A Fast Spectral Subtractional Solver for Elliptic Equations

Elena Braverman[1]
Dept. of Mathematics and Statistics, University of Calgary,
2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada
Boris Epstein
The Academic College of Tel Aviv - Yaffo, Antokolsky Str. 4,
Tel Aviv 64044, Israel,
Moshe Israeli[2]
Technion-Israel Institute of Technology, Computer Science Dept.,
Haifa 32000, Israel
and
Amir Averbuch
School of Computer Sciences, Tel Aviv University,
Tel Aviv 69978, Israel

Corresponding author: Dr. Elena Braverman
Corresponding author's mailing Address: Department of Mathematics and Statistics, University of Calgary, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada
Corresponding author's e-mail maelena@math.ucalgary.ca
Corresponding author's phone: (403)-220-3956
Corresponding author's Fax: (403)-282-5150

Abstract

The paper presents a fast subtractional spectral algorithm for the solution of the Poisson equation and the Helmholtz equation which does not require an extension of the original domain. It takes $O(N^2 \log N)$ operations, where N is the number of collocation points in each direction. The method is based on the eigenfunction expansion of the right hand side with integration and the successive solution of the corresponding homogeneous equation using Modified Fourier Method. Both the right hand side and the boundary conditions are not assumed to have any periodicity properties. This algorithm is used as a preconditioner for the iterative solution of elliptic equations

[1]The research was partially supported by the University Research Grant of the University of Calgary.
[2]The research was partially supported by the VPR fund for promotion of research at the Technion.

with non-constant coefficients. The procedure enjoys the following properties: fast convergence and high accuracy even when the computation employs a small number of collocation points. We also apply the basic solver to the solution of the Poisson equation in complex geometries.

AMS subject classifications. 65N35, 65T20
Keywords: Fast spectral direct solver, the Poisson equation, the modified Helmholtz equation, preconditioned iterative algorithm for elliptic equations, equations in complex geometries.

1 Introduction

A variety of problems in computational physics require the solution of the Poisson equation and the modified Helmholtz equation, for example, these equations arise in the determination of the pressure field for incompressible Navier-Stokes equations after the semi-implicit discretization in time [21].

Application of high-order (pseudo) spectral methods, which are based on global expansions into orthogonal polynomials (Chebyshev or Legendre polynomials), to the solution of elliptic equations, results in full (dense) matrix problems. The cost of inverting a full $N \times N$ matrix without using special properties is $O(N^3)$ operations [8]. Besides, the accuracy decreases considerably with the growth of N due to accumulation of round-off errors. The computation becomes more efficient if the domain is decomposed into smaller subdomains where partial solutions are found and subsequently patched by the Fast Multipole Method [17, 19].

It is well known that a Fourier method for the solution of the Poisson or the Helmholtz equation in principle has an exponential convergence but faces the Gibbs phenomenon for non-periodic boundary conditions [15]. The methods to resolve the Gibbs phenomenon are described in [16] (see also references to this review article). They can be classified as Fourier space filters and methods concerned with an adjustment in a physical space. For the solution of the Poisson equation or the modified Helmholtz equation with Fourier series we have to restore a solution rather than the original right hand side (RHS) which is presented in the Fourier space. Since the accuracy of the solution degrades due to the Gibbs phenomenon in the RHS representations, then the algorithm can benefit if the RHS is presented as a sum of a smooth periodic function and another function which can be integrated analytically. Rather simple for 1D problems, the implementation of this idea becomes more complicated for higher dimensions. This procedure is called sometimes the subtraction technique (a function which is later integrated analytically is subtracted from the RHS). It is to be emphasized that the subtraction methods can resolve the problem of poor convergence in cases of corner singularities when the boundary conditions are discontinuous in a corner or do not match the right hand side [1, 6]. These methods can be incorporated with any numerical scheme, when boundary conditions degrade the accuracy, as far as appropriate subtraction functions can be found. From this point of view, the present paper is an illustration how the subtractional technique can be successfully involved in the Fourier 2-D algorithm for the solution of the Poisson, Helmholtz and modified Helmholtz equations.

To the best of our knowledge the application of the subtraction technique in the resolution of the Gibbs phenomenon for the Fourier series solution goes back to Sköllermo [25], where a modification of the Fourier method was developed for the Poisson equation

$$\Delta u = f \tag{1.1}$$

in the rectangle $[0, 1] \times [0, 1]$ with periodic boundary conditions. The solution involved the following steps:

1. The right hand side was presented as a sum of two functions $f(x, y) = f_1(x, y)+p_1(x, y)$, where f_1 is a smooth function in the rectangle while $p_1(x, y)$ is a polynomial function having the same values at the corners as $f(x, y)$.

2. The solution $P_1(x, y)$ which corresponds to the right hand side $p_1(x, y)$ can be computed analytically; the "smoothed" function $f_1(x, y)$ is expanded into sine series

$$f_1(x, y) \sim \sum\sum \sin k\pi x \sin l\pi y, \text{ denote}$$

$$u_1(x, y) \sim -\sum\sum \frac{1}{\pi^2(k^2 + l^2)} \sin k\pi x \sin l\pi y.$$

3. Finally, the sum $u(x, y) = u_1(x, y)+P_1(x, y)$ is an approximate solution of the problem.

We solve the Poisson equation or the modified Helmholtz equation in the rectangular domain with an equispaced grid. Then the subtraction technique (in the physical space) should be used for the resolution of the Gibbs phenomenon rather than other methods due to the following reasons.

a). After subtraction, Fast Fourier Transform can be applied to the remaining part of RHS with high convergence.

b). The algorithm keeps the diagonal representation of the Laplace operator, so, unlike Chebyshev and Legendre expansions, it is not necessary to find an inverse of a full matrix.

c). Generally, the computation of the subtraction functions is even less time consuming than FFT implementation.

It is to be noted that the subtraction algorithm in [25] was developed for some specific boundary conditions only.

For arbitrary boundary conditions and arbitrary RHS the Gibbs phenomenon in spectral Fourier method can be essentially reduced by the following procedure. The function f in the right hand side is extended to a larger domain and replaced by a new function which coincides with f in the original domain and it is periodic together with a certain number of its derivatives in the larger domain. A considerable effort was concerned with a construction of a smooth extension of the right hand side (see, for example, [1, 2, 11] and references therein). The accuracy of the solution strongly depends on the number of extension points [12]. At the next step the extended function is multiplied by a bell function, so that the resulting function coincides with the right hand side in the original domain and vanishes together with some of its first derivatives in the extended domain. This procedure is called folding and is described in the Appendix. The method is fast ($O((N+\varepsilon)^2 \log N)$) operations, where N is the number of points in each direction and ε is the number of extension points) and gives any prescribed polynomial rate of convergence [1, 2, 6, 7]. However, if we use the Poisson solver or the Helmholtz solver as a part of an iterative procedure or as a part of a time-dependent problem then we have to construct an extension of the right hand side at each time or iteration step.

The purpose of the present paper is to develop a direct robust and fast ($O(N^2 \log N)$, where N is a number of points in each direction) spectral solver which can be employed for iterative solution of equations with non-constant coefficients and does not assume the extension of the original domain. This algorithm is based on the following previously developed methods:

1. Since we solve a boundary value problem for the Poisson (Helmholtz) equation then a final part of our scheme is the solution of the corresponding homogeneous equation. The boundary conditions is the difference of the original boundary conditions and ones obtained when the nonhomogeneous equation is solved. Such spectral solver was developed in [1, 2]. We apply the algorithm of [1] without any modifications. It is also based on the subtraction technique incorporated with the Fourier method.

2. Polynomial type subtraction functions (of a degree higher than 2) [25, 2] are not optimal due to possible oscillations, high amplitudes or rates of change which can also degrade the accuracy of the Fourier expansion of the remaining part. Exponential-type functions [1] appeared to be more efficient in subtractional algorithms for the solution of the Poisson equation and Helmholtz equations. This idea was also employed for 3D problems in [6, 7, 20].

Here we present a fast ($O(N^2 \log N)$) direct spectral solver for the solution of the Poisson equation, the modified Helmholtz equation and the Helmholtz equation. The advantage (compared to the solvers of [1, 2]) is that it does not require an extension of the right hand side to a wider domain. In addition, we present an iterative spectral algorithm for a high order solution of elliptic equations with non-constant coefficients

$$\mathcal{L}u = f \tag{1.2}$$

in rectangular subdomains. The method is based on preconditioning and employs a fast direct Poisson solver as a preconditioner.

The idea of preconditioning is to solve the equation

$$\mathcal{L}_0^{-1}\mathcal{L}u = \mathcal{L}_0^{-1}f \tag{1.3}$$

rather than (1.2). The preconditioned iterations have the form

$$\mathcal{L}_0 u^{n+1} = f - (\mathcal{L} - \mathcal{L}_0)u^n. \tag{1.4}$$

Equations (1.2) and (1.3) have the same solution but the iterative method for the solution of Eq. (1.3) may converge much faster if the condition number of the operator $\mathcal{L}_0^{-1}\mathcal{L}$ is bounded. In practical implementations, the rate of convergence depends on the "closeness" of the operator $\mathcal{L}_0$ to $\mathcal{L}$. Low order approximations (finite-difference or finite-element) to $\mathcal{L}$ are usually applied as $\mathcal{L}_0$ [23]. The use of high order finite-difference preconditioners improves the rate of convergence but requires to invert a dense matrix at each iteration step which makes these preconditioners inefficient. We use $\mathcal{L}_0$ which is an elliptic operator with constant coefficients. The operator $\mathcal{L}_0$ is efficiently inverted by a fast Poisson solver which is described in Section 2. Two other factors that can speed the rate of convergence: choice of constants in the Poisson operator and the use of local relaxation parameter are discussed in Section 4. Similar approach for problems with periodic boundary conditions was developed in [26].

This method is also applicable for the solution of the Poisson equation in complex geometries. Numerical methods for the Poisson equation in irregular domains and simulations of CFD problems in complex geometries were considered in [3, 10, 22, 26, 13, 14], see also references therein. Generally [15], spectral solution of equations in complex geometries employs either transformation or patching. The present algorithm can be incorporated in the solution of the Poisson equations in complex geometries in the following ways:

1. By applying some transformation, the equation with constant coefficients in an irregular domain can be transformed into an equation with non-constant coefficients in a rectangular domain. The latter equation can be solved by the preconditioned iterations. This scheme is described and implemented in Section 5.

2. An irregular domain can be decomposed into subdomains, most of them of rectangular geometry. Then by applying the present solver in regular domains and some other solver (for example, [22]) in irregular subdomains with successive derivatives patching (for the appropriate spectral algorithm see [20]) we obtain a smooth solution in complex geometries.

3. The present algorithm can be a part of methods using the extension of the original complex domain to an involving regular domain [13, 14]. For example, if a part of the original boundary coincides with a certain rectangle, then the present algorithm should be used rather than the extension of the original domain to a larger rectangle.

The paper is organized as follows. A direct method for the solution of the Poisson equation, the Helmholtz equation and the modified Helmholtz equation is presented in Section 2. This method does not require the extension of the right hand side beyond the computational domain. Section 3 contains relevant numerical examples. The direct Poisson solver is used as the basic kernel for the solution of elliptic equation with variable coefficients as described in Section 4. In Section 5 the above scheme is applied to the solution of the Poisson equation in domains with complex geometry.

2 Subtractional spectral solvers

2.1 General description and a one-dimensional example

As we will see in Section 4, the proposed solution of the elliptic equation with non-constant coefficients will be based on a fast solver for the Poisson equation with constant coefficients. This efficient solver is employed as a preconditioner for the solution of the equation with non-constant coefficients. A 2-D and 3-D fast solvers for the Poisson equations were proposed in [1, 2, 6]. They can be used as the basic kernel for the proposed algorithm. However they are based on extension of the right hand side to a wider domain on each iterative step. In this section we propose a new modified algorithm where we do not need to extend the domain while retaining the spectral accuracy as in the original method. In addition, as before, the algorithm has a good performance even on a domain with a small number of grid points.

