An efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions

Hanquan Wang[*] Yong Zhang[†] Xiu Ma[*] Jun Qiu[*] Yan Liang[‡]

Abstract

Fourth-order compact finite difference scheme has been proposed for solving the Poisson equation with Dirichlet boundary conditions for some time. An efficient implementation of such scheme is often desired for practical usage. In this paper, based on fast discrete Sine transform, we design such an efficient algorithm. To do this, Poisson equation is first discretized by fourth-order compact finite difference method. The subsequent discretized system is not solved by the usual method–matrix inversion, instead it is solved with the fast discrete Sine transform. Detailed numerical algorithm of this fast solver for one-dimensional, two-dimensional and three dimensional Poisson equation have been presented. Numerical results in one dimension, two dimensions, three dimensions and four dimensions have shown that the applied compact finite difference scheme has fourth order accuracy and can be efficiently implemented.

Keywords Poisson equation, fourth-order, compact finite difference scheme, discrete Sine transform

1 Introduction

Efficient and accurate numerical methods for numerical approximations of partial differential equations appearing in science and engineering have been a goal of mathematicians, engineers, physicists, and other scientists for decades. In the last fifty years, many numerical approaches, which include finite difference method, finite element method, spectral method and finite volume method, have dominated the numerical approximations of partial differential equations. However, the finite difference method remains as a fundamental technique in solving

[*]School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, Yunnan Province, P. R. China, 650221. Email address: *hanquan.wang@gmail.com.*

[†]Wolfgang Pauli Institute c/o Fak. Mathematik, University Wien, Oskar-Morgenstern-Platz, 1090 Vienna, Austria. Email address: *yong.zhang@univie.ac.at.*

[‡]School of Mathematics and Statistics, Yunnan University, Kunming, Yunnan Province, P. R. China, 650091.

partial differential equation appeared in diverse physical fields such as quantum mechanics, electro-magnetics, and fluid mechanics.

Numerical schemes of second order explicit finite difference schemes are commonly used because their implementation is relatively simple. For example, a typical explicit finite difference scheme to approximate the first derivative of function $u(x)$ at point x_i is the centered scheme given by $(u_{i+1} - u_{i-1})/(2h)$, where h is the mesh size of a uniform partition of the definition domain of $u(x)$. The local truncation error of this approximation is $O(h^2)$. Better approximations can be obtained by increasing the order of the truncation error of the finite difference scheme. This is commonly accomplished by including more points in the stencil of the numerical schemes. As an example, consider an explicit centered finite difference formula with a five point stencil approximating the first derivative u'_i by

$$\frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12h}, \tag{1.1}$$

which has a local truncation error of $O(h^4)$. A disadvantage of this approach is the need to include more equations for grid points near and at the boundaries.

An alternative is to not enlarge the stencil, but involve values of the derivative at some nodes where the function is already evaluated. For instance, consider the finite difference approximation of the first derivative proposed in 1966 by Collatz [3], which approximates the derivative values at three grid points with function values over the same three grid points:

$$\frac{1}{4}u'_{i-1} + u'_i + \frac{1}{4}u'_{i+1} = \frac{3}{4h}(u_{i+1} - u_{i-1}). \tag{1.2}$$

As it will be proven later, this new scheme has a local truncation error of $O(h^4)$.

However, if (1.1) is used over a discretized domain, four additional formulas are needed at the two points on both ends where the stencil protrudes the domain. On the contrary, scheme (1.2) only requires additional formulas at each of the endpoints. Assuming that at least one boundary condition is known, only one additional formula may be needed. Thus the proposed implicit scheme (1.2) gives an advantage over the explicit one (1.1). Scheme (1.2) is usually called a compact finite difference scheme [3, 9].

Compact finite difference methods have been known for almost fifty years. As mentioned above, some particular formulas are reported by Collatz [3]. Their implementation as finite difference schemes approximating partial differential equations began in the early 1970s for some fluid mechanics problems [6, 14]. Since that time, several distinct classes of compact finite difference schemes have been developed. The two most common are the centered schemes and the upwind ones. In 1992, a seminal paper with an in-depth analysis of centered compact schemes showed that these compact schemes have spectral-like resolution for short waves [9]. Upwind compact schemes were also developed for solving nonlinear hyperbolic problems [4, 22]. In recent years, due to the appearance of faster and more powerful computing possibilities as well as the development of algorithms for fast matrix inversion, compact schemes are proving more advantageous. For instance, compact schemes on several problems have been applied into dealing with wall-bounded flows described by the Navier- Stokes equations, in large-eddy simulation of supersonic boundary-layer flow, and also in the scattering of electromagnetic waves [11, 13, 15] . Compact finite difference methods have been found in applications into solving convection-diffusion equation [20], Burgers' equation [8], wave equation [2], Euler equations [1], Schrödinger equation [5, 7, 16, 18, 21], Gross-Pitaevskii equations [19, 17], and the Poisson equation [12].

Although compact finite difference scheme can get higher order accuracy using fewer cell points, they are implicit methods and usually one needs to do matrix inversion when applied to solving partial differential equations. In this paper, a fast implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions is proposed. We discretize the Dirichlet boundary value problem of Poisson equation with the fourth-order compact finite difference method, and solve the resulting discretized system with fast discrete Sine transform, instead of matrix inversion. Poisson equation is one of most important partial differential equations. It has widespread application in electro-magnetics, mechanical engineering and theoretical physics. Design of fast solver for numerically solving Poisson equation may become imperative and necessary for compact finite difference method's application. It may be helpful for us to develop fast and efficient algorithm for many other kinds of partial differential equations when discretizing them with compact finite difference schemes.

The paper is organized as follows: we first briefly introduce how to design the compact finite difference scheme in Section 2. In Section 3, We next show how to discretize Poisson equation with the fourth-order compact finite difference scheme and how to apply the fast discrete Sine transform to solve the resulting discretized system. Detailed numerical algorithm for Poisson equation in one dimension, two dimensions and three dimensions have been presented as well. In Section 4, numerical results for Poisson equation in one dimension, two dimensions, three dimensions and four dimensions discretized by the fourth-order compact difference method are shown, respectively. In Section 5, some conclusions are drawn and discussions are made.

2 Compact finite difference scheme

In this section, we briefly introduce how compact finite difference scheme can be constructed. More information on how to construct them can be found in [9].

Before we start to approximate the derivative of $u(x)$ at grid points $x_i = a + ih$ $(0 < i < M)$ by the compact finite difference method, we assume that $u_i := u(x_i)$, $u'_i := \left(\frac{du}{dx}\right)(x_i)$, $u''_i := \left(\frac{d^2u}{dx^2}\right)(x_i)$, $u'''_i := \left(\frac{d^3u}{dx^3}\right)(x_i)$, $u''''_i := \left(\frac{d^4u}{dx^4}\right)(x_i)$, where i and M are some integers. a and b are some constants and $h = \frac{b-a}{M}$ stands for the mesh size in space x-direction.

Traditionally, finite difference method approximates the derivative of function $u(x)$ at x_i through a linear combination of the value of function $u(x)$ at neighboring grid points around x_i. For example, one can approximate u_i' by $(u_{i+1}-u_{i-1})/(2h)$, u_i'' by $(u_{i-1}-2u_i+u_{i+1})/h^2$, and u_i'''' by $(-u_{j+2}+16u_{j+1}-30u_j+16u_{j-1}-u_{j-2})/(12h^2)$. This is an explicit way to approximate the derivative of function $u(x)$ at x_i through the function $u(x)$ itself at neighboring points.

However, compact finite difference method implicitly approximates the derivative of function $u(x)$ at x_i. In the following, we introduce the basic idea of how to construct the compact finite difference method to approximate the first-order derivative and the second-order derivative of $u(x)$ at x_i.

