**Example 5.2** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x-1)^2(x+1)^2(4-x)^2y^2(y-1)^2(y-2)^2}{9.9057}.$$

This problem is analogous to that given in Example 4.1. The choices  $M \equiv M_x^1 = M_x^2 = M_y^1 = M_y^2$  and  $N \equiv N_x^1 = N_x^2 = N_y^1 = N_y^2$  are made. The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y^1 = h_y^2 = \pi/\sqrt{2M}$ . A mesh plot of the approximate solution with  $M = 8$  is shown in Figure 29, and a contour plot which displays the convergence of the approximations for increasing  $M$  is shown in Figure 30. As seen in Table 26, the method performs well on this problem and the results are consistent with those of Example 4.1. A comparison of the results in Table 25 and 26 shows that in this example convergence is more rapid but the approximation starts off less accurate.

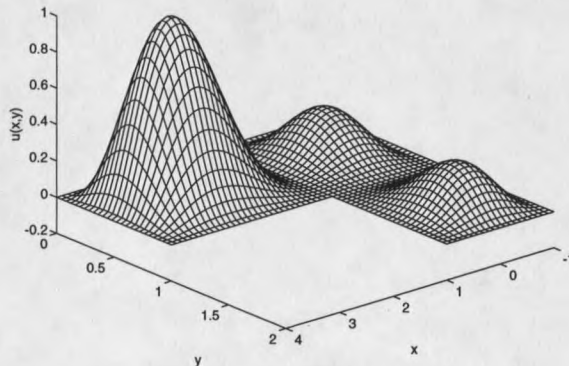


Figure 29: Sinc-Galerkin patching solution for Example 5.2 with  $M = 8$

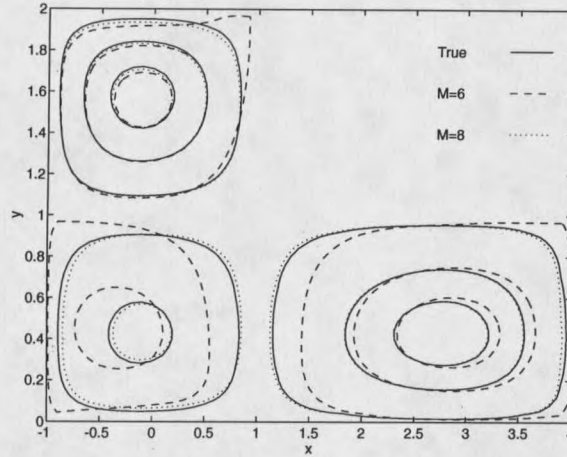


Figure 30: Contour plots from the Sinc-Galerkin patching solution for Example 5.2

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$2.6124e - 00$	$3.1236e - 00$
4	$1.1107e + 00$	$7.8281e - 01$	$8.1220e - 01$
6	$9.0690e - 01$	$1.4393e - 01$	$1.4717e - 01$
8	$7.8540e - 01$	$1.3051e - 02$	$1.3920e - 02$
10	$7.0248e - 01$	$6.0314e - 03$	$6.2352e - 03$

Table 26: Error in the Sinc-Galerkin patching method for Example 5.2

**Example 5.3** Consider the problem

$$\begin{aligned} -\Delta u(x, y) &= f(x, y), \quad (x, y) \in \Omega \\ u(x, y) &= 0, \quad (x, y) \in \partial\Omega, \end{aligned}$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x-1)(x+1)\sqrt{4-x}(y-1)(y-2)\sqrt{y}}{5.4371}$$

This problem is similar to that given in Example 4.6. The choices  $M \equiv M_x^1 = M_x^2 = M_y^1 = M_y^2$  and  $N \equiv N_x^1 = N_x^2 = N_y^1 = N_y^2$  are made. The choice  $\alpha = 1$  implies that  $M = N$  and  $h = h_x^1 = h_x^2 = h_y^1 = h_y^2 = \pi/\sqrt{2M}$ . A mesh plot of the approximate solution with  $M = 8$  is shown in Figure 31, and a contour plot which displays the convergence of the approximations for increasing  $M$  is shown in Figure 32. As seen in Table 27, the method performs well on this problem and the results are consistent with those of Example 4.6. Notice that the singularities in this problem are located along the lines  $x = 4$  and  $y = 0$ . To take advantage of the location of these singularities, set  $M_x^1$  and  $M_y^1$  to be small numbers, and allow  $M_x^2 = M_y^2$  to vary. As seen in Table 28, for  $M_x^2 = M_y^2 = 10$  the same accuracy can be obtained with  $M_x^1 = M_y^1 = 6$ , and thus  $h_x^1 = h_y^1 = .90690$ . Additionally, the decrease in system size allows the case  $M_x^2 = 12$  to be run.