The idea of the algorithm is the following. The right hand side of the Poisson equation is presented as a sum of functions and series. Each term of the sum is either an eigenfunction of the Laplace operator or a Laplacian of some known function. The series are Fourier type series. As a result we obtain an efficient direct spectral algorithm for the solution of the Poisson equation based on the Fast Fourier Transform and the subtraction technique.

The methods to accelerate the convergence of the Fourier series (including 2-D case) were discussed in [4, 5]. Here we illustrate our idea on a simple 1-D example.

We solve a 1-D equation $u'' = f$ on the segment $[0,1]$. The expansion of RHS in the sine series is not accurate as far as $f(0), f(1)$ do not vanish with some of its first even derivatives. If we define

$$f_0(x) = f(0)\frac{\sinh(1-x)}{\sinh 1} + f(1)\frac{\sinh x}{\sinh 1},$$

then $g(x) = f(x) - f_0(x)$ vanishes at the endpoints. Further, if $\dfrac{\partial^2 g}{\partial x^2}(0) = A$, $\dfrac{\partial^2 g}{\partial x^2}(1) = B$, then

$$f_2(x) = \frac{A}{\lambda_1^2 - \lambda_2^2}\left[\frac{\sinh(\lambda_1(1-x))}{\sinh(\lambda_1)} - \frac{\sinh(\lambda_2(1-x))}{\sinh(\lambda_2)}\right] + \frac{B}{\lambda_1^2 - \lambda_2^2}\left[\frac{\sinh(\lambda_1 x)}{\sinh(\lambda_1)} - \frac{\sinh(\lambda_2 x)}{\sinh(\lambda_2)}\right]$$

subtracts the second derivatives at the endpoints. Similarly $f_4(x)$ can be constructed. Functions f_0, f_2, f_4 can be easily integrated analytically. Table 1 illustrates the convergence of the subtractional Poisson method for 1D equation $u'' = f$ with RHS

$$f(x) = \exp\{\alpha(x-x_0)^2[2\alpha^2(x-x_0)^2 - \alpha]\}$$

on the segment $[0,1]$. Here three subtractional steps were applied, i.e. f was presented as a sum $f(x) = \varphi(x) + f_0(x) + f_2(x) + f_4(x)$. Here $\varphi(x)$ can be expanded into sine series with $O(1/N^6)$ accuracy.

Assume that u is the exact solution of Eq. (2.1) and u' is the computed solution. Here and later on we will use the following measures to estimate the errors:

$$\begin{aligned}
\varepsilon_{MAX} &= \max|u_i' - u_i| \\
\varepsilon_{MSQ} &= \sqrt{\frac{\sum_{i=1}^{N}(u_i'-u_i)^2}{N}} \\
\varepsilon_{\mathcal{L}^2} &= \sqrt{\frac{\sum_{i=1}^{N}(u_i'-u_i)^2}{\sum_{i=1}^{N} u_i^2}}
\end{aligned}$$

α, x_0	$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
$\alpha = 4, x_0 = 0.8$	16×16	4.1e-6	2.0e-6	2.6e-6
	32×32	4.1e-8	1.9e-8	2.5e-8
	64×64	3.5e-10	1.6e-10	2.1e-10
	128×128	3.1e-12	1.7e-12	1.7e-12
$\alpha = 200, x_0 = 0.5$	16×16	1.5e-2	4.0e-3	5.4e-3
	32×32	9.3e-8	4.6e-8	6.1e-8
	64×64	3.5e-10	1.7e-10	2.3e-10
	128×128	2.8e-12	1.4e-12	1.8e-12

Table 1: MAX, MSQ and $\mathcal{L}^2$ errors for the exact solution $u(x) = \exp\{\alpha(x - x_0)^2\}$ of the 1D Poisson equation in the domain $[0, 1]$.

It is to be noted that as a subtraction part an exponential like functions are used rather than polynomial; the latter can expose an oscillatory behavior.

2.2 Outline of the algorithm

Consider the Poisson equation in a rectangular domain:

$$\Delta u(x, y) = f(x, y), \qquad 0 \le x \le a, \quad 0 \le y \le b, \tag{2.1}$$

with the Dirichlet or Neumann boundary conditions. The solution incorporates two steps:

1. An arbitrary solution u_1 of the nonhomogeneous equation (2.1) is found.
2. A solution u_2 of the boundary problem for the homogeneous equation is derived, such that $u_1 + u_2$ satisfies the original boundary conditions.

The Modified Fourier algorithm for Step 2 developed in [1, 2] was used without any changes, so we will concentrate on Step 1 of the algorithm.

The Poisson solver is based on the eigenfunction method. If the right hand side of Eq. (2.1) is smooth and vanishes together with some first even derivatives at the boundaries, then it can be presented as a fast converging sine series:

$$f(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_{mn} \sin\left(\frac{n\pi x}{a}\right) \sin\left(\frac{m\pi y}{b}\right). \tag{2.2}$$

Then, a solution of Eq. (2.1) can be written as

$$u(x, y) = -\sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{a_{mn}}{\pi^2\left(\frac{n^2}{a^2} + \frac{m^2}{b^2}\right)} \sin\left(\frac{n\pi x}{a}\right) \sin\left(\frac{m\pi y}{b}\right). \tag{2.3}$$

Assume that $f(x, y)$ does not vanish at the boundaries

$$f(x, 0) = \varphi_1(x), \quad f(x, b) = \varphi_2(x), \quad f(0, y) = \varphi_3(y), \quad f(b, y) = \varphi_4(y), \tag{2.4}$$

while the functions $\varphi_i,\ i=1,\dots,4$, can be expanded into sine series

$$\varphi_1(x)=\sum_{k=1}^{\infty} b_{1k}\sin(\lambda_{1k}x),\quad \varphi_2(x)=\sum_{k=1}^{\infty} b_{2k}\sin(\lambda_{2k}x),$$

$$\varphi_3(y)=\sum_{k=1}^{\infty} b_{3k}\sin(\lambda_{3k}y),\quad \varphi_4(y)=\sum_{k=1}^{\infty} b_{4k}\sin(\lambda_{4k}y), \tag{2.5}$$

where

$$\lambda_{1k}=\lambda_{2k}=\frac{\pi k}{a},\quad \lambda_{3k}=\lambda_{4k}=\frac{\pi k}{b}. \tag{2.6}$$

Let us determine the function f_{edge} which coincides with f at the boundaries and such that the solution of the Poisson equation

$$\Delta u = f_{edge}$$

can be immediately solved. If we define the following function (see Fig. 1 for edge $x=0$)

$$\begin{aligned} f_{edge}(x,y) &= \sum_{k=1}^{\infty}\Bigg(b_{1k}\sin(\lambda_{1k}x)\frac{\sinh(\tilde\lambda_{1k}(b-y))}{\sinh(\tilde\lambda_{1k}b)} + b_{2k}\sin(\lambda_{2k}x)\frac{\sinh(\tilde\lambda_{2k}y)}{\sinh(\tilde\lambda_{2k}b)} \\ &+ b_{3k}\sin(\lambda_{3k}y)\frac{\sinh(\tilde\lambda_{3k}(a-x))}{\sinh(\tilde\lambda_{3k}a)} + b_{4k}\sin(\lambda_{4k}y)\frac{\sinh(\tilde\lambda_{4k}x)}{\sinh(\tilde\lambda_{4k}a)}\Bigg) \end{aligned} \tag{2.7}$$

where $\tilde\lambda_{ik}^2-\lambda_{ik}^2=1$, then $f_1=f-f_{edge}$ vanishes at the boundaries of the rectangle and $u_{edge}=f_{edge}$.

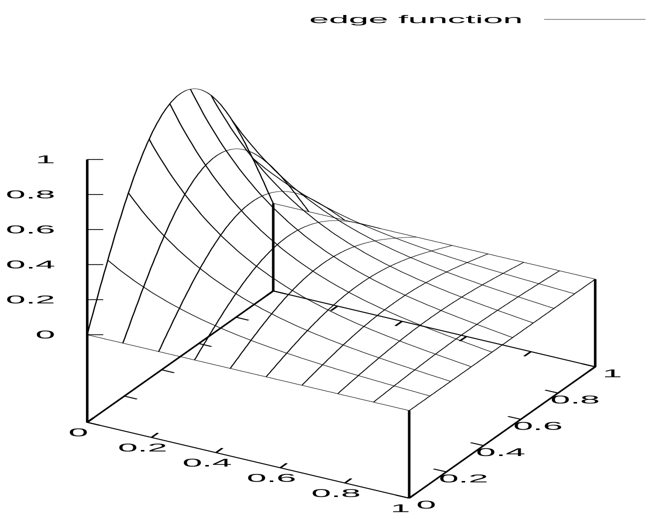

Figure 1: The form of the edge function at edge $x=0$.

If u_1 is a solution of Eq. (2.1) with the right hand side f_1, then u_1+f_{edge} is a solution of (2.1) with the same original right hand side f.

The convergence of series (2.5) can be slow due to the Gibbs phenomenon if the boundary functions f_{edge} do not vanish in the corners together with some of its even derivatives.

However the boundary functions can also be presented as a sum of some corner function and a function which vanishes at the corners together with some first even derivatives. If in the original boundary conditions (2.4)

$$\varphi_1(0) = \varphi_3(0) = A_1, \quad \varphi_1(a) = \varphi_4(0) = A_2,$$

$$\varphi_2(0) = \varphi_3(b) = A_3, \quad \varphi_2(a) = \varphi_4((b) = A_4,$$

then, for instance, after the subtraction of the function

$$f_{corner}(x,y) = A_1 \frac{\sinh(a-x)}{\sinh(a)} \frac{\sinh(b-y)}{\sinh(b)} + A_2 \frac{\sinh(x)}{\sinh(a)} \frac{\sinh(b-y)}{\sinh(b)}$$

$$+A_3 \frac{\sinh(a-x)}{\sinh(a)} \frac{\sinh(y)}{\sinh(b)} + A_4 \frac{\sinh(x)}{\sinh(a)} \frac{\sinh(y)}{\sinh(b)} \tag{2.8}$$

(see the function of type f_{corner} in Fig. 2 for corner (1,1)) the right hand side vanishes at the corners. Function $0.5 \cdot f_{corner}$ should be added to the solution.

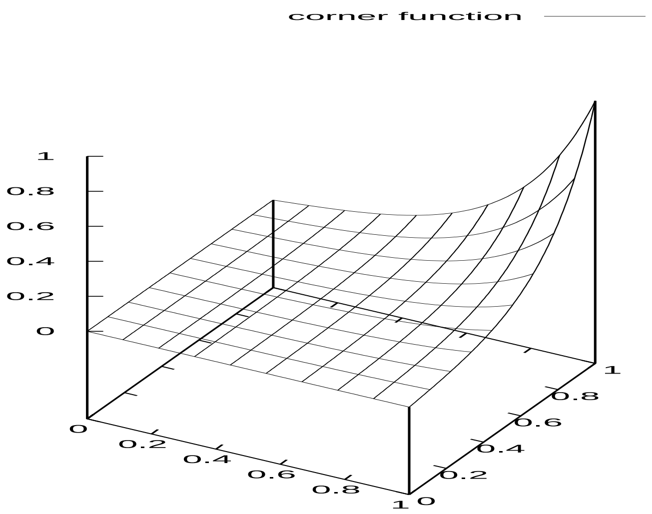

Figure 2: The form of the corner function in the corner (1,1).