2.1 Approximation of the first-order derivative

The general centered compact finite difference method gives an approximation of the first-order derivative through [9]

$$\begin{aligned}&\beta u_{i-2}'+\alpha u_{i-1}'+u_i'+\alpha u_{i+1}'+\beta u_{i+2}'\\&=c\frac{u_{i+3}-u_{i-3}}{6h}+b\frac{u_{i+2}-u_{i-2}}{4h}+a\frac{u_{i+1}-u_{i-1}}{2h},\end{aligned}\tag{2.1}$$

where α,β, a,b,c are some constants to be determined. By Taylor's expansion, we find

$$\begin{aligned}u(x_i\pm h)&=u(x_i)\pm u'(x_i)h+u''(x_i)\frac{h^2}{2!}\pm\cdots,\\u(x_i\pm 2h)&=u(x_i)\pm u'(x_i)2h+u''(x_i)\frac{(2h)^2}{2!}\pm\cdots,\\u(x_i\pm 3h)&=u(x_i)\pm u'(x_i)3h+u''(x_i)\frac{(3h)^2}{2!}\pm\cdots,\end{aligned}\tag{2.2}$$

and

$$\begin{aligned}u'(x_i\pm h)&=u'(x_i)\pm u''(x_i)h+u'''(x_i)\frac{h^2}{2!}\pm\cdots,\\u'(x_i\pm 2h)&=u'(x_i)\pm u''(x_i)2h+u'''(x_i)\frac{(2h)^2}{2!}\pm\cdots.\end{aligned}\tag{2.3}$$

Plugging Eq. (2.2) and Eq. (2.3) into Eq. (2.1), and matching the coefficients of similar terms, i.e., $O(1)$, (h^2), $O(h^4)$, $\cdots$, we can obtain equations for those undetermined coefficients α,β, a,b, and c

$$a+b+c = 1+2\alpha+2\beta, \tag{2.4}$$

$$a+2^2b+3^2c = 2\frac{3!}{2!}(\alpha+2^2\beta), \tag{2.5}$$

$$a+2^4b+3^4c = 2\frac{5!}{4!}(\alpha+2^4\beta), \tag{2.6}$$

$$\vdots$$

from which it follows: (1) when a, b, c, α, β satisfy Eq. (2.4), from Eq. (2.1), one can get a finite difference scheme with second-order accuracy; (2) when a, b, c, α, β satisfy both Eq. (2.4) and Eq. (2.5), from Eq. (2.1), one can get a finite difference scheme with fourth-order accuracy; (3) when a, b, c, α, β satisfy Eq. (2.4), Eq. (2.5), and Eq. (2.6) simultaneously, one can get a finite difference scheme with sixth-order accuracy. Higher-order finite difference scheme for the first order derivative can be constructed in a similar way.

Compact finite difference scheme can be simply derived by setting $b = c = 0$. For example, if we choose

$$\beta = 0,\ a = \frac{2}{3}(\alpha + 2),\ b = \frac{1}{3}(4\alpha - 1),\ c = 0, \tag{2.7}$$

and set $b = 0$, then from Eq. (2.7) we have

$$\alpha = 1/4,\ \beta = 0,\ a = 3/2,\ b = 0,\ c = 0. \tag{2.8}$$

Plugging Eq. (2.8) into Eq. (2.1), we obtain a compact finite difference scheme with fourth-order accuracy for u'_i as follows:

$$\frac{1}{4}u'_{i-1} + u'_i + \frac{1}{4}u'_{i+1} = \frac{3}{4h}(u_{i+1} - u_{i-1}). \tag{2.9}$$

It is obvious to see that a, b, c, α, β in Eq. (2.8) satisfy both Eq. (2.4) and Eq.(2.5), but they do not satisfy Eq.(2.6).

2.2 Approximation of the second-order derivative

The centered compact finite difference method obtains an approximation of the second-order derivative from [9]

$$\begin{gathered}\beta u''_{i-2} + \alpha u''_{i-1} + u'_i + \alpha u''_{i+1} + \beta u''_{i+2} = \\ c\frac{u_{i+3} - 2u_i + u_{i-3}}{9h^2} + b\frac{u_{i+2} - 2u_i + u_{i-2}}{4h^2} + a\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2},\end{gathered} \tag{2.10}$$

where $\alpha, \beta,\ a, b, c$ are again some constants to be determined.

By Taylor's expansion, we have

$$\begin{aligned}u''(x_i \pm h) &= u''(x_i) \pm u'''(x_i)h + u''''(x_i)\frac{h^2}{2!} \pm \cdots, \\ u''(x_i \pm 2h) &= u''(x_i) \pm u'''(x_i)2h + u''''(x_i)\frac{(2h)^2}{2!} \pm \cdots.\end{aligned} \tag{2.11}$$

Plugging Eq. (2.3) and Eq. (2.11) into Eq. (2.10), we find

$$a + b + c = 1 + 2\alpha + 2\beta, \tag{2.12}$$

$$a + 2^2 b + 3^2 c = \frac{4!}{2!}(\alpha + 2^2\beta), \tag{2.13}$$

$$a + 2^4 b + 3^4 c = \frac{6!}{4!}(\alpha + 2^4\beta), \tag{2.14}$$

$$\vdots$$

from which it follows: (1) when a, b, c, α, β satisfy Eq. (2.12), from Eq. (2.10), one can get a finite difference scheme with second-order accuracy; (2) when a, b, c, α, β satisfy both Eq. (2.12) and Eq. (2.13), from Eq. (2.10), one can get a finite difference scheme with fourth-order accuracy; (3) when a, b, c, α, β satisfy Eq. (2.12), Eq. (2.13), and Eq. (2.14) simultaneously, one can get a finite difference scheme with sixth-order accuracy.

Specifically, if one choose $\alpha = \frac{1}{10}, \beta = 0$, $a = \frac{6}{5}, b = 0, c = 0$, they satisfy Eq. (2.12) and Eq. (2.13), but they do not satisfy Eq. (2.14). Therefore we can get the following compact finite difference scheme with fourth-order accuracy for u_i'' [9, 21] as follows :

$$\frac{1}{10}u_{i-1}'' + u_i'' + \frac{1}{10}u_{i+1}'' = \frac{6}{5} \cdot \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \tag{2.15}$$

Of course, we can construct more high-order compact finite difference schemes from (2.12), (2.13) and (2.14), and refer to [9] for more details.

3 Approximation of Poisson equation by compact finite difference scheme

In this section, we aim to show how to use the fourth order compact finite difference scheme (2.15) to solve the Poisson equation with Dirichlet boundary conditions in one dimension, two dimensions and three dimensions, respectively. To reduce the computation cost, we construct a fast solver for the discretized system based on fast discrete Sine transform.

3.1 Poisson equation in one dimension

We solve the following one-dimensional Poisson equation with Dirichlet boundary conditions

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

where $a, b \in \mathbb{R}$, $u(x)$ is the unknown function and $f(x)$ is some given source function.

As the usual finite difference method, we first discrete the interval $[a, b]$ uniformly with grid points being $x_i = a + i\,h, h = (b-a)/M, i = 0, 1, 2, \cdots, M$ where M is positive integer. Hereafter, we denote $u_i \approx u(x_i), u_i^{''} \approx u''(x_i), i = 0, \cdots, M$ and $U = (u_1, u_2, \cdots, u_{M-1})^T$, $U'' = (u_1'', u_2'', \cdots, u_{M-1}'')^T$, and $F = (f_1, f_2, \cdots, f_{M-1})^T \in \mathbb{R}^{M-1}$.

From Eq. (2.15), we can rewrite the finite difference in a matrix formula as follows:

$$AU'' = BU, \tag{3.2}$$

where $A, B \in \mathbb{R}^{M-1\times M-1}$ are given explicitely:

$$A = \begin{pmatrix} 10 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 10 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 10 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 10 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 10 \end{pmatrix}, B = \frac{12}{h^2}\begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix}.$$

Since B is a tridiagonal invertible matrix, Eq. (3.2) can be rewritten as

$$U = B^{-1}AU''. \tag{3.3}$$

In addition, we have $-u''(x_i) = f(x_i),\ i = 1, 2\cdots, M-1$ from Eq. (3.1). Then (3.3) is rewritten as

$$U = -B^{-1}AF. \tag{3.4}$$

Solving (3.4) by matrix inversion, we can get the approximated solution U. Notice that in the above derivation, we have assumed that $u''_0 = u''_M = 0$. If not, we may use the approximation such as $u''_0 = -f(x_0)$ and $u''_M = -f(x_M)$ and can do similarly.

In the following, based on fast discrete Sine transform, we construct a fast solver for the linear system (3.4) and bypass the matrix inversion computation. It is well known that the discrete forward/backward Sine transform for $U \in \mathbb{R}^{M-1}$ and $\widehat{U}$ are given as follows:

$$u_j = \sum_{k=1}^{M-1} \widehat{u}_k \sin(\frac{jk\pi}{M}), \qquad j = 1, 2, \cdots, M-1, \tag{3.5}$$

$$\widehat{u}_k = \frac{2}{M}\sum_{j=1}^{M-1} u_j \sin(\frac{ik\pi}{M}), \quad k = 1, 2, \cdots, M-1, \tag{3.6}$$

from which, we can define $\widehat{U} = (\widehat{u}_1, \cdots, \widehat{u}_{M-1})^T$ as the discrete Sine transform of U.

