

Article

Fourth-Order Compact Finite-Difference Scheme with Discrete Sine Transform for Solving 2D Heat Conduction Equation with DBCs

Chunming Liu ¹ and Xiaozhong Tong ^{1,2,*} 

¹ School of Geosciences and Info-Physics, Central South University, Changsha 410083, China; liuchunming@csu.edu.cn

² Key Laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring (Ministry of Education), Central South University, Changsha 410083, China

* Correspondence: csumaysnow@csu.edu.cn

Abstract

Finite-difference approaches are widely employed to solve partial differential equations in numerous practical applications. However, their computational efficiency is often limited by the need to solve linear systems through matrix inversion or iterative solvers, a challenge that is particularly acute in high-dimensional problems. Consequently, there is a growing demand for methods that ensure both high accuracy and computational efficiency. To address the two-dimensional (2D) heat conduction problem, we propose a novel hybrid technique that integrates a fourth-order implicit compact finite-difference approach with the discrete sine transform (DST). The incorporation of the DST significantly reduces the computational burden associated with solving the heat conduction equation on large grids. Detailed numerical experiments were conducted to evaluate this solver for 2D heat conduction equations subject to homogeneous Dirichlet boundary conditions (DBC). The results demonstrate that the proposed method not only achieves substantial reductions in computational cost but also maintains a high level of numerical accuracy. All numerical experiments were performed on a computer running MATLAB R2024b.

Keywords: heat conduction equation; Dirichlet boundary conditions (DBC); implicit finite-difference; fourth-order compact scheme; discrete sine transform (DST)

MSC: 86A25



Academic Editors: Ching-Sung Liu and Xiaobing Feng

Received: 5 February 2026

Revised: 2 March 2026

Accepted: 10 March 2026

Published: 11 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In the fields of mathematics and physics, the heat conduction equation is a significant time-dependent parabolic partial differential equation (PDE). It is commonly employed to model the distribution of temperature or heat within a specified region as it changes over time [1]. Transient heat conduction problems involve variables in both the spatial domain and the time domain. Numerical methods for approximating solutions to the heat conduction equation are essential for addressing a wide range of mathematical and physical problems. Currently, the spatial-domain discretization methods primarily include the finite-difference approach [2,3], the finite-element scheme [4–6], the finite-volume technique [7–9], the boundary-element method [10–12], the meshless method [13–15], and the Fourier or Chebyshev spectral method [16,17]. For discretizing the time-domain variable, the time derivative is typically approximated using a finite difference [18]. These

numerical approaches often rely on different discretization schemes in the spatial domain. However, solving large-scale systems of equations is typically required when addressing complex numerical models. Furthermore, the computational cost increases substantially when tackling higher-order heat conduction problems.

The finite-difference method (FDM) approximates partial derivatives in partial differential equations (PDEs) using differential operators, thereby enabling the computation of numerical solutions through the resulting linear system [19]. As a specific variant, the finite-difference, time-domain (FDTD) approach discretizes both space and time, applying finite-difference approximations to derivatives in each domain. This method offers notable advantages in efficiency due to its straightforward formulation, conceptual simplicity, and ease of implementation [20]. Despite its widespread use, the conventional FDM exhibits significant limitations in computational precision. A common approach to enhancing accuracy involves mesh refinement; however, this strategy inevitably increases storage requirements and computational effort, particularly in high-dimensional scenarios. Moreover, solving the resulting large linear or nonlinear systems via direct matrix inversion or iterative solvers incurs substantial computational costs. These challenges have motivated the exploration of alternative techniques to achieve more efficient solutions. Consequently, a crucial objective for improving the computational efficiency of the FDM is to avoid the computational burden associated with large-scale matrix inversion. Significant research efforts have been directed toward addressing these limitations, with notable advancements reported in improving the efficiency and accuracy of finite-difference approaches for solving heat equations in two- or three-dimensional spaces [21–24].