Similarly, the second derivatives at the corners are subtracted. Let

$$\frac{\partial^2 \varphi_1}{\partial x^2}(0) = B_1.$$

For example, the function

$$f_{2^{nd}corner} = B_1 \left(\frac{\sinh(\lambda_1(a-x))}{\sinh(\lambda_1 a)} \frac{\sinh(\lambda_1(b-y))}{\sinh(\lambda_1 b)} \right.$$

$$\left. - \frac{\sinh(\lambda_2(a-x))}{\sinh(\lambda_2 a)} \frac{\sinh(\lambda_2(b-y))}{\sinh(\lambda_2 b)} \right), \tag{2.9}$$

with $\lambda_1^2 - \lambda_2^2 = 1$, has the same second derivative at the left bottom corner as the original function. Then the partial solution is

$$u_{2^{nd}corner} = \frac{B_1}{\lambda_1^2 + \lambda_2^2} \left(\frac{\sinh(\lambda_1(a-x))}{2\lambda_1^2 \sinh(\lambda_1 a)} \frac{\sinh(\lambda_1(b-y))}{\sinh(\lambda_1 b)} \right.$$

$$\left. - \frac{\sinh(\lambda_2(a-x))}{2\lambda_2^2 \sinh(\lambda_2 a)} \frac{\sinh(\lambda_2(b-y))}{\sinh(\lambda_2 b)} \right). \tag{2.10}$$

The combination of four similar functions $f_{2^{nd}corner}$ produces the appropriate values at all the corners. This algorithm has an $O(\frac{1}{N^4})$ accuracy if it is combined with an appropriate method for the solution of the Laplace equation with specified boundary conditions which agrees with [15].

We can continue this procedure. Fourth derivatives can be subtracted in the corners which is a linear combination of $f_{2^{nd}corner}$ function and a corresponding partial solution can be subtracted from the solution. We can also subtract a function from the right hand side (and, respectively, the solution), such that the remaining part vanishes together with its second derivatives at the corners. Specifically, if

$$\frac{\partial^2 f_1}{\partial y^2}(x,0) = \sum_{k=1}^{\infty} c_{1k} \sin(\lambda_{1k} x), \quad \frac{\partial^2 f_2}{\partial y^2}(x,b) = \sum_{k=1}^{\infty} c_{2k} \sin(\lambda_{2k} x) \tag{2.11}$$

and

$$\frac{\partial^2 f_3}{\partial x^2}(0,y) = \sum_{k=1}^{\infty} c_{3k} \sin(\lambda_{3k} y), \frac{\partial^2 f_4}{\partial x^2}(a,y) = \sum_{k=1}^{\infty} c_{4k} \sin(\lambda_{4k} y) \tag{2.12}$$

then the following function subtracts the second derivatives at the edges (which is the difference of two functions of the form depicted in Fig. 1)

$$\begin{aligned} f_{2^{nd}edge}(x,y) &= \sum_{k=1}^{\infty} \frac{c_{1k}}{\mu_{1k}^2 - \nu_{1k}^2} \sin(\lambda_{1k} x) \left(\frac{\sinh(\mu_{1k}(b-y))}{\sinh(\mu_{1k} b)} - \frac{\sinh(\nu_{1k}(b-y))}{\sinh(\nu_{1k} b)} \right) \\ &+ \sum_{k=1}^{\infty} \frac{c_{2k}}{\mu_{2k}^2 - \nu_{2k}^2} \sin(\lambda_{2k} x) \left(\frac{\sinh(\mu_{2k} y)}{\sinh(\mu_{2k} b)} - \frac{\sinh(\nu_{2k} y)}{\sinh(\nu_{2k} b)} \right) \\ &+ \sum_{k=1}^{\infty} \frac{c_{3k}}{\mu_{3k}^2 - \nu_{3k}^2} \sin(\lambda_{3k} y) \left(\frac{\sinh(\mu_{3k}(a-x))}{\sinh(\mu_{3k} a)} - \frac{\sinh(\nu_{3k}(a-x))}{\sinh(\nu_{3k} a)} \right) \\ &+ \sum_{k=1}^{\infty} \frac{c_{4k}}{\mu_{4k}^2 - \nu_{4k}^2} \sin(\lambda_{4k} y) \left(\frac{\sinh(\mu_{4k} x)}{\sinh(\mu_{4k} a)} - \frac{\sinh(\nu_{4k} x)}{\sinh(\nu_{4k} a)} \right) \end{aligned} \tag{2.13}$$

where

$$\mu_{ik}^2 - \nu_{ik}^2 - \lambda_{ik}^2 = 1.$$

The corresponding partial solution $u_{2^{nd}edge}$ of Eq. (2.1) with $f - f_{edge} - f_{2^{nd}edge}$ as the right hand side has the following form

$$u_{2^{nd}edge}(x,y) = \sum_{k=1}^{\infty} \frac{c_{1k}}{\mu_{1k}^2 - \nu_{1k}^2} \sin(\lambda_{1k} x) \left(\frac{\sinh(\mu_{1k}(b-y))}{(\mu_{1k}^2 - \lambda_{1k}^2) \sinh(\mu_{1k} b)} - \frac{\sinh(\nu_{1k}(b-y))}{(\nu_{1k}^2 - \lambda_{1k}^2) \sinh(\nu_{1k} b)} \right)$$

$$+\sum_{k=1}^{\infty}\frac{c_{2k}}{\mu_{2k}^2-\nu_{2k}^2}\sin(\lambda_{2k}x)\left(\frac{\sinh(\mu_{2k}y)}{(\mu_{2k}^2-\lambda_{2k}^2)\sinh(\mu_{2k}b)}-\frac{\sinh(\nu_{2k}y)}{(\nu_{2k}^2-\lambda_{2k}^2)\sinh(\nu_{2k}b)}\right)$$
$$+\sum_{k=1}^{\infty}\frac{c_{3k}}{\mu_{3k}^2-\nu_{3k}^2}\sin(\lambda_{3k}y)\left(\frac{\sinh(\mu_{3k}(a-x))}{(\mu_{3k}^2-\lambda_{3k}^2)\sinh(\mu_{3k}a)}-\frac{\sinh(\nu_{3k}(a-x))}{(\nu_{3k}^2-\lambda_{3k}^2)\sinh(\nu_{3k}a)}\right)$$
$$+\sum_{k=1}^{\infty}\frac{c_{4k}}{\mu_{4k}^2-\nu_{4k}^2}\sin(\lambda_{4k}y)\left(\frac{\sinh(\mu_{4k}x)}{(\mu_{4k}^2-\lambda_{4k}^2)\sinh(\mu_{4k}a)}-\frac{\sinh(\nu_{4k}x)}{(\nu_{4k}^2-\lambda_{4k}^2)\sinh(\nu_{4k}a)}\right). \tag{2.14}$$

However each "edge" step takes $O(N^2 \log N)$ operations compared to $O(N^2)$ for each corner subtraction. In the iterative procedure and all numerical examples we apply one edge subtractional step and one (values of RHS in the corners are subtracted only) or two (values and the second derivatives in the corners of RHS) subtractional steps in the corners.

Finally, the fourth order algorithm can be summarized as follows:

1. The function f_{corner} (2.8) is computed and subtracted from RHS; at the same time the function $\frac{1}{2}f_{corner}$ is added to the solution.

2. For the remaining part of RHS the function $f_{2^{nd}corner}$ (2.11) is subtracted from RHS and $u_{2^{nd}corner}$ is added to the solution.

3. The egdes of RHS (2.4) are expanded into sine series (2.5), the function f_{edge} (2.7) is subtracted from the right hand side and added to the solution.

4. The rest of RHS is expanded into the sine series (2.2) and u in (2.3) is added to the solution, which completes the algorithm.

It is to be emphasized that here we assume smooth boundary conditions (the values at the corners are matched and the Laplacian in the corners coincides with the right hand side). The removal of singularities in singular cases by the subtraction technique was discussed in [1, 2, 24] for 2-D and in [6] for 3-D problems.

2.3 Helmholtz equation and Modified Helmholtz equation

A similar algorithm is applicable to the Helmholtz equations. The same expansion for the right hand side is valid, the only difference is the integration results.

Consider the Helmholtz equation (HE)

$$\Delta u(x,y) + k^2 u(x,y) = f(x,y) \tag{2.15}$$

and the modified Helmholtz equation (MHE)

$$\Delta u(x,y) - \lambda^2 u(x,y) = f(x,y) \tag{2.16}$$

If the right hand side is expanded into Fourier series then the solution is

$$u(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{a_{mn}}{-\pi^2\left(\frac{n^2}{a^2}+\frac{m^2}{b^2}\right)+k^2}\sin\left(\frac{n\pi x}{a}\right)\sin\left(\frac{m\pi y}{b}\right) \tag{2.17}$$

for the HE (here we assume $k^2 \neq \pi^2(n^2/a^2 + m^2/b^2)$ for all $n, k = 1, 2, \ldots$) and

$$u(x, y) = -\sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{a_{mn}}{\pi^2\left(\frac{n^2}{a^2} + \frac{m^2}{b^2}\right) + \lambda^2} \sin\left(\frac{n\pi x}{a}\right) \sin\left(\frac{m\pi y}{b}\right) \tag{2.18}$$

for the MHE.

The same subtraction steps, as were used for the Poisson equation, can be implemented for the Helmholtz equations. If f_{edge} is defined by (2.7) and $\tilde{\lambda}_{ij}$ are chosen such that

$$\tilde{\lambda}_{ij}^2 - \lambda_{ij}^2 - \lambda^2 = 1$$

for the MHE then the same edge function f_{edge} defined by (2.7), is the corresponding part of the solution. For HE we choose some $r > k$ and let

$$\tilde{\lambda}_{ij}^2 = \sqrt{\lambda_{ij}^2 + r^2 - k^2} \quad r > k,$$

then

$$u_{edge} = \frac{1}{r^2} f_{edge}.$$

Corner functions are the same for the HE as in (2.8). The corresponding part of the solution is

$$u_{corner} = \frac{1}{2 + k^2} f_{corner}.$$

For the MHE we can choose, for instance,

$$f_{corner}(x, y) = A_1 \frac{\sinh(\lambda(a - x))}{\sinh(\lambda a)} \frac{\sinh(b - y)}{\sinh(b)} + A_2 \frac{\sinh(\lambda x)}{\sinh(\lambda a)} \frac{\sinh(b - y)}{\sinh(b)}$$

$$+A_3 \frac{\sinh(\lambda(a - x))}{\sinh(\lambda a)} \frac{\sinh(y)}{\sinh(b)} + A_4 \frac{\sinh(\lambda x)}{\sinh(\lambda a)} \frac{\sinh(y)}{\sinh(b)} = u_{corner}(x, y). \tag{2.19}$$

Similarly, the subtraction function (2.9), for the second derivatives in the corners, is appropriate for the Helmholtz equation. The corresponding parts of the solution for the HE are

$$B_1 \left(\frac{1}{2(\lambda_1^2 + k^2)} \frac{\sinh(\lambda_1(a - x))}{\sinh(\lambda_1 a)} \frac{\sinh(\lambda_1(b - y))}{\sinh(\lambda_1 b)} \right.$$
$$\left. - \frac{1}{2(\lambda_2^2 + k^2)} \frac{\sinh(\lambda_2(a - x))}{\sinh(\lambda_2 a)} \frac{\sinh(\lambda_2(b - y))}{\sinh(\lambda_2 b)} \right), \tag{2.20}$$

and for the MHE are

$$B_1 \left(\frac{1}{2(\lambda_1^2 - \lambda^2)} \frac{\sinh(\lambda_1(a - x))}{\sinh(\lambda_1 a)} \frac{\sinh(\lambda_1(b - y))}{\sinh(\lambda_1 b)} \right.$$
$$\left. - \frac{1}{2(\lambda_2^2 - \lambda^2)} \frac{\sinh(\lambda_2(a - x))}{\sinh(\lambda_2 a)} \frac{\sinh(\lambda_2(b - y))}{\sinh(\lambda_2 b)} \right). \tag{2.21}$$

In the latter case, λ_1 and λ_2 are chosen to exceed λ.