From (3.5), we can approximate u_{i+1}, u_{i-1}, u''_i, u''_{i+1},u''_{i-1} as follows:

$$u_{i+1} = \sum_{k=1}^{M-1} \hat{u}_k \sin(\frac{(i+1)k\pi}{M}), \quad u_{i-1} = \sum_{k=1}^{M-1} \hat{u}_k \sin(\frac{(i-1)k\pi}{M}),$$

$$u''_i = \sum_{k=1}^{M-1} \widehat{u''}_k \sin(\frac{ik\pi}{M}), \quad u''_{i+1} = \sum_{k=1}^{M-1} \widehat{u''}_k \sin(\frac{(i+1)k\pi}{M}),$$

$$u''_{i-1} = \sum_{k=1}^{M-1} \widehat{u''}_k \sin(\frac{(i-1)k\pi}{M}).$$

Plugging the above equations into Eq.(2.15), we have

$$\sum_{k=1}^{M-1} \widehat{u''}_k \left\{ \frac{1}{10}\sin(\frac{(i-1)k\pi}{M}) + \sin(\frac{ik\pi}{M}) + \frac{1}{10}\sin(\frac{(i+1)k\pi}{M}) \right\}$$
$$= \frac{6}{5h^2}\sum_{k=1}^{M-1} \hat{u}_k \left\{ \sin(\frac{(i-1)k\pi}{M}) + \sin(\frac{(i+1)k\pi}{M}) - 2\sin(\frac{ik\pi}{M}) \right\}.$$

or in a simple form

$$\sum_{k=1}^{M-1} \widehat{u''}_k \left(\frac{1}{5}\cos(\frac{k\pi}{M}) + 1 \right) \sin(\frac{ik\pi}{M}) = \frac{6}{5h^2}\sum_{k=1}^{M-1} \hat{u}_k \left(2\cos(\frac{k\pi}{M}) - 2 \right) \sin(\frac{ik\pi}{M}).$$

Finally, we have

$$\hat{u}_k = -\hat{u}''_k \left(\frac{24\sin^2(\frac{k\pi}{2M})}{h^2} \right)^{-1} \left(\cos(\frac{k\pi}{M}) + 5 \right) \tag{3.7}$$

for $k = 1, 2, \cdots, M-1$.

In addition, from Eq. (3.1), we obtain $-u''_i = f_i$ $(i = 1, 2, \cdots, M-1)$. By the inverse Sine transform, we get to know $-\widehat{u''}_k = \hat{f}_k$ $(k = 1, 2, \cdots, M-1)$. In summary, we obtain:

Algorithm 1 Fast solver for the one-dimensional Poisson equation with Dirichlet boundary conditions

```
1. Given M and grid points $x_i$, compute $\hat{F}$ from $F$ via fast discrete Sine
   transform.
2. Compute $\hat{U}$ by (3.7) using $\hat{F}$, followed by the inverse discrete Sine
   transform of $\hat{U}$ to obtain $U$.
```

Before we jump to higher dimension problems, we would like to discuss first the efficiency and its generlization here.

As for the efficiency, in the algorithm, one can apply the fast discrete Sine transform, and they can help reduce the computational cost from $O(M^2)$ by direct summation of (3.5)-(3.6) to $O(M \log M)$ arithmetical operations. In fact, the fast discrete Sine transform is implemented via the Fast Fourier Transform (FFT), and we refer the readers to [10] for more details.

Compared with direct *matrix inversion* of (3.4), the computational cost will drop dramatically from $O(M^3)$ to $O(M \log M)$. Although the tridiagonal system in our case can be solved within $O(M)$ operations, direct extension to higher space dimensions will be tedious and quite details-involved, as to be shown in the coming subsections. While the extension of Algorithm (1) to higher space dimensions is quite straightforward by tensor product, and such advantages are quite essential for efficient implementation.

We also would like to point out the Algorithm 1 applies to non-homogenous source function f, i.e. $f(a) \neq 0$ or $f(b) \neq 0$. In fact, the Sine transform applies successfully simply because the discrete Sine basis happen to be eigenvectors of the tridiagonal matrix B. Once AF in (3.4) is known, by expanding the vectors into the discrete Sine basis, we shall obtain exactly the same relation as (3.7).

Therefore we know that the proposed algorithm can be efficiently implemented. In the next two subsections, we show that the above algorithm can be extended to solve two-dimensional problem and three-dimensional problem without much difficulty.

3.2 Poisson equation in two dimensions

In this subsection, we wish to apply the fourth-order compact finite difference scheme (2.15) into solving the following two-dimensional Poisson equation with Dirichlet boundary conditions

$$\begin{aligned} &-\Delta u = f \quad a < x < b,\ c < y < d, \\ &u(a,y) = 0, \quad u(b,y) = 0, \\ &u(x,c) = 0, \quad u(x,d) = 0, \end{aligned} \tag{3.8}$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, $a, b, c, d \in \mathbb{R}$, $u = u(x,y)$ are unknown and $f = f(x,y)$ is some given function.

In the numerical discretization, we first do the partition of the domain $[a,b] \times [c,d]$, and we get the mesh-size in x-direction and y-direction, i.e., $h_x = \frac{b-a}{M}, h_y = \frac{d-c}{N}$, respectively. Grid points are defined as (x_i, y_j) where $x_i = a + i\,h_x, i = 0, 1, 2, \cdots, M,\ y_j = c + j\,h_y, j = 0, 1, 2, \cdots, N$. In addition, u_{ij} are denoted as approximation of $u(x_i, y_j)$. Similarly, u_{ij}^{xx} and u_{ij}^{yy} are defined as the approximation of the second-order partial derivatives $u_{xx}(x_i, y_j)$ and $u_{yy}(x_i, y_j)$, respectively.

Taking into consideration of the compact finite difference scheme of (2.15), we can obtain the following approximations

$$u_{i-1,j}^{xx} + 10u_{ij}^{xx} + u_{i+1,j}^{xx} = \frac{12}{h_x^2}(u_{i-1,j} - 2u_{ij} + u_{i+1,j}), \tag{3.9}$$

$$u_{i-1,j}^{yy} + 10u_{ij}^{yy} + u_{i+1,j}^{yy} = \frac{12}{h_y^2}(u_{i-1,j} - 2u_{ij} + u_{i+1,j}), \tag{3.10}$$

$$i = 1, 2, \cdots, M-1,\ j = 1, 2, \cdots, N-1.$$

By means of Kronecker product $\bigotimes$, one can reformulate Eqs. (3.9) and (3.10) into the following matrix form

$$A_x \bigotimes I\ U^{xx} = \frac{12}{h_x^2} B_x \bigotimes I\ U, \tag{3.11}$$

$$I \bigotimes A_y\ U^{yy} = \frac{12}{h_y^2} I \bigotimes B_y\ U, \tag{3.12}$$

where $A_x, B_x \in \mathbb{R}^{M-1\times M-1}$, $A_y, B_y \in \mathbb{R}^{N-1\times N-1}$ with A_x, A_y and B_x, B_y taking the general form of $\widetilde{A}$ and $\widetilde{B}$ respectively:

$$\widetilde{A} = \begin{pmatrix} 10 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 10 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 10 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 10 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 10 \end{pmatrix}, \widetilde{B} = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix},$$

$$U^{xx} = \begin{pmatrix} u_{11}^{xx} \\ \vdots \\ u_{1,N-1}^{xx} \\ u_{21}^{xx} \\ \vdots \\ u_{2,N-1}^{xx} \\ \vdots \\ u_{M-1,1}^{xx} \\ \vdots \\ u_{M-1,N-1}^{xx} \end{pmatrix}, \quad U^{yy} = \begin{pmatrix} u_{11}^{yy} \\ \vdots \\ u_{1,N-1}^{yy} \\ u_{21}^{yy} \\ \vdots \\ u_{2,N-1}^{yy} \\ \vdots \\ u_{M-1,1}^{yy} \\ \vdots \\ u_{M-1,N-1}^{yy} \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} \\ \vdots \\ u_{1,N-1} \\ u_{21} \\ \vdots \\ u_{2,N-1} \\ \vdots \\ u_{M-1,1} \\ \vdots \\ u_{M-1,N-1} \end{pmatrix}. \tag{3.13}$$

Furthermore, setting

$$A_1 = A_x \bigotimes I, \quad A_2 = I \bigotimes A_y,$$
$$B_1 = B_x \bigotimes I, \quad B_2 = I \bigotimes B_y,$$

Eqs. (3.11) and (3.12) can be written as

$$A_1 U^{xx} = \frac{12}{h_x^2} B_1 U, \tag{3.14}$$

$$A_2 U^{yy} = \frac{12}{h_y^2} B_2 U, \tag{3.15}$$

From which, it follows that

$$U^{xx} = \frac{12}{h_x^2} A_1^{-1} B_1 U, \tag{3.16}$$

$$U^{yy} = \frac{12}{h_y^2} A_2^{-1} B_2 U. \tag{3.17}$$

Here, we have assumed that $u''_{0j} = u''_{Mj} = 0$ for all $0 \le j \le N$ and $u''_{i0} = u''_{iN} = 0$ for all $0 \le i \le M$.