Fourier-related transforms play a fundamental role in solving PDEs [25–28]. Many PDE problems, including Poisson equations with constant or variable coefficients, can be efficiently solved using these transforms. Based on the signal's time continuity, discreteness, and periodicity, Fourier-related transforms are generally categorized into two types: the continuous-time Fourier transform (CFT) and the discrete-time Fourier transform (DTFT). The rapid advancement of digital technology, particularly in computer hardware and software, has provided abundant resources for discrete signal processing. Notably, the fast Fourier transform (FFT) has emerged as a prevalent tool for computing the DTFT, which calculates the discrete-time Fourier series for periodic signals. Fourier-related transforms are particularly effective for solving the Poisson equation under various boundary conditions, including periodic, Dirichlet, Neumann, and Robin types [29]. When combined with the finite-difference method (FDM), these transforms enable the derivation of solutions through distinct discrete Fourier transformations. For example, the discrete sine transform (DST) is well-suited for problems with homogeneous Dirichlet boundary conditions (DBC), while the discrete cosine transform (DCT) is applicable to Neumann boundary conditions [30]. These transforms have been previously utilized to efficiently solve the linear systems arising from finite-difference discretizations of PDEs, offering an attractive alternative to direct matrix inversion or iterative solvers [31,32].

The paper aims to develop a novel hybrid technique that integrates a fourth-order implicit compact finite-difference approach with the DST. To demonstrate our proposed methodology, we focus the analysis on the 2D heat conduction equation governed by DBCs on rectangular domains. The inherent nature of DBCs eliminates specialized approximation formulas near boundaries, thereby ensuring the validity of the method after boundary treatment. The technical foundations are documented in references [33–35]. The formalism of the proposed method is also highly similar for parabolic equations.

The remainder of the paper is organized as follows. Section 2 presents a detailed formulation of the fourth-order compact finite-difference scheme. Section 3 explains the development and implementation of the DST-accelerated compact finite-difference algorithm

for 2D heat conduction equations with homogeneous DBCs. Section 4 presents numerical experiments demonstrating the application of the proposed DST-accelerated method to 2D heat conduction equations. Finally, Section 5 summarizes the main conclusions.

2. Fourth-Order Compact Finite-Difference Scheme

The second-order derivative is approximated using the following centered compact finite-difference scheme [36]:

$$\beta u''_{i-2} + \alpha u''_{i-1} + u''_i + \alpha u''_{i+1} + \beta u''_{i+2} = c \frac{u_{i-3} - 2u_i + u_{i+3}}{9(\Delta x)^2} + b \frac{u_{i-2} - 2u_i + u_{i+2}}{4(\Delta x)^2} + a \frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2} \quad (1)$$

where u''_i denotes the approximation of the second-order derivative at node i , and Δx represents the spatial mesh size along the x -direction. The relations among the coefficients a , b , c , α , and β are established by aligning the coefficients of the Taylor series across different orders. Specifically, these coefficients are computed by employing the Taylor series expansions of the components in Equation (1):

$$\begin{cases} u''_{i+1} = u''(x_i + \Delta x) = u''_i + \Delta x \cdot u_i^{(3)} + \frac{(\Delta x)^2}{2!} u_i^{(4)} + \frac{(\Delta x)^3}{3!} u_i^{(5)} + \frac{(\Delta x)^4}{4!} u_i^{(6)} \\ u''_{i-1} = u''(x_i - \Delta x) = u''_i - \Delta x \cdot u_i^{(3)} + \frac{(\Delta x)^2}{2!} u_i^{(4)} - \frac{(\Delta x)^3}{3!} u_i^{(5)} + \frac{(\Delta x)^4}{4!} u_i^{(6)} \\ u''_{i+2} = u''(x_i + 2\Delta x) = u''_i + 2\Delta x \cdot u_i^{(3)} + \frac{2^2(\Delta x)^2}{2!} u_i^{(4)} + \frac{2^3(\Delta x)^3}{3!} u_i^{(5)} + \frac{2^4(\Delta x)^4}{4!} u_i^{(6)} \\ u''_{i-2} = u''(x_i - 2\Delta x) = u''_i - 2\Delta x \cdot u_i^{(3)} + \frac{2^2(\Delta x)^2}{2!} u_i^{(4)} - \frac{2^3(\Delta x)^3}{3!} u_i^{(5)} + \frac{2^4(\Delta x)^4}{4!} u_i^{(6)} \end{cases} \quad (2)$$