3 Numerical results

3.1 Subtractional Poisson solver

In Examples 1-6 we test our algorithm for various number of subtraction steps. If only corner functions f_{corner} and edge function f_{edge} are subtracted (one subtraction step), then the expected convergence of ε_{MAX} is $O(1/N^2)$ [15], where N is the number of collocation points in each direction. If functions $f_{2^{nd}corner}$ are subtracted as well (two subtraction steps), then the expected convergence is $O(1/N^4)$.

Example 1. We solve the Poisson equation with boundary conditions corresponding to the RHS

$$f(x,y) = 4\alpha\left[\alpha\left((x-x_0)^2+(y-y_0)^2\right)-1\right]\exp\left\{-\alpha\left((x-x_0)^2+(y-y_0)^2\right)\right\}, \qquad (3.22)$$

where $x_0 = 0.5,\ y_0 = 0.5,\ \alpha = 1$ in the domain $[0,1]\times[0,1]$ (see Table 2).

$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
8×8	1.8e-5	1.2e-5	1.4e-5
16×16	1.2e-6	8.0e-6	9.5e-7
32×32	7.9e-8	5.2e-8	6.1e-8
64×64	5.0e-9	3.3e-9	3.9e-9
128×128	3.2e-10	2.1e-10	2.5e-10

Table 2: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.1). The RHS is (3.22) with $x_0 = y_0 = 0.5$, $\alpha = 1$.

Figure 3 demonstrates the form of the right hand side after the subtraction of the "corner functions" which corresponds to $f - f_{corner} - f_{2^{nd}corner}$ for the RHS defined in (3.22) and centered in the middle of the square domain. Figure 4 illustrates the right hand side after all the subtraction steps $f - f_{corner} - f_{2^{nd}corner} - f_{edge}$ for the RHS defined in (3.22), with $x_0 = 0.5,\ y_0 = 0.5,\ \alpha = 1$.

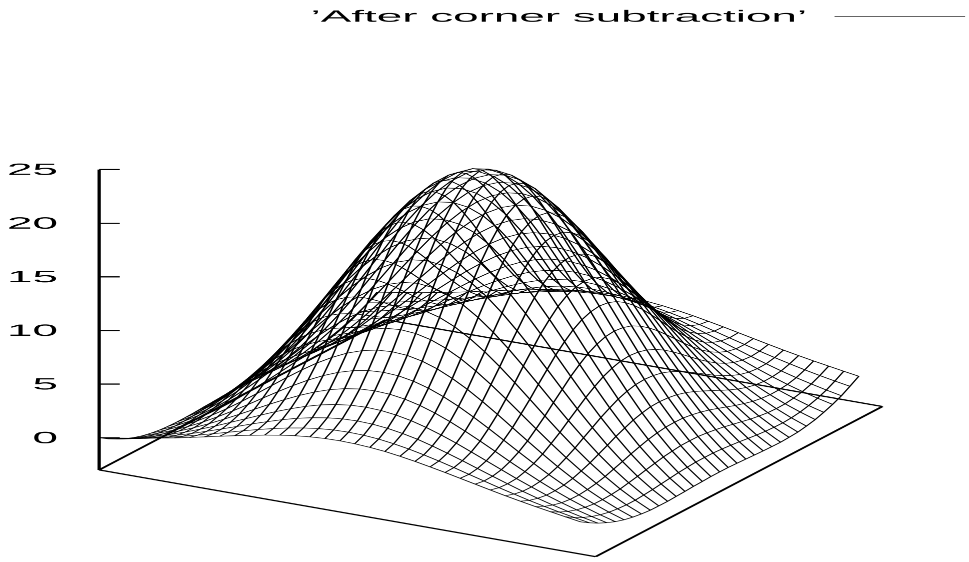

Figure 3: The form of the right hand side after subtraction of the corner functions $f - f_{corner} - f_{2^{nd}corner}$ for the RHS defined in (3.22) with $x_0 = y_0 = 0.5$, $\alpha = 1$.

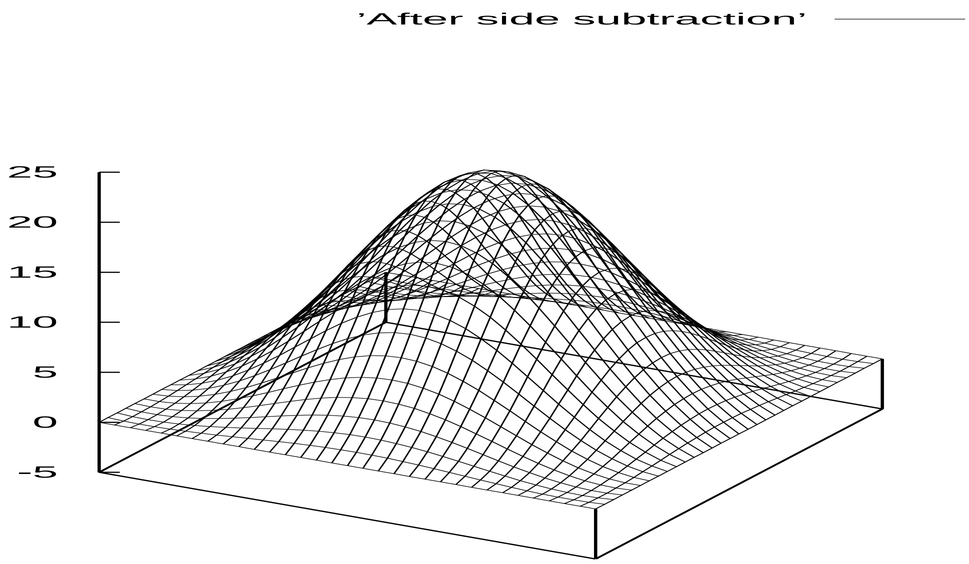

Figure 4: The form of the right hand side after the subtraction of the corner and the side functions $f - f_{corner} - f_{2^{nd}corner} - f_{edge}$ for the RHS defined in (3.22) with $x_0 = y_0 = 0.5$, $\alpha = 1$.

We observe that in Fig. 3 the remaining part of the right hand side vanishes in the corners only, while in Fig. 4 it vanishes at all the boundary of the solution domain.

Example 2. Consider the Poisson equation with the RHS

$$f(x,y) = -5\cos x\cos(2y) \tag{3.23}$$

Table 3 presents the maximal error without subtraction, in the case when only f_{corner} is subtracted, for $f - f_{corner} - f_{edge}$ and, finally, for all subtraction steps $f - f_{corner} - f_{edge} - f_{2^{nd}corner}$.

$N_x \times N_y$	Description of subtraction steps			
	none	$f - f_{corner}$	$f - f_{corner} - f_{edge}$	all steps
16×16	1.5e-3	5.2e-4	1.4e-5	2.8e-6
32×32	4.4e-4	1.4e-4	3.0e-6	1.8e-7
64×64	1.2e-4	3.8e-5	7.3e-7	1.1e-8
128×128	3.1e-5	9.8e-6	1.8e-7	7.2e-10

Table 3: Maximal errors for the solution of the Poisson equation (2.1) and the RHS (3.23).

Example 3. Consider the Poisson equation with an exact solution $u(x,y) = \dfrac{1}{x^2+y^2+2}$ which corresponds to the RHS

$$f(x,y) = -\frac{8-4x^2-4y^2}{(x^2+y^2+2)^3} \tag{3.24}$$

Table 4 presents the maximal error without subtraction, for the subtractional scheme $f - f_{corner} - f_{edge}$ and, finally, for all subtraction steps $f - f_{corner} - f_{edge} - f_{2^{nd}corner}$.

$N_x \times N_y$	Description of subtraction steps		
	none	$f - f_{corner} - f_{edge}$	all steps
16×16	1.0e-4	6.4e-6	6.1e-7
32×32	2.7e-5	1.5e-6	4.4e-8
64×64	6.9e-6	3.6e-7	2.9e-9
128×128	1.7e-6	9.0e-8	1.8e-10

Table 4: Maximal errors for the solution of the Poisson equation (2.1) and the RHS (3.24).

Example 4. For our next example we consider the Poisson equation with the RHS including a steep Gaussian function (3.22), where $x_0 = 0.5$, $y_0 = 0.5$, $\alpha = 200, 400$ in the domain $[0,1] \times [0,1]$ (such examples were considered in [2, 17]). Table 5 presents a comparison of the present method and the adaptive Poisson solver presented in [17]. The relative errors are compared to the eighth order method ($K = 8$) and the sixteen order method. Certainly Table 5 does not demonstrate the superiority of the present forth order method method when compared to high order FMM method incorporated with Chebyshev polynomial expansions in subdomains, but only demonstrates an extremely high performance of the eigenfunction

(Fourier) method in the case when the RHS nearly vanishes at the boundaries. The RHS is a function (3.22) with $\alpha = 400$.

number of grid points	relative error $\varepsilon_{\mathcal{L}^2}$		
	present method	[17] with $K = 8$	[17] with $K = 16$
256	1.0	1.2e-1	2.31
1024	2.5e-4	5.4e-3	5.1e-4
4096	2.6e-13	4.9e-5	7.9e-7
16384	7.4e-16	1.8e-7	4.7e-11

Table 5: Relative errors for the solution of Eq. (2.1), with the RHS including a steep Gaussian function (3.22), with $\alpha = 400$: comparison of the present method and the method presented in [17], the eigth order method ($K = 8$) and the sixteen order method ($K = 16$).

Table 6 presents a comparison of maximal errors for the present method and the Poisson solver which used extension and folding of the original domain (see [2]). The RHS is (3.22) with $\alpha = 200$. It is to be emphasized that the present method uses a smaller number of points than the previous algorithm [2] which employed the Fourier transform in the extended domain.

number of grid points	maximal error ε_{MAX}	
	present method	method of [2]
$16^2 = 256$	2.1e-2	2.1e-2
$32^2 = 1024$	1.8e-7	1.7e-7
$64^2 = 4096$	4.4e-16	6.0e-15

Table 6: Maximal errors for the solution of Eq. (2.1), with the RHS including a steep Gaussian function (3.22), $\alpha = 200$: comparison of the present method and the method presented in [2]. The number of grid points in [2] is given without taking into account extension points.

Example 5. Consider the case (see examples with the same right hand side in [17, 2]) when the RHS is a sum

$$f(x, y) = \sum_{i=1}^{14} 4\alpha_i \left(\alpha_i r_i^2 - 1\right) e^{-\alpha r_i^2}, \tag{3.25}$$

where $r_i^2 = (x - x_i)^2 + (y - y_i)^2$, the centers (x_i, y_i) are randomly located in the box $[0.1, 0.9]^2$ and $\alpha_i \in [1024, 16384]$. The results are presented in Table 7. It is to be noted that (3.25) and (3.22) as in the previous example were chosen as benchmark problems in [17].

$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
64×64	2.6	3.6e-1	5.2
128×128	5.1e-2	1.7e-3	2.4e-2
256×256	3.0e-6	4.2e-8	6.0e-7
512×512	1.1e-15	1.5e-16	2.2e-15

Table 7: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.1). The RHS is a sum (3.25), where $1024 \leq \alpha_i \leq 16384$.

Example 6. Finally, consider the case when the exact solution of equation (2.1) is both steep and oscillating

$$u = e^{-\alpha_i r^2} \cos(c(x+y-1)), x, y \in [0,1] \times [0,1]$$

where $r^2 = (x-0.5)^2 + (y-0.5)^2$, $\alpha = 30, c = 20$. This corresponds to the RHS

$$f(x,y) = \Big[4\alpha\left(\alpha r^2 - 1\right)\cos(c(x+y-1)) - 2c^2\cos(c(x+y-1))$$

$$+4c\alpha(x+y-1)\sin(c(x+y-1))\Big]\, e^{-\alpha r^2} \tag{3.26}$$

Similar examples for the Helmholtz equation and the Poisson equation in a disc were considered in [12, 9]. The results are presented in Table 8.