Moreover, Eq. (3.8) at points (x_i, y_j) can be discretized as follows

$$-(U_{ij}^{xx} + U_{ij}^{yy}) = F_{ij}, \quad i = 1, \cdots, M-1, \;\; j = 1, \cdots, N-1, \tag{3.18}$$

or in matrix formulation

$$-(U^{xx}+U^{yy})=F, \tag{3.19}$$

where

$$F=\begin{pmatrix} f_{11}, f_{12}, \cdots f_{1,N-1}, f_{21}, \cdots f_{2,N-1}, \cdots f_{M-1,1}, \cdots f_{M-1,N-1} \end{pmatrix}^T. \tag{3.20}$$

Combining Eqs. (3.19), (3.16) and (3.17), we have

$$(\frac{12}{h_x^2}A_1^{-1}B_1+\frac{12}{h_y^2}A_2^{-1}B_2)U=-F. \tag{3.21}$$

Finally, in matrix form, we have

$$U=-C^{-1}F, \qquad C=\frac{12}{h_x^2}A_1^{-1}B_1+\frac{12}{h_y^2}A_2^{-1}B_2. \tag{3.22}$$

we can summarize the whole numerical procedure as follows:

1. Given a, b, M, N and grid points (x_i, y_j) , $i=0,1,\cdots,M, j=0,1,\cdots,N$, compute F in (3.20) by evaluation of $f(x_i,y_j)$.
2. Compute A_1, B_1, A_2, B_2, the inverse matrices of A_1, A_2 and C defined in Eq. (3.22).
3. Perform the matrix inversion of C and get U via Eq. (3.22).

If following the procedure above, one needs to do matrix inversion. It will become more costly when M and N are increased. Iterative method might be a choice for efficient matrix inversion and can be used to reduce the cost of matrix inversion.

In the following, we solve the above discretization with two-dimensional discrete Sine transform. We note that the discrete Sine transform for u_{ij} and its inverse for $\hat{u}_{kl}$ in two dimensions are defined as

$$u_{ij}=\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\hat{u}_{kl}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N}), \tag{3.23}$$

$$\hat{u}_{kl}=\frac{2}{M}\frac{2}{N}\sum_{i=1}^{M-1}\sum_{j=1}^{N-1}u_{ij}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N}). \tag{3.24}$$

In addition, we note that

$$u_{ij}=\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\hat{u}_{kl}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N}), \tag{3.25}$$

$$u_{i+1,j}=\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\hat{u}_{kl}\sin(\frac{(i+1)k\pi}{M})\sin(\frac{jl\pi}{N}), \tag{3.26}$$

$$u_{i-1,j}=\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\hat{u}_{kl}\sin(\frac{(i-1)k\pi}{M})\sin(\frac{jl\pi}{N}), \tag{3.27}$$

and

$$u_{ij}^{xx} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{xx} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}), \tag{3.28}$$

$$u_{i-1,j}^{xx} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{xx} \sin(\frac{(i-1)k\pi}{M}) \sin(\frac{jl\pi}{N}), \tag{3.29}$$

$$u_{i+1,j}^{xx} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{xx} \sin(\frac{(i+1)k\pi}{M}) \sin(\frac{jl\pi}{N}). \tag{3.30}$$

Plugging them into Eq. (3.9), we obtain

$$\begin{aligned} &\sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{xx} \sin(\frac{ik\pi}{M}) \sin(\frac{il\pi}{N}) \left(\frac{1}{5}\cos(\frac{k\pi}{M}) + 1\right) \\ &= \frac{6}{5h_x^2} \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl} \sin(\frac{jl\pi}{N}) \sin(\frac{ik\pi}{M}) \left(2\cos(\frac{k\pi}{M}) - 2\right), \end{aligned} \tag{3.31}$$

from which it follows:

$$\hat{u}_{kl}^{xx} = \hat{u}_{kl} \frac{-\frac{24}{h_x^2}\sin^2(\frac{k\pi}{2M})}{\cos(\frac{k\pi}{M}) + 5}. \tag{3.32}$$

Similarly, we know that

$$u_{ij} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}), \tag{3.33}$$

$$u_{i,j+1} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl} \sin(\frac{ik\pi}{M}) \sin(\frac{(j+1)l\pi}{N}), \tag{3.34}$$

$$u_{i,j-1} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl} \sin(\frac{ik\pi}{M}) \sin(\frac{(j-1)l\pi}{N}), \tag{3.35}$$

and

$$u_{ij}^{yy} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{yy} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}), \tag{3.36}$$

$$u_{i,j-1}^{yy} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{yy} \sin(\frac{ik\pi}{M}) \sin(\frac{(j-1)l\pi}{N}), \tag{3.37}$$

$$u_{i,j+1}^{yy} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1} \hat{u}_{kl}^{yy} \sin(\frac{ik\pi}{M}) \sin(\frac{(j+1)l\pi}{N}). \tag{3.38}$$

Plugging them into Eq. (3.10) will give us

$$\hat{u}_{kl}^{yy} = \hat{u}_{kl} \frac{-\frac{24}{h_y^2}\sin^2(\frac{l\pi}{2N})}{\cos(\frac{l\pi}{N}) + 5}. \tag{3.39}$$

In addition, the two-dimensional Poisson equation $-\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$ at point (x_i, y_j) can be discretized as $-(u_{ij}^{xx} + u_{ij}^{yy}) = f_{ij}$. By doing the inverse Sine transform, we can get

$$-(\hat{u}_{kl}^{xx} + \hat{u}_{kl}^{yy}) = \hat{f}_{kl}, k = 1, 2 \cdots, M-1, \ l = 1, 2, \cdots, N-1. \tag{3.40}$$

for all k, l.

Finally, we combine Eqs. (3.32), (3.39) and (3.40) together, we reach

$$-\hat{f}_{kl} = \hat{u}_{kl} \left(\frac{-\frac{24}{h_x^2} \sin^2(\frac{k\pi}{2M})}{\cos(\frac{k\pi}{M}) + 5} + \frac{-\frac{24}{h_y^2} \sin^2(\frac{l\pi}{2N})}{\cos(\frac{l\pi}{N}) + 5} \right), \tag{3.41}$$

where $\hat{f}_{kl}$ can be calculated from

$$\hat{f}_{kl} = \frac{2}{M} \frac{2}{N} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} f_{ij} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}) \tag{3.42}$$

with $f_{ij} = f(x_i, y_j)$ being known value.

Thus, from Eqs. (3.41) and (3.42) we obtain $\hat{u}_{kl}$ and by means of discrete Sine transform one get the approximation of $u(x,y)$ at grid points (x_i, y_j), i.e., u_{ij} for all $i = 1, 2 \cdots, M-1$ and $j = 1, 2, \cdots, N-1$. We summarise the whole procedure as follows:

Algorithm 2 Fast solver for two-dimensional Poisson equation with Dirichlet boundary conditions

1. Given integers M N, and grid points (x_i, y_j), evaluate F and compute its discrete sine transform $\hat{F}$.
2. Compute $\hat{U}$ by Eq. (3.41) and perform the inverse discrete sine transform to get U.