and

$$\begin{cases} u_{i+1} = u(x_i + \Delta x) = u_i + \Delta x \cdot u'_i + \frac{(\Delta x)^2}{2!} u''_i + \frac{(\Delta x)^3}{3!} u_i^{(3)} + \frac{(\Delta x)^4}{4!} u_i^{(4)} \\ u_{i-1} = u(x_i - \Delta x) = u_i - \Delta x \cdot u'_i + \frac{(\Delta x)^2}{2!} u''_i - \frac{(\Delta x)^3}{3!} u_i^{(3)} + \frac{(\Delta x)^4}{4!} u_i^{(4)} \\ u_{i+2} = u(x_i + 2\Delta x) = u_i + 2\Delta x \cdot u'_i + \frac{2^2(\Delta x)^2}{2!} u''_i + \frac{2^3(\Delta x)^3}{3!} u_i^{(3)} + \frac{2^4(\Delta x)^4}{4!} u_i^{(4)} \\ u_{i-2} = u(x_i - 2\Delta x) = u_i - 2\Delta x \cdot u'_i + \frac{2^2(\Delta x)^2}{2!} u''_i - \frac{2^3(\Delta x)^3}{3!} u_i^{(3)} + \frac{2^4(\Delta x)^4}{4!} u_i^{(4)} \\ u_{i+3} = u(x_i + 3\Delta x) = u_i + 3\Delta x \cdot u'_i + \frac{3^2(\Delta x)^2}{2!} u''_i + \frac{3^3(\Delta x)^3}{3!} u_i^{(3)} + \frac{3^4(\Delta x)^4}{4!} u_i^{(4)} \\ u_{i-3} = u(x_i - 3\Delta x) = u_i - 3\Delta x \cdot u'_i + \frac{3^2(\Delta x)^2}{2!} u''_i - \frac{3^3(\Delta x)^3}{3!} u_i^{(3)} + \frac{3^4(\Delta x)^4}{4!} u_i^{(4)} \end{cases} \quad (3)$$

Substituting Equations (2) and (3) into Equation (1) and rearranging yields

$$\begin{cases} a + b + c = 1 + 2\alpha + 2\beta \\ a + 2^2b + 3^2c = \frac{4!}{2!}(\alpha + 2^2\beta) \\ a + 2^4b + 3^4c = \frac{6!}{4!}(\alpha + 2^4\beta) \end{cases} \quad (4)$$

Different subsystems can be constructed by selecting some or all of the equations from Equation (4), which enables the determination of the coefficients a , b , c , α , and β .

In this paper, we focus on the system derived from the first two equations of Equation (4). With five unknowns present, we can assign arbitrary values to two of the variables and subsequently determine the others using the Gaussian elimination method. For simplicity, we take $\beta = 0 = b = c$, then we can obtain $\alpha = \frac{1}{10}$, and $a = \frac{6}{5}$. Consequently, we can obtain the following fourth-order accurate compact finite-difference scheme:

$$\frac{1}{10} u''_{i-1} + u''_i + \frac{1}{10} u''_{i+1} = \frac{6}{5} \frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2} \quad (5)$$