$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
16×16	1.3e-4	2.0e-5	1.3e-4
32×32	3.2e-6	7.3e-7	4.7e-6
64×64	1.8e-7	2.3e-8	1.4e-7
128×128	1.4e-8	1.6e-9	9.7e-9
256×256	9.1e-10	1.0e-10	6.2e-10
512×512	5.9e-11	6.3e-12	3.9e-11

Table 8: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.1). The RHS is (3.26).

As can be observed in Examples 1-6, the error decays at least as $1/N^4$ for two subtraction steps algorithm (f_{corner}, $f_{2^{nd}corner}$ and f_{edge} are subtracted).

3.2 Subtractional solver for the modified Helmholtz equation

The examples below illustrate the convergence of the subtractional solver for the modified Helmholtz equation.

Example 7. Consider the case when the RHS is

$$-4\alpha\left[\alpha\left((x-0.5)^2+(y-0.5)^2\right)-1-\lambda^2\right]\exp\left\{-\alpha\left((x-0.5)^2+(y-0.5)^2\right)\right\}, \tag{3.27}$$

$x, y \in [0,1]$, with $\alpha = 1$. The results are presented in Table 9.

$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
16×16	1.2e-6	7.6e-7	9.0e-7
32×32	8.0e-8	4.9e-8	5.8e-8
64×64	5.1e-9	3.1e-9	3.7e-9
128×128	3.2e-10	2.0e-10	2.3e-10

Table 9: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of modified Helmholtz equation (2.16) and the RHS (3.27) with $\alpha = \lambda = 1$.

Table 10 presents the dependence of the errors on λ^2 in (2.16) for 64×64 collocation points and the RHS defined in (3.27).

λ^2	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
1	5.1e-9	3.1e-9	3.7e-9
4	1.2e-8	6.6e-9	7.7e-9
16	8.5e-8	3.8e-8	4.4e-8
64	9.8e-7	3.6e-7	4.2e-7
100	3.4e-6	7.7e-7	9.0e-7
400	2.5e-5	1.4e-5	1.6e-5

Table 10: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.16) with various λ and 64×64 grid points and and the RHS (3.27) with $\alpha = \lambda = 1$.

Example 8. Consider the case when the exact solution of modified Helmholtz equation (2.16) involves a steep Gaussian function (3.27), where $\alpha = 200, 400,\ \lambda = 1$. The results are presented in Table 11.

$N_x \times N_y$	α	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
200	16×16	2.1e-2e-4	1.4e-3	1.7e-2
	32×32	1.8e-7	1.7e-8	1.9e-7
	64×64	4.4e-16	3.6e-17	4.1e-16
400	16×16	4.7e-1	6.2e-2	9.6e-1
	32×32	2.7e-4	1.5e-5	2.5e-4
	64×64	2.1e-13	1.6e-14	2.6e-13
	128×128	5.6e-16	3.5e-17	5.6e-16

Table 11: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.16) with the RHS (3.27), where $\alpha = 200, 400,\ \lambda = 1$.

Example 9. Consider the case when the exact solution of equation (2.16) is both steep and oscillating

$$u = e^{-\alpha_i r^2} \cos(c(x + y - 1)),\ \ x, y \in [0, 1] \times [0, 1]$$

where $r^2 = (x - 0.5)^2 + (y - 0.5)^2$, $\alpha = 200, c = 100$. This corresponds to the RHS

$$f(x, y) = \Big[4\alpha \left(\alpha r^2 - 1\right) \cos(c(x + y - 1)) - 2c^2 \cos(c(x + y - 1))$$

$$+4c\alpha(x+y-1)\sin(c(x+y-1)) - \lambda^2\cos(c(x+y-1))\Big]\, e^{-\alpha r^2} \tag{3.28}$$

The results are presented in Table 12.

$N_x \times N_y$	ε_{MAX}	ε_{MSQ}	$\varepsilon_{\mathcal{L}^2}$
32×32	5.0e-1e-6	3.2e-2	3.7e-1
64×64	3.0e-8	2.6e-9	4.2e-8
128×128	2.1e-15	5.2e-16	8.3e-15

Table 12: MAX, MSQ and $\mathcal{L}^2$ errors for the solution of Eq. (2.16) with the RHS (3.28), $\alpha = 200,\ c = 100,\ \lambda = 1$.

4 Solution of elliptic equations with variable coefficients

4.1 Statement of the problem

We solve the Dirichlet problem for elliptic equation with variable coefficients

$$\mathcal{L}u = f, \quad (x, y) \in \Omega \tag{4.1}$$

$$u(x, y) = \phi(x, y), \quad (x, y) \in \partial\Omega \tag{4.2}$$

in a rectangular domain, where

$$(\mathcal{L}u)(x,y) = \frac{\partial}{\partial x}\left(a_1(x,y)\frac{\partial u}{\partial x}(x,y)\right) + \frac{\partial}{\partial y}\left(a_2(x,y)\frac{\partial u}{\partial y}(x,y)\right) \tag{4.3}$$

with the satisfied ellipticity condition

$$0 < a_i^{\min} \le a_i(x,y) \le a_i^{\max}. \tag{4.4}$$

After an appropriate transformation any elliptic operator can be presented as (4.3).

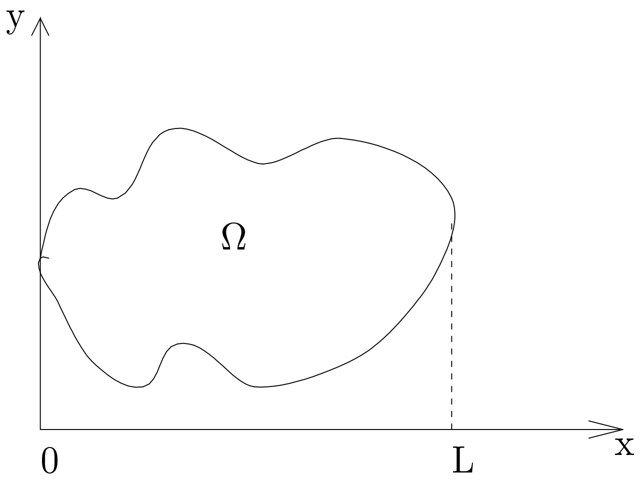

Figure 5: The Poisson equation is solved in a curvilinear domain

In addition, we solve the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \quad (x,y) \in \Omega, \tag{4.5}$$

in a curvilinear domain Ω (see. Fig.5). If this domain is mapped into a square region $\omega = \{0 \le \xi \le 1, 0 \le \eta \le 1\}$ with an appropriate orthogonal transformation then Eq. (4.5) is transformed into an elliptic equation with variable coefficients.

We will refer to the fast solver for the Poisson equation with constant coefficients λ_i $i = 1,2$ (see Section 2):

$$\mathcal{L}_0 u \equiv \frac{\partial}{\partial x}\left(\lambda_1 \frac{\partial u}{\partial x}(x,y)\right) + \frac{\partial}{\partial y}\left(\lambda_2 \frac{\partial u}{\partial y}(x,y)\right) \tag{4.6}$$

as the "basic (kernel) solver".

4.2 Solution by the iterative method

In order to solve Eq. (4.1) we will perform the Richardson preconditioned iterations

$$\mathcal{L}_0 u^{n+1} = \mathcal{L}_0 u^n + f - \mathcal{L} u^n \tag{4.7}$$

which will be referred as the *Defect Correction* (DC) iterations. If we denote by u the exact solution of (4.1) and $r^n = u^n - u$ then (4.7) can be rewritten as

$$\mathcal{L}_0 r^{n+1} = (\mathcal{L}_0 - \mathcal{L}) r^n \tag{4.8}$$

The following claim justifies the choice of λ_i in Eq. (4.6).

Proposition. Let $a_i^{\min}$ and $a_i^{\max}$ in (4.4) be the minimal and the maximal values of $a_i \in \Omega$, respectively. If we choose

$$\lambda_i = 0.5(a_i^{\min} + a_i^{\max}) \tag{4.9}$$

then (4.7) converges in the energetic norm

$$||g||_{\bar{a}}^2 \equiv \int_\Omega \left(a_1 g_x^2 + a_2 g_y^2\right) \, dx \, dy$$

defined for $g = 0$ on the boundary of Ω, with the rate of convergence

$$q = \max\left\{ \frac{{a_1}^{\max} - {a_1}^{\min}}{{a_1}^{\max} + {a_1}^{\min}}, \frac{{a_2}^{\max} - {a_2}^{\min}}{{a_2}^{\max} + {a_2}^{\min}} \right\}.$$

Proof. Multiplying (4.8) by r^{n+1} and integrating over Ω we have:

$$\int_\Omega \left(\lambda_1 (r^{n+1})_{x^2} + \lambda_2 (r^{n+1})_{y^2}\right) r^{n+1} \, dx \, dy$$

$$= \int_\Omega \left[(\lambda_1 r^n)_{x^2} - (a_1 r_x^n)_x r^{n+1} + (\lambda_2 r^n)_{y^2} - \left(a_2 r_y^n\right)_y r^{n+1}\right] \, dx \, dy.$$

When $r^n = 0$ on the boundary of Ω for all n, the integration by parts gives:

$$\sum_{i=1}^{2}\int_{\Omega}\lambda_i(r^{n+1}_{x_i})^2 \le \sum_{i=1}^{2}\int_{\Omega}|\lambda_i - a_i||r^n_{x_i}||r^{n+1}_{x_i}|. \tag{4.10}$$

The latter inequality (together with the Cauchy-Schwartz inequality) yields

$$||r^{n+1}||^2_{\lambda} \le ||r^n||_{\lambda-a}||r^{n+1}||_{\lambda-a}.$$

If $|\lambda_i - a_i| = |0.5 \cdot {a_i}^{\max} + 0.5 \cdot {a_i}^{\min} - a_i| \le q_i\lambda_i$, then

$$||r^{n+1}||_{\lambda} \le q||r^{n+1}||_{\lambda}.$$

Thus, the solution of an elliptic equation with variable coefficients is reduced to an iterative procedure which uses the fast Poisson solver (which is described in Section 2) with constant coefficients as the base (kernel) solver. ■

We will now slightly change the iterative procedure (Eq. (4.7)) by introducing a local relaxation parameter τ:

$$\mathcal{L}_0 u^{n+1} = \mathcal{L}_0 u^n + \tau(f - \mathcal{L}u^n) \tag{4.11}$$

It is clear that if the iterations defined in (4.11), which we refer as the Defect Correction Weighted (DCW) iterations, converge then the iterations in (4.11) converge to the solution of Eq. (4.7). Now we will discuss a subclass of solutions where a simple choice of τ can essentially increase the rate of convergence. To this end let us consider the case $a_1(x,y) = a_2(x,y) \equiv a(x,y)$ and consequently $\lambda_1 = \lambda_2 = \lambda$. In fact we only need $a_1(x_1,x_2) \sim a_2(x_1,x_2)$.

In order to illustrate how to choose the parameter τ we first assume that

$$\tau = \begin{cases} 2, & \lambda/a(x_1,x_2) \ge 2 \\ 0.5, & \lambda/a(x_1,x_2) < \frac{3}{4} \\ 1, & \text{otherwise.} \end{cases} \tag{4.12}$$

Then, following the above proof, we observe that the "local" rate of convergence is essentially improved in the regions where $\tau = .5$ or $\tau = 2$. A penalty is paid in regions where there is a transition from one value of τ to another (this penalty obviously remains if τ is smoothed by multiplying it by an appropriate smooth function).