3.3 Poisson equation in three dimensions

In this subsection, we aim to apply the fourth-order compact finite difference scheme (2.15) into solving three-dimensional Poisson equation with Dirichlet boundary conditions

$$\begin{aligned} &-\triangle u = f(x,y,z) \quad a < x < b, c < y < d, e < z < f, \\ &u(a,y,z) = 0, \quad u(b,y,z) = 0 \quad u(x,y,e) = 0, \\ &u(x,c,z) = 0, \quad u(x,d,z) = 0 \quad u(x,y,f) = 0. \end{aligned} \tag{3.43}$$

For numerical discretization in three dimension, we do partition on the definition domain $[a,b] \times [c,d] \times [e,f]$ and get the mesh size in $x-, y-, z-$ direction respectively, i.e., $h_x = (b-a)/M, h_y = (d-c)/N$, $h_z = (f-e)/P$ (M, N, P are some positive integers), grid points (x_i, y_j, z_p) ($x_i = a + ih_x$ ($i = 0, 1, \cdots, M$), $y_j = c + jh_y$ ($j = 0, 1, \cdots, N$), $z_p = e + ph_z$ ($p = 0, 1, \cdots, P$)). We denote u_{ijp}, u^{xx}_{ijp}, u^{xx}_{ijp}, u^{yy}_{ijp}, u^{zz}_{ijp} to be the approximation of functions $u(x, y, z)$, $u_{xx}(x, y, z)$, $u_{yy}(x, y, z)$, $u_{zz}(x, y, z)$ at grid points (x_i, y_j, z_p), respectively.

By extending the compact finite difference scheme of (2.15) into three dimensions, we can obtain the following approximations for the second-order partial derivative u^{xx}_{ijk}, u^{yy}_{ijk} and u^{zz}_{ijk} respectively,

$$u^{xx}_{i-1,j,p} + 10u^{xx}_{ijp} + u^{xx}_{i+1,j,p} = \frac{12}{h_x^2}(u_{i-1,j,p} - 2u_{ijp} + u_{i+1,j,p}), \quad (3.44)$$

$$u^{yy}_{i,j-1,p} + 10u^{yy}_{ijp} + u^{yy}_{i,j+1,p} = \frac{12}{h_y^2}(u_{i,j-1,p} - 2u_{ijp} + u_{i,j+1,p}), \quad (3.45)$$

$$u^{zz}_{i,j,p-1} + 10u^{zz}_{ijp} + u^{zz}_{i,j,p+1} = \frac{12}{h_z^2}(u_{i,j,p-1} - 2u_{ijp} + u_{i,j,p+1}), \quad (3.46)$$

for all $i = 1, 2, \cdots, M-1, j = 1, 2, \cdots, N-1, p = 1, 2, \cdots, P-1$.

In addition, by discretizing Eq. (3.43) at grid points (x_i, y_j, z_p), we get

$$-(u^{xx}_{ijp} + u^{yy}_{ijp} + u^{zz}_{ijp}) = f_{ijp}. \quad (3.47)$$

Eqs. (3.44)-(3.47) could be rewritten into matrix formulation as what we have done in two dimensions, from which, we might obtain the approximation of u_{ijp} for all $i = 1, 2, \cdots, M-1, j = 1, 2, \cdots, N-1, p = 1, 2, \cdots, P-1$. Doing numerical computation in this way would be more costly especially when integers M, N, P become larger. We do not write down the detailed matrix formulation, as we do not wish to do matrix inversion.

Here we construct a fast solver for finding u_{ijp} for all $i = 1, 2, \cdots, M-1, j = 1, 2, \cdots, N-1, p = 1, 2, \cdots, P-1$ from Eqs. (3.44)-(3.47). We note that the discrete Sine transform for u_{ijk} in three dimensions and its inverse are as follows:

$$u_{ijp} = \sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1} \hat{u}_{klq} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}) \sin(\frac{qp\pi}{P}), \quad (3.48)$$

$$\hat{u}_{klq} = \frac{2}{M}\frac{2}{N}\frac{2}{P}\sum_{i=1}^{M-1}\sum_{j=1}^{N-1}\sum_{p=1}^{P-1} u_{ijp} \sin(\frac{ik\pi}{M}) \sin(\frac{jl\pi}{N}) \sin(\frac{qp\pi}{P}), \quad (3.49)$$

Similarly we can define discrete Sine transform for u^{xx}_{ijk} (or u^{yy}_{ijk} or u^{zz}_{ijk} or f_{ijk}) in three dimensions and its inverse.

From Eq. (3.44), according to the three dimensional Sine transform, we can rewrite it as

$$\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\hat{u}^{xx}_{klq}\sin(\frac{(i-1)k\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})+$$
$$+10\sum_{k=1}^{M}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\hat{u}^{xx}_{klq}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})+$$
$$\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\hat{u}^{xx}_{klq}\sin(\frac{(i+1)k\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})=$$
$$\frac{12}{h_x^2}\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\Bigg[\hat{u}^{xx}_{klq}\sin(\frac{(i-1)k\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})$$
$$+\frac{12}{h_x^2}\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\hat{u}^{xx}_{klq}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})$$
$$+\frac{12}{h_x^2}\sum_{k=1}^{M-1}\sum_{l=1}^{N-1}\sum_{q=1}^{P-1}\hat{u}^{xx}_{klq}\sin(\frac{(i+1)k\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P})\Bigg], \quad (3.50)$$

from above, we get

$$\hat{u}^{xx}_{klq}=-\hat{u}_{klq}\frac{\frac{24\sin^2(\frac{k\pi}{2M})}{h_x^2}}{\cos(\frac{k\pi}{M})+5}. \quad (3.51)$$

Similarly, from Eqs. (3.45) and (3.46), we can obtain, respectively,

$$\hat{u}^{yy}_{klq}=-\hat{u}_{klq}\frac{\frac{24\sin^2(\frac{l\pi}{2N})}{h_y^2}}{\cos(\frac{l\pi}{N})+5}, \quad (3.52)$$
$$\hat{u}^{zz}_{klq}=-\hat{u}_{klq}\frac{\frac{24\sin^2(\frac{q\pi}{2P})}{h_z^2}}{\cos(\frac{q\pi}{P})+5}. \quad (3.53)$$

In addition, doing three-dimensional Sine transform in Eq. (3.47), we can get

$$-(\hat{u}^{xx}_{klq}+\hat{u}^{yy}_{klq}+\hat{u}^{zz}_{klq})=\hat{f}_{klq}. \quad (3.54)$$

Combining Eqs. (3.51) -(3.54), we can find that

$$-\hat{f}_{klq}=\hat{u}_{klq}\left(\frac{-\frac{24}{h_x^2}\sin^2(\frac{l\pi}{2M})}{\cos(\frac{k\pi}{M})+5}+\frac{-\frac{24}{h_y^2}\sin^2(\frac{l\pi}{2N})}{\cos(\frac{l\pi}{N})+5}+\frac{-\frac{24}{h_z^2}\sin^2(\frac{q\pi}{2P})}{\cos(\frac{q\pi}{P})+5}\right), \quad (3.55)$$

where $\hat{f}_{klq}$ can be calculated from the inverse Sine transform in three dimensions

$$\hat{f}_{klq}=\frac{2}{M}\frac{2}{N}\frac{2}{P}\sum_{i=1}^{M-1}\sum_{j=1}^{N-1}\sum_{p=1}^{P-1}f_{ijp}\sin(\frac{ik\pi}{M})\sin(\frac{jl\pi}{N})\sin(\frac{qp\pi}{P}), \quad (3.56)$$

from which we can obtain u_{klq}. Finally we can do three-dimensional Sine transform of u_{klq} and get the approximation u_{ijk}.

In summary, we get

Algorithm 3 Fast solver for three-dimensional Poisson equation with Dirichlet boundary conditions

1. Given integers M, N, P, and grid points (x_i, y_j, z_p), evaluate f_{ijp} and compute the inverse Sine transform of f_{ijp} to get $\hat{f}_{klq}$.
2. From (3.55), we can obtain $\hat{u}_{klq}$.
3. Compute the discrete sine transform of $\hat{u}_{klq}$ to have u_{ijp}.

We remark that the above algorithm for three-dimensional problem can be extended to solve Poisson equation in d-dimensions ($d \geq 4$).

4 Numerical results

We have constructed a fast solver for Poisson equation in one dimension, two dimensions, three dimensions and four dimensions, respectively. In this section, we apply the proposed numerical algorithms to solve Poisson equation with Dirichlet boundary conditions and test their numerical accuracy. All the computation is conducted on a laptop computer with Intel(R) Core(TM) i3 CPU.