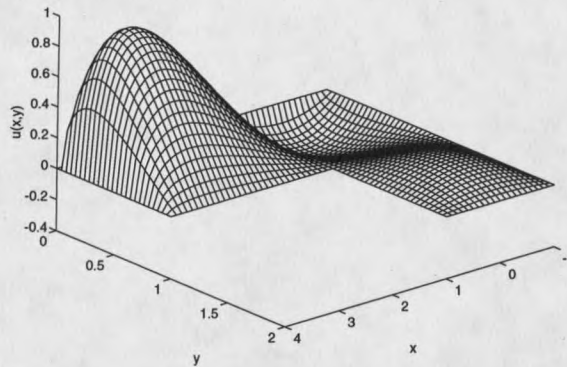
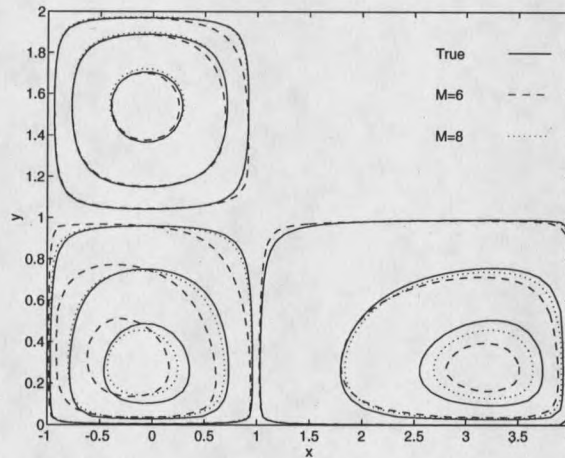
Figure 31: Sinc-Galerkin patching solution for Example 5.3 with  $M = 8$ 

Figure 32: Contour plots from the Sinc-Galerkin patching solution for Example 5.3

$M$	$h$	$\ E_S\ $	$\ E_U\ $
2	$1.5708e + 00$	$3.0523e - 01$	$3.2472e - 01$
4	$1.1107e + 00$	$2.0470e - 01$	$1.9217e - 01$
6	$9.0690e - 01$	$1.3837e - 01$	$1.9645e - 01$
8	$7.8540e - 01$	$9.5542e - 02$	$1.0058e - 01$
10	$7.0248e - 01$	$6.7841e - 02$	$6.9221e - 02$

Table 27: Error in the Sinc-Galerkin patching method for Example 5.3

$M_x^2 = M_y^2$	$h_x^2 = h_y^2$	$\ E_S\ $	$\ E_U\ $
2	1.5708e + 00	1.3275e + 01	1.6965e + 01
4	1.1107e + 00	2.4700e + 00	2.6105e + 00
6	9.0690e - 01	1.3837e - 01	1.8585e - 01
8	7.8540e - 01	9.5542e - 02	1.6747e - 01
10	7.0248e - 01	6.7841e - 02	5.8683e - 02
12	6.4127e - 01	5.0452e - 02	4.1236e - 02

Table 28: Error in the Sinc-Galerkin patching method for Example 5.3 with  $M_x^1 = M_y^1 = 6$ .

**Example 5.4** Consider the problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega,$$

where  $f(x, y)$  is chosen so that the true solution is given by

$$u(x, y) = \frac{(x-1)(x+1)\sqrt{4-x}(y-1)(y-2)y}{3.1877}.$$

The choices  $M \equiv M_x^1 = M_x^2 = M_y^1 = M_y^2$  and  $N \equiv N_x^1 = N_x^2 = N_y^1 = N_y^2$  are made. The choice  $\alpha = 1$  implies that  $M = N$  and  $h \equiv h_x^1 = h_x^2 = h_y^1 = h_y^2 = \pi/\sqrt{2M}$ . A mesh plot of the approximate solution with  $M = 8$  is shown in Figure 33, and a contour plot which displays the convergence of the approximations for increasing  $M$  is shown in Figure 34. As seen in Table 29, the method performs well on this problem and the results are consistent with those of Example 4.6 and Example 5.3.