Alternatively, τ can be chosen as $\tau = \frac{\lambda}{a(x,y)}$. In this case, the method will be referred as DCWA and inequality (4.10) becomes

$$\int_{\Omega}\lambda\left[(r^{n+1}_x)^2 + (r^{n+1}_y)^2\right]\,dx\,dy \le \int_{\Omega}a(x,y)\left(|r^n_x||\tau_x| + |r^n_y||\tau_y|\right)|r^{n+1}|$$

$$= \int_{\Omega}a(x,y)^{-1}\left(|r^n_x||a_x| + |r^n_y||a_y|\right)|r^{n+1}|. \tag{4.13}$$

As r^{n+1} vanishes on the boundary of Ω then by the Poincaré inequality we obtain that the right hand side of the estimate in Eq. (4.13) is bounded by $C||r^n||_{\lambda}||r^{n+1}||_{\lambda}$, where the constant C depends on the size of Ω and the function a. We are interested in cases where C is small. It takes place when $|\nabla(a)a^{-1}r^{n+1}|$ is small in comparison to the gradient $|\nabla(r^{n+1})|$. Thus, the iteration process (4.11) converges if an upper bound of $|\nabla aa^{-1}|$ is efficiently bounded at least in regions which are non-adjacent to the boundary $\partial\Omega$. Near the boundary $|r^{n+1}|$ can be estimated by a distance from the boundary multiplied by the mean value of $|r_x^{n+1}|$ or $|r_y^{n+1}|$ along the path leading to a nearby boundary point. If this condition is only partially satisfied then it is possible to combine two choices of τ by taking the bounded value of τ for "small" a: when $a(x,y) < 0.5 \cdot \lambda$ then $\tau = 2$. This approach is referred as the DCWB method.

Case 1A and 1B and Example 2 in Section 4.3 employ the DCWA and DCWB methods, respectively. The other illustrative applications use the original DC process.

4.3 Numerical results for elliptic equations with non-constant coefficients

The following considerations will be useful for the application of the above Poisson solver (section 2) to the solution of elliptic equations (with non-constant coefficients).

Remarks

1. An additional transformation of variables is required in order to reduce the Poisson equation with constant coefficients to the standard one:

$$x_i = x_i{}'\sqrt{\lambda_i}.$$

   It transforms Eq. (4.1) in the rectangle $[0,1] \times [0,1]$ into a rectangle of the form $[0, 1/(\lambda_1)^{0.5}] \times [0, 1/(\lambda_2)^{0.5}]$. If the ratio of λ_1 and λ_2 (or $\lambda_2 : \lambda_1$) is small, then the accuracy of the basic solver (and thus of the whole method) may deteriorate. We will discuss this with more details in Section 4.3 (examples 4 and 6).

2. For a converging iterative process, the accuracy of the solution is determined only by the approximation of the variable coefficients operator $\mathcal{L}$ with $\mathcal{L}_0$, which is employed as a fast kernel (see Eq. (4.6)). This means that the spectral accuracy can be achieved if there is a spectral approximation for the differential operator $\mathcal{L}$.

3. In principle, it is possible to use other constants than ones in the relation (4.9) for the operator $\mathcal{L}_0$. As far as the corresponding iterative process converges, it converges to the solution of Eq.(4.1). This can be used in order to improve the ratio between λ_1 and λ_2 (see Remark 1 above). For example, it is possible to augment λ_1 and λ_2 by constant values C_1 and C_2 thus improving the robustness of the basic solver. The use of this technique is demonstrated in Examples 4 and 6.

4. In order to evaluate $\mathcal{L}u$ we have to compute derivatives. This is also implemented using a combination of the FFT and the subtraction technique. If $g(x)$ is defined in $[0,\pi]$, then $h(x)$, which is derived as explained above, is subtracted from g such that

$g-h$ vanishes at 0 and π with some of its first even derivatives. Then $g-h$ is expanded (using FFT in practical implementations) into a sine series

$$g(x) - h(x) = \sum_{i=1}^{\infty} b_i \sin(ix),$$

then the derivative g' is computed as

$$g'(x) = h'(x) + \sum_{i=1}^{\infty} i b_i \cos(ix).$$

It is to be noted that the analytic expression for h' is known.

5. In all the following applications the initial solution is attained by simply applying the basic solver (4.6) with the given boundary conditions.

In the following we present a number of applications. Examples 1 and 2 demonstrate the ability of the method to provide spectral accuracy solutions for elliptic equations in a rectangle. Example 3 deals with variable coefficients which are step functions. Solution of such an equation can be considered as a preconditioner to a more complicated case when these constants approximate the values of non-constant coefficients. In Example 4 we solve the Poisson equation with constant coefficients a and b for the ill-posed case $a \gg b$. Examples 5 and 6 in Section 5 indicate that the proposed solution has the potential to solve the Poisson equation in irregular domains.

Example 1: Poisson equation with variable coefficients. We solve the Dirichlet problem for the Poisson equation with the coefficients $a_1(x, y) = a_2(x, y) = \delta + x + y$ in the square $[0, 1] \times [0, 1]$.

Case 1A. The RHS is $6x + 6y + 4\delta$ and the Dirichlet boundary condition corresponds to the exact solution $u = x^2 + y^2$. The results achieved by the DCWA method are compared with those from the DC method. The comparison is displayed in Table 13 when $\delta = 1.0, 0.1$. We observe that the DCWA method has a faster convergence than the DC method. It is to be emphasized that the DCWA method achieved high accuracy after a small number of iterations (see Fig. 6). In order to get full convergence up to several tens of iterations are needed.

		DCWA iterations		DC iterations	
$N_1 \times N_2$	δ	10 iter.	Full convergence	10 iter.	Full convergence
8×8	1.0	3.1E-7	3.8E-8	1.1E-6	3.9E-8
16×16		1.0E-7	4.6E-10	1.0E-6	4.5E-10
32×32		2.6E-8	6.0E-12	1.1E-6	6.0E-12
64×64		6.6E-9	8.0E-13	1.1E-6	4.1E-13
128×128		1.6E-9	1.5E-12	1.1E-6	1.3E-12
8×8	0.1	5.5E-7	2.7E-8	2.5E-4	5.3E-8
32×32		5.5E-8	6.0E-12	4.2E-4	7.0E-12
128×128		7.7E-9	3.0E-12	4.3E-4	5.0E-12

Table 13: L_2 error after 10 iterations. RHS is $6x+6y+4\delta$ and the exact solution is $u = x^2+y^2$.

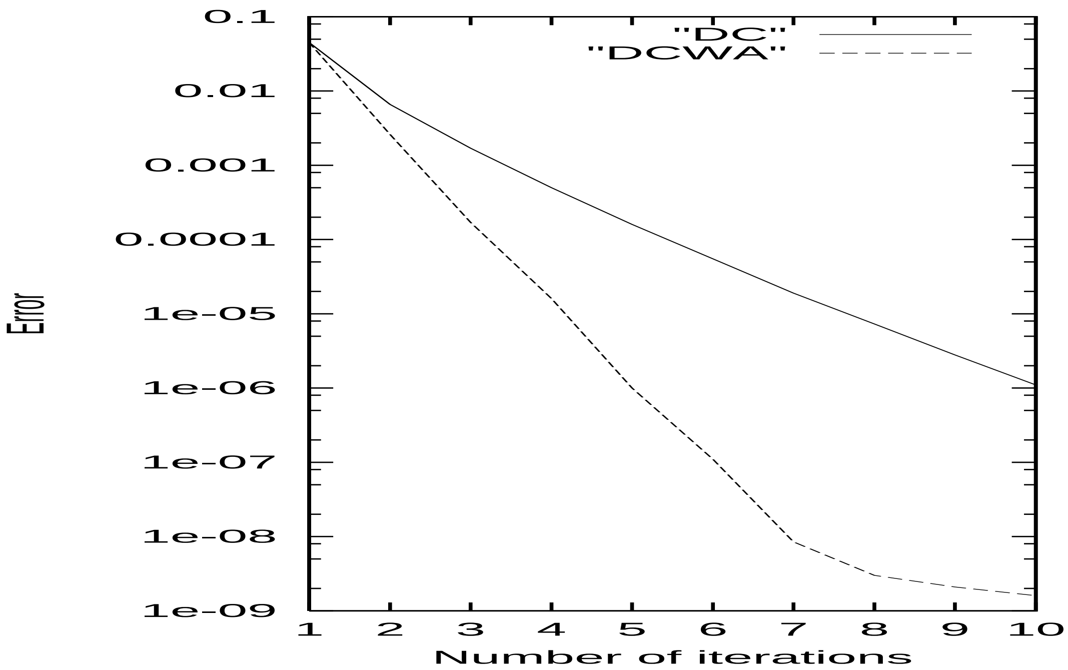

Figure 6: L_2 error for DC and DCWA type iterations

Case 1B. Table 14 displays the results when the RHS is

$$f(x, y) = 2\pi \sin 2\pi(x + y) - 8\pi^2(\delta + x + y) \sin 2\pi x \sin 2\pi y,$$

the exact solution is $u = \sin 2\pi x \sin 2\pi y$ and $\delta = 1$. The results demonstrate faster convergence from the DCWA iterations when compared to the DC method. In the near-degenerated case, $\delta = .01$, more iterations are required (at a given number of collocation points) in order to recover the full accuracy when $\delta = 1$. The results from the DCWA method, given in Table 15, show that a spectral accuracy is achieved after a small number of iterations even for grids with a large number of collocation points.

	DC iterations			DCWA iterations	
	10 iter.	Full convergence		Full convergence	
$N_1 \times N_2$	L_2 error	L_2 error	iter.	L_2 error	iter.
8×8	2.4E-4	2.2E-4	6	2.4E-4	4
16×16	9.6E-6	5.6E-6	13	5.6E-6	6
32×32	5.7E-6	9.0E-8	16	8.2E-8	7
64×64	5.7E-6	7.5E-10	23	7.5E-10	9
128×128	5.7E-6	6.6E-12	27	6.6E-12	11

Table 14: L_2 errors from the DC and the DCWA methods. $\delta = 1$. RHS is $2\pi \sin 2\pi(x+y) - 8\pi^2(1 + x + y) \sin 2\pi x \sin 2\pi y$ and the exact solution is $u = \sin 2\pi x \sin 2\pi y$.

				Full convergence	
$N_1 \times N_2$	10 iter.	15 iter.	20 iter.	L_2 error	No of iter.
8×8	2.1E-4	2.1E-4	2.1E-4	2.1E-4	6
16×16	5.6E-6	5.6E-6	5.6E-6	5.6E-6	8
32×32	2.2E-7	8.5E-8	8.5E-8	8.5E-8	13
64×64	1.8E-7	2.7E-9	8.7E-10	8.0E-10	21
128×128	1.8E-7	3.4E-10	8.2E-11	6.5E-12	35

Table 15: L_2 errors for the DCWA method. RHS is $2\pi \sin 2\pi(x+y) - 8\pi^2(0.01 + x + y)\sin 2\pi x \sin 2\pi y$ and the exact solution is $u = \sin 2\pi x \sin 2\pi y$.

Example 2 : An oscillatory case. In this example

$$a_1(x,y) = a_2(x,y) = (1 + \delta + \sin(4\pi x)\sin(4\pi y))^2.$$

The domain is $[0,1] \times [0,1]$ and $f(x,y) = 2\theta(\theta_x u_x + \theta_y u_y) - 8\pi^2\theta^2 u$, where $\theta = a_1(x,y)$ and $u = \sin 2\pi x \sin 2\pi y$ is the exact solution. This case is particularly difficult to solve since $a_1(x,y)$ and $a_2(x,y)$ are oscillatory which deteriorates the accuracy of the basic solver and the overall rate of convergence with DC iterations. To improve the convergence we employed the DCWB iterations since simple DCWA iterations are predictably unstable here. Similar to Example 1, it was found that the DCWB iterations possess much better rate of convergence than with the DC iterations. They reduce the computational error by additional of 2-3 orders after first 10 iterations. The overall number of iterations required to get the final accuracy is more than twice less with DCWB than with DC. Table 16 presents the numerical results from using DCWB.