4.1 One-dimensional result

In the first example, we consider the problem with zero Dirichlet boundary conditions

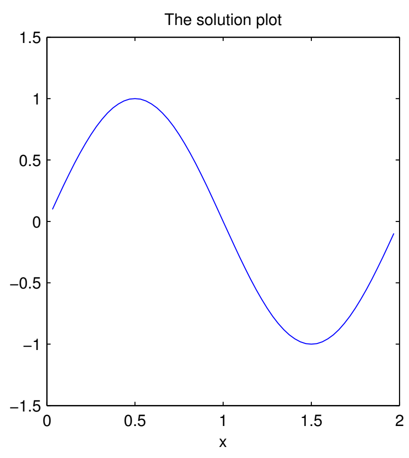
$$\begin{cases} -u''(x) = \pi^2 sin(\pi x), \quad 0 < x < 2, \\ u(0) = 0, \quad u(2) = 0. \end{cases}$$

The exact solution to this problem is $u_{exact}(x) = \sin(\pi x)$. We solve the problem with the numerical algorithm 1 presented in subsection 3.1. From Figure 1, we find that both numerical solution and exact solution at grid points x_i ($i = 1, \cdots, 64-1$) agree very well. The maximum error is less than 4×10^{-7}. In addition, we have done numerical error analysis, which is shown in Table 1, by increasing the number of grid points double. In this table, the error is defined as the discrete L^2 norm, i.e., error=$\sqrt{\sum_{i=1}^{M-1} |u_i - u_{exact}(x_i)|^2 h}$.

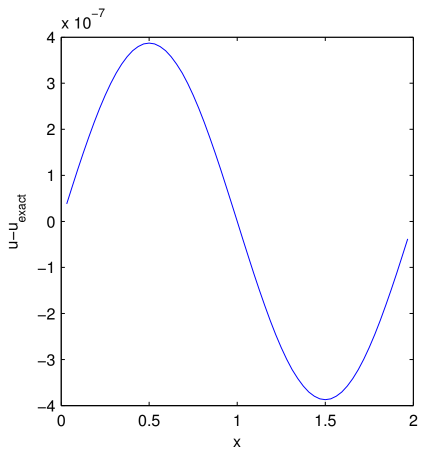
We next aim to solve the problem with nonzero Dirichlet boundary conditions

$$\begin{cases} -u''(x) = 12e^{-x^2}(-x^2 + 1/2), \quad -8 < x < 8, \\ u(-8) = -8, \quad u(8) = 8. \end{cases}$$

The exact solution to this problem is $u_{exact}(x) = 3e^{-x^2} + x$. In the numerical computation, we denote $U(x) = u(x) - x$ using change of variable. Applying nu-



(a)



(b)

Figure 1: (a) Approximated solution of $u(x)$ at grid points; (b)error between approximated solution and exact solution at grid points.

M	8	16	32	64
error	0.0016	9.9699e-5	6.2026e-6	3.8722e-7
rate	-	16.0483	16.0737	16.0183

Table 1: Error analysis for the fourth-order compact finite difference scheme in the first one-dimensional problem.

merical algorithm 1 presented in subsection 3.1, we solve the following problem for $U(x)$

$$\begin{cases} -U''(x) = 12e^{-x^2}(-x^2+1/2), & -8 < x < 8, \\ U(-8)=0, \quad U(8)=0. \end{cases}$$

and get the approximation of $U(x)$ at grid points. Because of $u(x) = U(x) + x$, it will give us the approximated solution of $u(x)$ at grid points, which is shown in Figure 2. Furthermore, Table 2 gives us the error analysis for this problem.

From both Table 1 and Table 2, we find that the error is decreasing 16-fold as grid number M grows doubly, from which we can derive fourth-order accuracy of the method.

M	32	64	128	256	512
error	0.0099	5.7298e-4	3.5205e-5	2.1911e-6	1.3681e-7
rate	-	17.2784	16.2752	16.0674	16.0167

Table 2: Error analysis for the fourth-order compact finite difference scheme in the second one-dimensional problem.

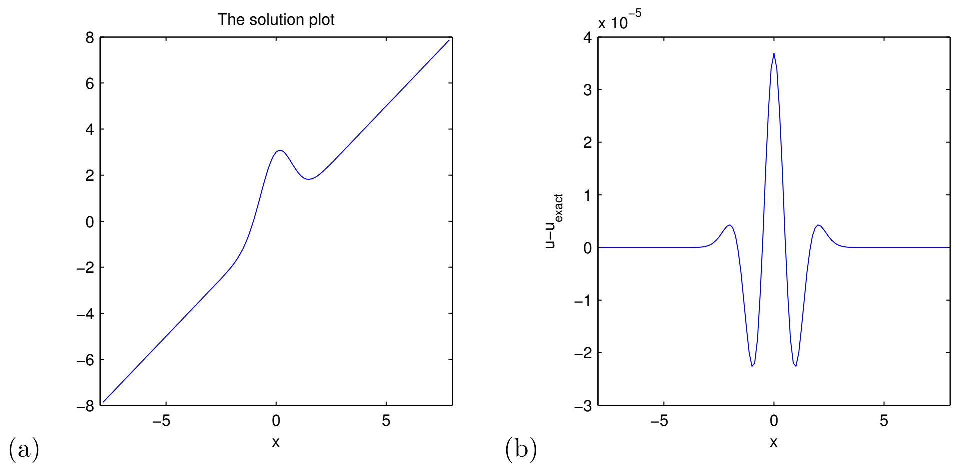


Figure 2: (a) Approximated solution of $u(x)$ at grid points; (b)error between approximated solution and exact solution at grid points.

$M = N$	16	32	64	128
error	0.0012	7.4881e-5	4.6597e-6	2.9091e-7
rate	-	16.0254	16.0699	16.0177

Table 3: Error analysis for the fourth-order compact finite difference scheme in the first two-dimensional problem.

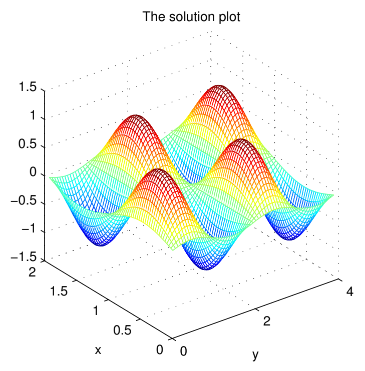
4.2 Two-dimensional result

We first consider the problem with zero Dirichlet boundary conditions

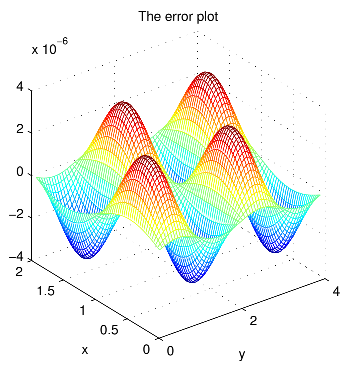
$$\begin{cases} -\Delta u = 2\pi^2 \sin(\pi x)\sin(\pi y) & 0 < x < 2,\ 0 < y < 4 \\ u(x,y) = 0, \quad (x,y) \in \partial D & (D = [0,2] \times [0,4]) \end{cases}$$

The exact solution to the problem is $u_{exact}(x, y) = \sin(\pi x)\sin(\pi y)$. We solve the problem with the numerical algorithm 2 presented in subsection 3.2. In the numerical computation shown in Figure 3, we take $M = N = 64$ and find that both numerical solution and exact solution at those points (x_i, y_j) $(i, j = 1, 2, \cdots, 64-1)$ agree very well. The maximum error is less than 4×10^{-6}. In addition, we have done numerical error analysis as shown in Table 3. In the table, the error is defined as the discrete L^2 norm, i.e.,

$$\text{error} = \sqrt{\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} |u_{ij} - u_{exact}(x_i, y_j)|^2 h_x h_y}.$$



(a)



(b)

Figure 3: (a) Approximated solution $u(x, y)$ at grid points; (b)error between approximated solution and exact solution at grid points (x_i, y_j).

$M = N$	32	64	128	256	512
error	0.0083	4.8115e-4	2.9566e-5	1.8401e-6	1.1489e-7
rate	-	17.2503	16.2738	16.0676	16.0162

Table 4: Error analysis for the fourth-order compact finite difference scheme in the second two-dimensional problem.

We next consider the problem with nonzero Dirichlet boundary conditions

$$\begin{cases} -\Delta u = 12e^{-x^2-2y^2}(-x^2-4y^2+1.5), \quad -8 < x < 8,\ -4 < y < 4, \\ u(x,y) = -12, \quad x = -8, y = -4, \\ u(x,y) = -4, \quad x = -8, y = 4, \\ u(x,y) = 4, \quad x = 8, y = -4, \\ u(x,y) = 12, \quad x = 8, y = 4. \end{cases}$$

The exact solution is $u_{exact}(x, y) = 3e^{-x^2-2y^2} + x + y$. In the numerical computation, we define $U(x, y) = u(x, y) - x - y$, solve the problem for $U(x, y)$ with zero Dirichlet boundary conditions

$$\begin{cases} -\Delta U = 12e^{-x^2-2y^2}(-x^2-4y^2+1.5), \quad -8 < x < 8,\ -4 < y < 4 \\ U(x,y) = 0, \quad (x,y) \in \partial D \quad (D = [-8,8] \times [-4,4]). \end{cases}$$

After we get the approximation of $U(x, y)$ at grid points, because of $u(x, y) = U(x, y) + x + y$, we then can obtain the approximated solution of $u(x, y)$ at grid points. Figure 4 and Table 4 show us the approximated solution, error plot and the error analysis, respectively.

From Table 3 and Table 4, we see that the error is decreasing 16-fold as grid number $M = N$ grows doubly, from which we can know that the method has fourth-order accuracy.

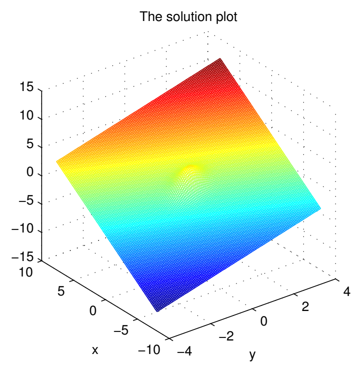
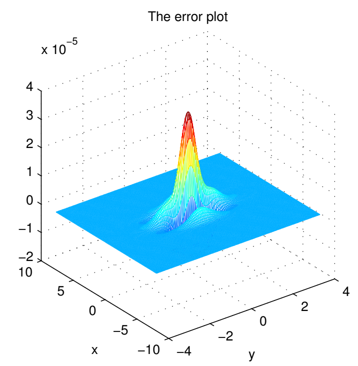
(a)  (b) 

Figure 4: (a) Approximated solution $u(x, y)$ at grid points; (b)error between approximated solution and exact solution at grid points.

4.3 Three-dimensional result

We first solve the problem with zero Dirichlet boundary conditions

$$\begin{cases} -\Delta u = 3\pi^2 \sin(\pi x)\sin(\pi y)\sin(\pi z), \quad 0 < x < 2, 0 < y < 4, 0 < z < 8, \\ u(x, y, z) = 0, \quad (x, y, z) \in \partial D, \end{cases}$$

where $D = [0, 2]\times[0, 4]\times[0, 8]$. The exact solution to this problem is $u_{exact}(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z)$. We solve the problem with the numerical algorithm 3 presented in subsection 3.2. In the numerical computation, we take $M = N = Q = 64$ and find that both numerical solution and exact solution at grid points $(0, y_j, z_p)$ $(j, p = 1, 2, \cdots, 64 - 1)$ agree very well, which is shown in Figure 5. The maximum error is less than 4×10^{-6}. We have done numerical error analysis

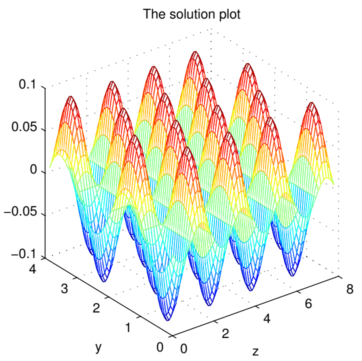
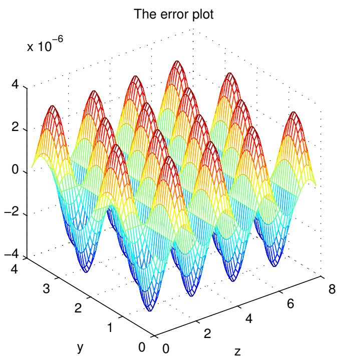
(a)  (b) 

Figure 5: (a) Approximated solution $u(0.5, y, z)$ at grid points ; (b) error between approximated solution and exact solution at grid points $(0.5, y_j, z_p)$.

as shown in Table 5. In this table, the error is defined as the discrete L^2 norm,

$M = N = Q$	32	64	128	256
error	0.0016	1.0020e-4	6.2357e-6	3.8931e-7
rate	-	15.9681	16.0688	16.0173

Table 5: Error analysis for the fourth-order compact finite difference scheme in the first three-dimensional problem.

$M = N = Q$	32	64	128	256
error	0.0071	4.1817e-4	2.5722e-5	1.6012e-006
rate	-	16.9787	16.2573	16.0642

Table 6: Error analysis for the fourth-order compact finite difference scheme in the second three-dimensional problem

i.e., error= $\sqrt{\sum_{i=1}^{M-1}\sum_{j=1}^{N-1}\sum_{p=1}^{Q-1}|u_{ijp} - u_{exact}(x_i, y_j, z_p)|^2 h_x h_y h_z}$.

We next consider the problem with nonzero Dirichlet boundary conditions

$$\begin{cases} -\Delta u = 12e^{-x^2-2y^2-3z^2}(-x^2 - 4y^2 - 9z^2 + 3), \\ \qquad -8 < x < 8,\ -4 < y < 4,\ -4 < z < 4, \\ u(x, y, z) = x + y + z, \quad (x, y, z) \in \quad \partial D. \end{cases}$$

where $D = [-8, 8] \times [-4, 4] \times [-4, 4]$. The exact solution to this problem is $u_{exact}(x) = 3e^{-x^2-2y^2-3z^2} + x + y + z$. In the numerical computation, we let $U(x, y, z) = u(x, y, z) - x - y - z$, solve the problem for $U(x, y, z)$ with zero Dirichlet boundary conditions

$$\begin{cases} -\Delta U = 12e^{-x^2-2y^2-3z^2}(-x^2 - 4y^2 - 9z^2 + 3), \\ \qquad -8 < x < 8,\ -4 < y < 4,\ -4 < z < 4, \\ U(x, y, z) = 0, \quad (x, y, z) \in \partial D. \end{cases}$$

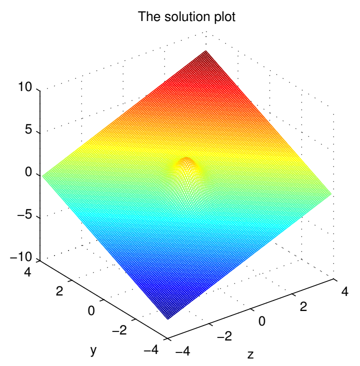
and obtain the approximation of $U(x, y, z)$ at grid points. Finally, using the fact that $u(x, y, z) = U(x, y, z) + x + y + z$ gives us the approximated solution of $u(x, y, z)$ at grid points. Figure 6 and Table 6 respectively show us the approximated solution, error plot and the error analysis for this example.

From Table 5 and Table 6, we observe that the error is decreasing 16-fold as grid number $M = N = P$ grows doubly, from which fourth-order accuracy of the method can be found.

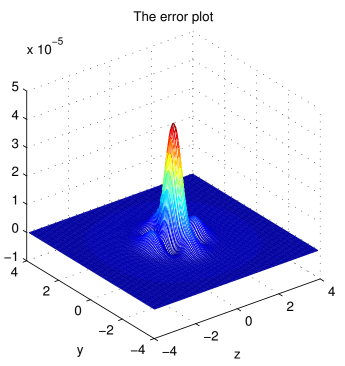
4.4 Four-dimensional result

We first solve the four-dimensional Poisson equation with zero Dirichlet boundary conditions

$$\begin{cases} -\Delta u = 3\pi^2 \sin(\pi x) \sin(\pi y) \sin(\pi z_1) \sin(\pi z_2), \\ \qquad 0 < x < 1, 0 < y < 2, 0 < z_1 < 3, 0 <, z_2 < 4, \\ u(x, y, z_1, z_2) = 0, \quad (x, y, z_1, z_2) \in \partial D, \end{cases}$$



(a)



(b)

Figure 6: (a) Approximated solution $u(0, y, z)$ at grid points; (b)error bet ween approximated solution and exact solution at grid points.