	$\delta = 3$		$\delta = 1$		
$N_1 \times N_2$	L_2 error	iter.	10 iter.	L_2 error	iter.
16×16	6.6E-5	6	7.7E-4	7.7E-4	7
32×32	5.6E-7	8	2.8E-5	4.0E-6	16
64×64	5.0E-9	10	8.3E-6	3.0E-8	28
128×128	3.6E-11	12	8.4E-6	1.0E-10	33

Table 16: Full convergence: L_2 error with DCWB where oscillatory coefficients were chosen and the exact solution is $u = \sin 2\pi x \sin 2\pi y$.

Example 3 : Discontinuous coefficients. We solve the equation when we have discontinuous coefficients $a_1(x,y) \neq a_2(x,y)$ which are step functions in the square $[0,1] \times [0,1]$:

$$a_1(x,y) = \begin{cases} 1-\delta, & \text{if} \quad x^2+y^2 \leq 0.4^2, \\ 1+\delta, & \text{if} \quad x \geq 0.6, \\ 1, & \text{otherwise.} \end{cases}$$

$$a_2(x,y) = \begin{cases} 1+2\delta, & \text{if} \quad x^2+y^2 \leq 0.4^2, \\ 1-2\delta, & \text{if} \quad x \geq 0.6, \\ 1, & \text{otherwise.} \end{cases}$$

Table 17 presents the results for the RHS which corresponds to the exact solution $u = \sin \pi x \sin \pi y$.

$N_1 \times N_2$	$\delta = 0.1$			$\delta = 0.25$		
	L_2 error	k	full convergence	L_2 error	k	full convergence
8×8	4.8e-5	3	9.8e-6	7.0e-5	4	1.6e-4
16×16	9.8e-6	4	1.5e-7	2.7e-5	4	1.5e-7
32×32	2.1e-6	4	1.4e-9	3.3e-6	5	1.5e-9
64×64	3.9e-7	4	1.2e-11	6.0e-7	6	1.1e-11
128×128	5.8e-8	6	5.0e-13	8.1e-8	8	5.3e-13

Table 17: L_2 errors for the DC method after k iterations and full convergence. Here the coefficients are step functions and the exact solution is $u = \sin \pi x \sin \pi y$

Figure 7 describes the convergence of the iterative process with $\delta = 0.1$.

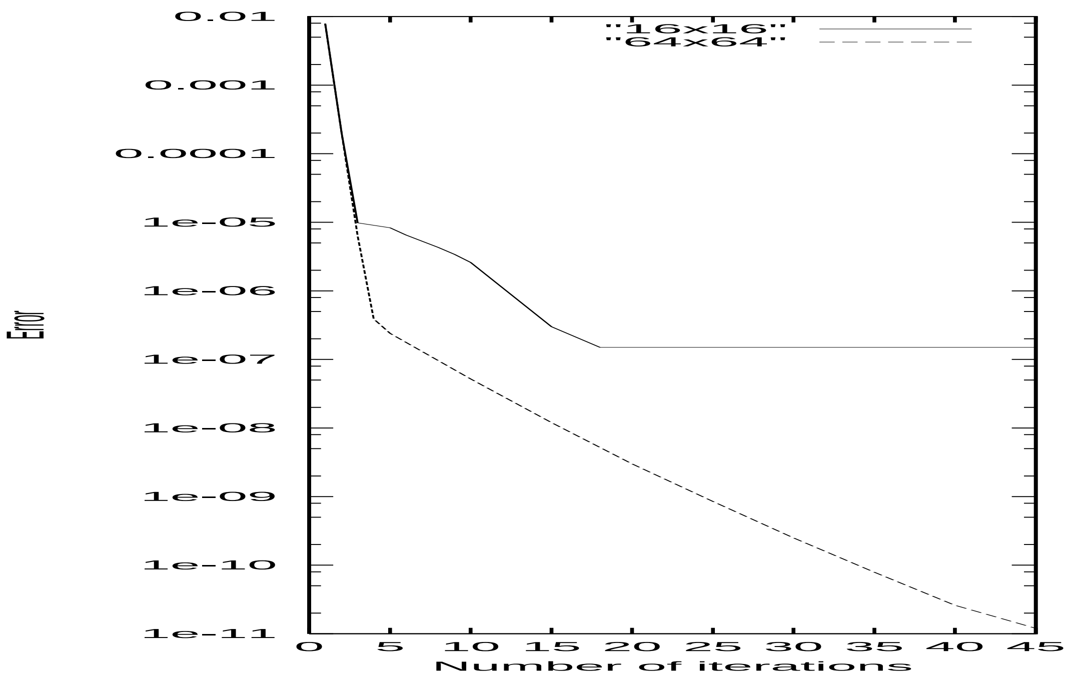

Figure 7: L_2 error for 16×16 and 64×64 grid points. $\delta = 0.1$

Example 4 : Poisson equation with $a_1 \gg a_2$. Here we solve the following Poisson equation with a small parameter:

$$\varepsilon u_{xx} + \frac{1}{\varepsilon} u_{yy} = f \tag{4.14}$$

where ε =const. If either ε or ε^{-1} are small positive numbers, then the basic solver fails to produce an accurate numerical solution. In order to prevent loss of accuracy it is common to increase the ratio N_2/N_1. However, for $\varepsilon \geq 15$ the basic solver is unable to produce an accurate solution even for 16×256 collocation points. To overcome this difficulty, we apply the technique mentioned in the beginning of Section 4.3, Remark 3. Instead of a direct application of the basic solver, we perform DC iterations with $\lambda_1 = \varepsilon$, $\lambda_2 = \varepsilon^{-1} + C_2$ where C_2 is a nonnegative constant. In this example, the RHS in Eq. (4.14) corresponds to the exact solution $u = \varepsilon^{-1}x^2 + \varepsilon y^2$. We can see from Table 18 that when $\varepsilon = 15$ accurate solutions are achieved after a relatively small number of iterations. The same technique was applied (using the same RHS) to $\varepsilon = 1000$ when the solution is highly decoupled in the x direction. When $\varepsilon = 1000$ then C_2 has to be greater than when we have $\varepsilon = 15$. In addition, in order to get a full convergence we need to have more iterations. The results for this case are given in Table 18:

$N_1 \times N_2$	ε	C_1	C_2	10 iter.	20 iter.	Full conv.,L_2 error	number of iter.
16×16	15	0.	0.20	7.5E-6	7.8E-7	5.9E-7	23
32×32		0.	0.25	1.6E-5	8.5E-7	2.8E-9	78
16×16	1000	0.	12.0	8.6E-4	9.5E-5	8.3E-5	25
32×32		0.	25.0	1.8E-2	5.7E-3	3.0E-7	200

Table 18: L_2 errors for the DC method. The exact solution is $u = \varepsilon^{-1}x^2 + \varepsilon y^2$

5 Solution of the Poisson equation in irregular domains

Examples 5 and 6 will use the iterative Poisson solver to solve Poisson equations in irregular domains. We transform the domain $\{(x, y) \in \Omega\}$ to a rectangle in (ξ, η)-domain by the orthogonal transformation

$$x = x(\xi, \eta), \quad y = y(\xi, \eta)$$

In the new coordinates ξ, η the original Poisson equation becomes:

$$\left[\frac{h_2}{h_1} u_\xi\right]_\xi + \left[\frac{h_1}{h_2} u_\eta\right]_\eta = h_1 h_2 f \tag{5.1}$$

where the coefficients h_1 and h_2 are defined by

$$h_1 = \sqrt{x_\xi^2 + y_\xi^2}, \quad h_2 = \sqrt{x_\eta^2 + y_\eta^2}.$$

The transformed Eq. (5.1) is solved in the computational domain by the above iterative method.

Example 5. Poisson equation in curved rectangles. We apply our solver to the Poisson equation in the domain bounded by the exponential curves $y = \exp(x)$, $y = \exp(x) + 1$, $y = \exp(-x)$, $y = \exp(-x) + 1$ (see Fig. 8).

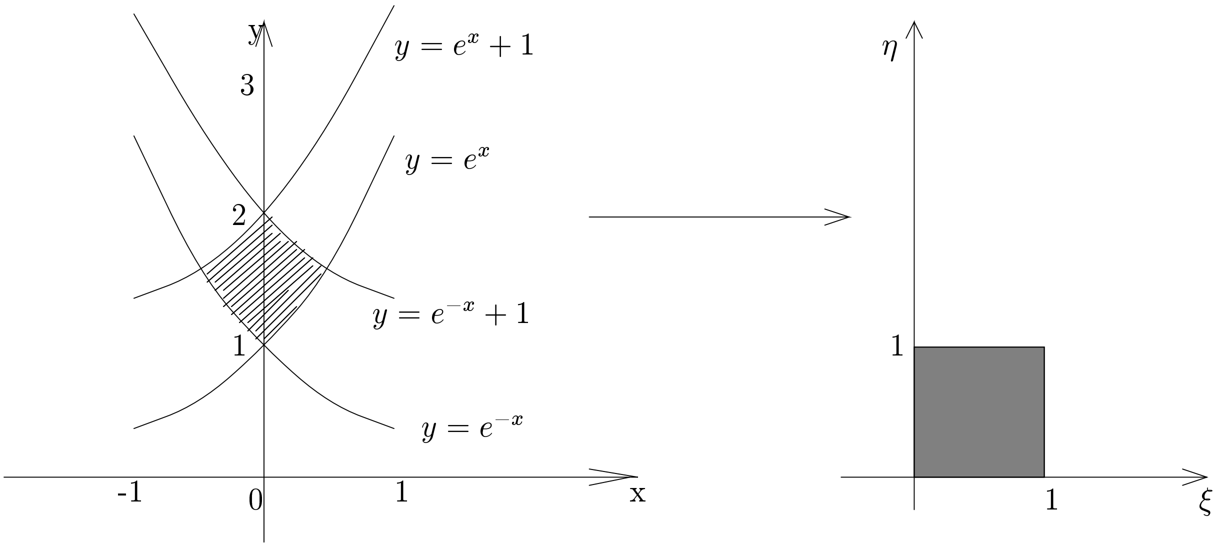

Figure 8: The curvilinear domain is transformed into a square.

In terms of the curvilinear coordinates

$$\xi = y - \exp(-x), \quad \eta = y - \exp(x)$$

the domain becomes a rectangle $[0,1] \times [0,1]$ in the computational plane $\{\xi, \eta\}$. In this plane we get, following the notation in the beginning of Section 4.3

$$h_1^2 = (1+\theta^2)/((\xi-\eta)^2+4), \quad h_2^2 = (1+\theta^2)/\theta^2((\xi-\eta)^2+4)$$

where $\theta = 0.5[\xi - \eta + \sqrt{(\xi-\eta)^2+4}]$. A trivial check confirms the orthogonality of the transformation. Thus, we have to solve the elliptic equation with variable coefficients $a_1(\xi,\eta) = h_2/h_1$, $a_2(\xi,\eta) = h_1/h_2$ and the RHS multiplied by $h_1 h_2$ as explained in the beginning of Section 4.3. The results are given in table (19) for the RHS which corresponds to the exact non-periodic solution $u = x^3 + y^3$.

	After 10 iterations	After 20 iterations	Full convergence	
$N_1 \times N_2$	L_2 Error	L_2 Error	L_2 Error	Iterations
8×8	1.2E-6	4.8E-8	4.8E-8	18
16×16	3.6E-7	1.9E-8	6.7E-10	33
32×32	9.6E-8	7.0E-9	7.1E-12	50
64×64	4.2E-8	2.3E-9	8.5E-13	70

Table 19: The iterations converge rapidly yielding an accurate solution after as much as 10-20 iterations. For full convergence, more iterations are needed when the number of collocation points increases.

Example 6: Poisson equation in irregular domains. Here we solve the Poisson equation in a irregular domain which lies in the first quadrant and is bounded by the following two curves:

$$\text{an ellipse} \quad x^2 + \frac{y^2}{2} = 1 \quad \text{and} \quad \text{a hyperbola} \quad \frac{x^2}{\varepsilon^2} - \frac{y^2}{1-\varepsilon^2} = 1,$$

where $1 > \varepsilon > 0$ is a parameter (see Fig. 9).