$M = N = P = Q$	8	16	32	64
error	0.0116	6.8524e-4	4.2175e-5	2.6256e-6
rate	-	16.9284	16.2475	16.0630

Table 7: Error analysis for the fourth-order compact finite difference scheme in the first four-dimensional problem.

where $D = [0, 1] \times [0, 2] \times [0, 3] \times [0, 4]$. The exact solution to this problem is $u_{exact}(x, y, z_1, z_2) = \sin(\pi x) \sin(\pi y) \sin(\pi z_1) \sin(\pi z_2)$. We extend the numerical algorithm 3 presented in subsection 3.3 and solve the above problem. Numerical error analysis is shown in Table 7. In the table, the error is defined as the discrete L^2 norm.

We next consider the problem with nonzero Dirichlet boundary conditions

$$\begin{cases} -\Delta u = 4e^{-x^2-2y^2-3z_1^2-4z_2^2}(-x^2 - 4y^2 - 9z_1^2 - 16z_2^2 + 5), \\ \qquad -8 < x < 8,\ -4 < y < 4,\ -4 < z_1 < 4,\ -4 < z_2 < 4, \\ u(x, y, z_1, z_2) = x + y + z_1 + z_2, \quad (x, y, z_1, z_2) \in \partial D. \end{cases}$$

where $D = [-8, 8]\times[-4, 4]\times[-4, 4]\times[-4, 4]$. The exact solution to this problem is $u_{exact}(x, y, z_1, z_2) = e^{-x^2-2y^2-3z_1^2-4z_2^2} + x + y + z_1 + z_2$. In the numerical computation, we let $U(x, y, z_1, z_2) = u(x, y, z_1, z_2) - x - y - z_1 - z_2$, solve the problem for $U(x, y, z_1, z_2)$ with zero Dirichlet boundary conditions

$$\begin{cases} -\Delta U = 12e^{-x^2-2y^2-3z_1^2-4z_2^2}(-x^2 - 4y^2 - 9z_1^2 - 16z_2^2 + 5), \\ \qquad -8 < x < 8,\ -4 < y < 4,\ -4 < z_1 < 4,\ -4 < z_2 < 4 \\ U(x, y, z_1, z_2) = 0, \quad (x, y, z_1, z_2) \in \partial D, \end{cases}$$

and obtain the approximation of $U(x, y, z_1, z_2)$ at grid points. Finally, using the fact that $u(x, y, z_1, z_2) = U(x, y, z_1, z_2)+x+y+z_1+z_2$ gives us the approximated solution of $u(x, y, z_1, z_2)$ at grid points.

$M = N = P = Q$	16	32	64	128
error	0.0759	0.0024	1.4219e-4	8.8607e-06
rate	-	31.6250	16.8788	16.0471

Table 8: Error analysis for the fourth-order compact finite difference scheme in the second four-dimensional problem

From Table 7 and Table 8, we see that the error is decreasing 16-fold as grid number $M = N = P = Q$ grows doubly, from which fourth-order accuracy of the method can be found.

5 Conclusions and discussions

Compact finite difference methods have become popular in the numerical discretization of partial differential equations in recent years. However, when solving higher-dimensional problems the cost of such method becomes larger. We have designed an efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions. The solver is based on fast discrete Sine transform and can be implemented efficiently and easily, as many mathematical software such as Matlab provide the subroutine for fast discrete Sine transform. Our numerical algorithm has been tested and numerical results in one dimension, two dimensions, three dimensions and four dimensions have shown that the proposed four-order compact finite difference scheme for Poisson equation have fourth-order accuracy. The numerical algorithm can be directly extended to solve the d-dimensional ($d \geq 4$) Poisson equation. Although the numerical algorithm presented in the paper mainly for Poisson equation with zero Dirichlet boundary conditions, we have shown that the algorithm can be extended to solve Poisson equation with nonzero Dirichlet boundary conditions, as long as Poisson equation with nonzero Dirichlet boundary conditions can be properly changed into Poisson equation with zero Dirichlet boundary conditions by using technique of change of variables. In addition, if we are faced with Poisson equation with Neumann boundary conditions and still wish to solve the problem with compact finite difference scheme, we may construct a fast solver based on fast discrete Cosine transform. Many compact finite difference schemes which discretize partial differential equations, such as Navier-Stokes equations or Schrödinger equations, are faced with solving Poisson-like equation, our fast solver may be extended to solve the corresponding discretized system as well.

Acknowledgments

The research of H. Wang is supported in part by the Natural Science Foundation of China under grant No. 11261065 and 91430103, and by Ministry of Education Program for New Century Excellent Talents in Chinese Universities under grant No. NCET-13-0995. The second author acknowledges support from the Austrian Science Foundation (FWF) under grant No. F41 (project VICOM), grant No. I830 (project LODIQUAS) and the Austrian Ministry of Science and Research via its grant for the WPI.

References

[1] S. Abarbanel and A. Kumar, Compact high-order schemes for the Euler equations, J. Sci. Comp., 1998, 3:275-288.

[2] M. Ciment and S. Leventhal, Higher order compact implicit schemes for the wave equation, Math. Comp., 1975, 132(29):985-994.

[3] L. Collatz, The numerical treatment of differential equations, Springer-Verlag, New York, 1966, pp. 538.

[4] B. Cockburn and C. Shu, Nonlinearly stable compact schemes for shock calculations, SIAM J. Numer. Anal. , 1994, 31:607-627.

[5] M. Dehghan and A. Taleei, A compact split-step finite difference method for solving the nonlinear Schrodinger equations with constant and variable coefficients, Comp. Phys. Comm., 2010, 181:43-51.

[6] R. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys., 1975, 19:90-109.

[7] J. Kalita, P. Chhabra and S. Kumar, A semi-discrete higher order compact scheme for the unsteady two-dimensional Shrödinger equation, J. Comput. Appl. Math., 2006, 197:141-149.

[8] W. Liao, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation, App. Math. Comp., 2008, 206:755-764.

[9] S. Lele, Compact finite difference scheme with spectral-like resolution, J. Comput. Phys., 1992, 103:16-42.

[10] J. Shen, T. Tang, Spectral and High-Order Methods with Applications, Science Press, Beijing, 2006.

[11] S. Sherer, Scattering of sound from axisymetric sources by multiple circular cylinders, J. Acoust. Soc. Amer., 2003, 115:488-496.

[12] W. Spotz and G. Carey, A high-order compact formulation for the 3D Poisson equation, Num. Meth. Parti. Diff. Equ., 1996, 12: 235-243.

[13] W. Spotz and G. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int. J. Num. Meth. Eng., 1995, 38(20): 3497-3512.

[14] A. Tolstykh, High accuracy non-centered compact difference schemes for fluid dynamics applications, World Scientific, New Jersey, 1994.

[15] M. Visbal and D. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, J. Comput. Phys., 2002, 181:155-185.

[16] S. Xie, G. Li and S. Yi, Compact finite-difference schemes with high accuracy for one-dimensional nonlinear Schrödinger equation, Comput. Methods Appl. Mech. Engrg., 2009, 198:1052-1060.

[17] T. C. Wang, Optimal point-wise error estimate of a compact difference scheme for the coupled Gross-Pitaevskii equations in one dimension, J. Sci. Comput., 2014, 59:158-186.

[18] T. C. Wang , B. L. Guo and Q. B. Xu, Fourth-order compact and energy conservative difference schemes for the nonlinear Schrödinger equation in two dimensions, J. Comput. Phys., 2013,243: 382-399.

[19] T. C. Wang and X. F. Zhao, Optimal l^∞ error estimates of finite difference methods for the coupled Gross-Pitaevskii equation in high dimensions, Sci. China Math., 2014, 57: 2189-2214.

[20] J. Zhang, An explicit fourth-order compact finite difference scheme for three-dimensional convection-diffusion equation, Comm. Num. Meth. Eng., 1998, 14:209-218.

[21] Y. Zhang, Optimal error estimates of compact finite difference discretizations for the Schrödinger-Poisson system, Commun. Comput. Phys., 2013, 13:1357-1388.

[22] X. Zhong, High-Order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition, J. Comput. Phys., 1998, 144:662-709.