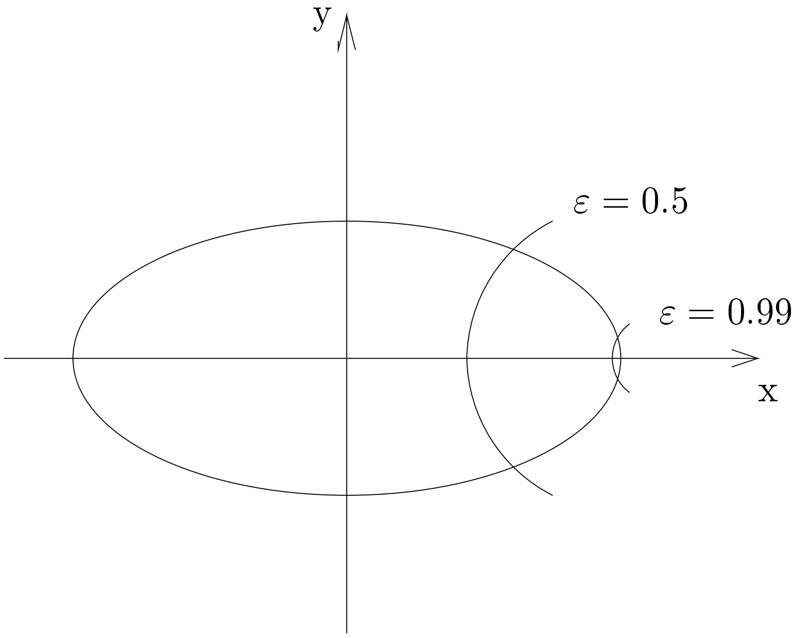

Figure 9: The curvilinear domain between the ellipse and the hyperbola (the first quadrant).

A simple check shows that the orthogonal transformation

$$x = \xi(1 - \varepsilon^2\eta^2)^{0.5}, \quad y = \varepsilon\eta(1 + \xi^2)^{0.5}$$

maps the above defined domain to a rectangle $[0,1] \times [0,1]$ in the computational plane (ξ, η). Similar to Example 5, we find the variable coefficients $a_1(\xi, \eta) = h_2/h_1$ and $a_2(\xi, \eta) = h_1/h_2$ by computing the coefficients h_1 and h_2:

$$h_1^2 = (1 + \xi^2 - \varepsilon^2\eta^2)/(1 + \xi^2), \quad h_2^2 = \varepsilon^2(1 + \xi^2 - \varepsilon^2\eta^2)/(1 - \varepsilon^2\eta^2)$$

and multiplying the RHS by $h_1 h_2$. Note that as ε goes to one then $a_2(\xi, \eta)$ goes to zero at $\eta = 1$ and thus $a_1(\xi, \eta)$ tends to infinity at the same point.

Consider the case with the exact solution $u = x_1^3 + x_2^3$ for two values of the parameter ε: $\varepsilon = .5$ and $\varepsilon = .99$. Results for $\varepsilon = .5$ are shown in Table 20. The iterations converge rapidly with an accurate solution available already after 10 iterations.

	After 10 iterations	After 20 iterations	Full convergence	
$N_1 \times N_2$	L_2 Error	L_2 Error	L_2 Error	Iterations
8×8	1.0e-7	1.0e-7	1.0e-7	6
16×16	3.2e-9	9.4e-10	9.4e-10	20
32×32	1.6e-9	6.3e-11	9.9e-12	25
64×64	5.6e-10	4.3e-11	4.1e-13	30
128×128	1.5e-10	1.9e-11	1.3e-13	40

Table 20: The convergence for $\varepsilon = 0.5$.

The case $\varepsilon = .99$ is almost singular as $a_1(\xi, \eta)$ becomes much bigger than $a_2(\xi, \eta)$ (in comparison with Example 4). To overcome this difficulty, more collocation points are required in the η direction. The results are given in Table 21. The closeness of ε to 1 also yields a slower rate of convergence than $\varepsilon = 0.5$. This results in a greater number of iterations which are required for obtaining an accurate solution. In order to improve the accuracy it is necessary to augment the ratio between N_2 and N_1. It can be done using the same technique as in Example 4, that is to perform DC iterations by means of a Poisson solver with augmented values of constant coefficients. Specifically, we add a positive value of C_2 to the second coefficient. As can be seen from Table 21, the resulting iterations converge and produce better than otherwise accuracy at a given number of collocation points.

		After 10 iter.	After 20 iter.	Full convergence	
	$N_1 \times N_2$	L_2 Error	L_2 Error	L_2 Error	Iterations
$C_1 = C_2 = 0$	16×64	2.2E-3	4.3E-5	8.2E-6	30
$C_1 = C_2 = 0$	16×128	2.2E-3	3.6E-5	9.5E-7	35
$C_1 = C_2 = 0$	32×128	2.2E-3	3.7E-5	2.2E-7	40
$C_1 = C_2 = 0$	16×256	diverges			
$C_1 = 0, C_2 = 0.1$	16×256	2.5E-3	4.9E-5	2.0-8	50

Table 21: The convergence for $\varepsilon = 0.99$.

6 Summary

This paper continues the flow of our previous results [1, 2] on fast direct spectral solvers for the Poisson equation and the Helmholtz equation. It presents the following novel elements:

1. A fast 2-D spectral algorithm for the solution of the Poisson equation and the (modified) Helmholtz equation is developed which does not require an extension of the original domain. This is a major improvement in comparison to [1, 2, 6, 7]. It takes $O(N^2 \log N)$ operations, where N is the number of collocation points in each direction (compared to $O((N+k)^2 \log N)$ for the previous 2D algorithms, where k is a number of extension points). The method is based on the eigenfunction expansion of the right hand side with successive integration. Both the right hand side and the boundary conditions are not assumed to have any periodicity properties. Compared to the spectral schemes based on Chebyshev polynomials the method does not need to find the inverse of the

full matrix (which results in $O(N^3)$ operations). All the integrations are direct which also reduces the round-off error.

2. Iterative procedure, which is based on this algorithm, was developed for elliptic equations with non-constant coefficients. The procedure enjoys the following properties: fast convergence and high accuracy even when the computation employs a small number of collocation points. The basic solver was used as a preconditioner. The algorithm was also applied to resolve the Poisson equation in irregular geometries.

The algorithm can be developed in the following directions:

1. The present solver can be incorporated as a part of Domain Decomposition algorithm for the resolution of the Poisson equation or the Helmholtz equation in regular subdomains, similar to [20].

2. The fast spectral subtractional solver can be extended to the 3-D case. Once developed, it can be employed as a basic algorithm for the iterative solution of the 3-D elliptic equations with non-constant coefficients and to the solution of the Poisson equation in complex geometries.

References

[1] A. Averbuch, M. Israeli, L. Vozovoi, On fast direct elliptic solver by modified Fourier method, *Numerical Algorithms* **15** (1997), 287-313.

[2] A. Averbuch, M. Israeli, L. Vozovoi, A fast Poisson solver of arbitrary order accuracy in rectangular regions, *SIAM J. Sci. Comput.* **19** (1998), 933-952.

[3] A.K. Bangia, P.F. Batcho, I.G. Kevrekidis and G.E. Karniadakis, Unsteady two-dimensional flows in complex geometries: comparative bifurcation studies with global eigenfunction expansions, *SIAM J. Sci. Comput.* **18** (1997), no. 3, 775–805.

[4] G. Baszenski and F.-J. Delvos, Accelerating the rate of convergence of bivariate Fourier expansions. Approximation theory, IV (College Station, Tex., 1983), 335–340, Academic Press, New York, 1983.

[5] G. Baszenski, F.-J. Delvos and M. Tasche, A united approach to accelerating trigonometric expansions. Concrete analysis, *Comput. Math. Appl.* **30** (1995), No. 3-6, 33–49.

[6] E. Braverman, M. Israeli, A. Averbuch, L. Vozovoi, A fast 3-D Poisson solver of arbitrary order accuracy, *J. Comput. Phys.* **144** (1998), 109-136.

[7] E. Braverman, M. Israeli and A. Averbuch, A fast spectral solver for 3-D Helmholtz equation, *SIAM J. Sci. Comput.* **20** (1999), no. 6, 2237-2260.

[8] C. Canuto, M.Y. Hussaini, A. Quarteroni, T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1989.

[9] H. Chen, Y. Su and B. D. Shizgal, A direct spectral collocation Poisson solver in polar and cylindrical coordinates, *J. Comput. Phys.* **160** (2000), 453-469.

[10] C.D. Dimitropoulos, B.J. Edwards, K.S. Chae and A.N. Beris, Efficient pseudospectral flow simulations in moderately complex geometries. *J. Comput. Phys.* **144** (1998), no. 2, 517–549.

[11] K. Eckhoff, On a high order numerical method for functions with singularities, *Math. Comp.* **67** (1998), no. 223, 1063–1087.

[12] O.F. Næss and K.S. Eckhoff, A modified Fourier-Galerkin method for the Poisson and Helmholtz equations, *J. Sci. Comput.* **17** (2002), no. 1-4, 529–539.

[13] M. Elghaoui and R. Pasquetti, A spectral embedding method applied to the advection-diffusion equation, *J. Comput. Phys.* **125** (1996), 464-476.

[14] M. Elghaoui and R. Pasquetti, Mixed Spectral-Boundary Element embedding algorithms for the Navier-Stokes equations in the vorticity-stream function formulation, *J. Comput. Phys.* **153** (1999), 82-100.

[15] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM: Philadelphia, 1977.

[16] D. Gottlieb and C.W. Shu, On the Gibbs phenomenon and its resolution, *SIAM Review* **39** (1997), no. 4, 644-668.

[17] L. Greengard and J.-Y. Lee, A direct adaptive Poisson solver of arbitrary order accuracy, *J. Comput. Phys.* **125** (1996), 415-424.

[18] M. Israeli, L. Vozovoi and A. Averbuch, Domain decomposition methods for solving parabolic PDEs on multiprocessors, *Appl. Numer. Math.* **12** (1993), 193-212.

[19] M. Israeli, L. Vozovoi and A. Averbuch, Spectral multi-domain technique with Local Fourier basis, *J. Scientific Computing* **8** (1993), 135-149.

[20] M. Israeli, E. Braverman and A. Averbuch, A hierarchical 3-D Poisson modified Fourier solver by domain decomposition, *J. Sci. Comput.* **17** (2002), no. 1-4, 471–479.

[21] E.G. Karniadakis, M. Israeli and S.A. Orszag, High-Order splitting methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **97** (1991), 414-443.

[22] A. McKenney, L. Greengard and A. Mayo, A fast Poisson solver for complex geometries, *J. Comput. Phys.* **118** (1995), 348-355.

[23] S. Orszag, Spectral methods for problems in complex geometries, *Advances in computer methods for partial differential equations, III* (Proc. Third IMACS Internat. Sympos., Lehigh Univ., Bethlehem, Pa., 1979), pp. 148–157, IMACS, New Brunswick, N.J., 1979.

[24] M. R. Schumack, W. W. Schultz and J. P. Boyd, Spectral method solution of the Stokes equations on non-staggered grids, *J. Comput. Phys.* **94**, (1991) 30.

[25] G. Skölermo, A Fourier method for numerical solution of Poisson's equation, *Math. Comput. (U.S.A.)* **29**, (1975) 697-711.

[26] L. Vozovoi, M. Israeli, A. Averbuch, Application of the multidomain local Fourier method for CFD in complex geometries, Computational Fluid Dynamics (Eds. D. Leutloff and R.C. Srivastava), Springer-Verlag: Berlin, Heidelberg, NY, 1995, pp. 245– 256.