3. Numerical Algorithm

3.1. An Implicit Fourth-Order Compact Finite-Difference Scheme for 2D Heat Conduction Equation

Consider the 2D heat conduction problem within the rectangular domain $[0, a] \times [0, b]$ defined as follows:

$$\begin{cases} \frac{\partial u}{\partial t} = \kappa \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f(x, y, t), & 0 < x < a, 0 < t < T \\ u(0, y, t) = u(a, y, t) = 0, & 0 \leq x \leq b, 0 \leq t \leq T \\ u(x, 0, t) = u(x, b, t) = 0, & 0 \leq x \leq a, 0 \leq t \leq T \\ u(x, y, 0) = \varphi(x, y), & 0 \leq x \leq a, 0 \leq y \leq b \end{cases} \quad (6)$$

where $u(x, y, t)$ denotes the unknown temperature distribution, κ is the positive coefficient called the thermal diffusivity, $f(x, y, t)$ represents the thermal source term, and $\varphi(x, y)$ is the initial temperature distribution. The computational domain $[0, a] \times [0, b] \times [0, T]$ is divided into an $N_x \times N_y \times N_t$ mesh with uniform spatial step sizes $\Delta x = a/N_x$ and $\Delta y = b/N_y$ in the x - and y -directions, respectively, and a time step size $\Delta t = T/N_t$. The computational grid points (x_i, y_j, t_p) are defined by

$$\begin{aligned} x_i &= i\Delta x, & i &= 0, 1, 2, \dots, N_x \\ y_j &= j\Delta y, & j &= 0, 1, 2, \dots, N_y \\ t_p &= p\Delta t, & p &= 0, 1, 2, \dots, N_t \end{aligned} \quad (7)$$

in which N_x , N_y and N_t are positive integers.

We use $u(x_i, y_j, t_p)$ to denote the approximated solution at the grid point (x_i, y_j) and time level $p\Delta t$. Considering the fourth-order compact finite-difference scheme from Equation (5), we can derive the following approximations:

$$\frac{1}{10}u_{xx}(x_{i-1}, y_j, t_p) + u_{xx}(x_i, y_j, t_p) + \frac{1}{10}u_{xx}(x_{i+1}, y_j, t_p) = \frac{6}{5} \frac{u(x_{i-1}, y_j, t_p) - 2u(x_i, y_j, t_p) + u(x_{i+1}, y_j, t_p)}{(\Delta x)^2} \quad (8)$$

$$\frac{1}{10}u_{yy}(x_i, y_{j-1}, t_p) + u_{yy}(x_i, y_j, t_p) + \frac{1}{10}u_{yy}(x_i, y_{j+1}, t_p) = \frac{6}{5} \frac{u(x_i, y_{j-1}, t_p) - 2u(x_i, y_j, t_p) + u(x_i, y_{j+1}, t_p)}{(\Delta y)^2} \quad (9)$$

where $i = 2, 3, \dots, N_x$, $j = 2, 3, \dots, N_y$, and $p = 1, 2, \dots, N_t + 1$.

Using the Kronecker tensor product $\otimes$, Equations (8) and (9) can be reformulated in the following matrix form:

$$A_x \otimes I \cdot u_{xx} = \frac{6}{5(\Delta x)^2} B_x \otimes I \cdot u \quad (10)$$

$$I \otimes A_y \cdot u_{yy} = \frac{6}{5(\Delta y)^2} I \otimes B_y \cdot u \quad (11)$$

where $u = \left(u_{11}, \dots, u_{1, N_x-1}, u_{21}, \dots, u_{2, N_x-1}, \dots, u_{N_y-1, 1}, \dots, u_{N_y-1, N_x-1} \right)^T$ is the solution vector at the interior mesh nodes, and u_{xx} and u_{yy} are defined as the approximations of the second-order partial derivatives $u_{xx}(x_i, y_j, t_p)$ and $u_{yy}(x_i, y_j, t_p)$, respectively, I is the identity matrix, and $A_x, B_x \in \mathbb{R}^{(N_x-1) \times (N_x-1)}$, $A_y, B_y \in \mathbb{R}^{(N_y-1) \times (N_y-1)}$ with A_x, A_y and B_x, B_y taking the general form of A and B , respectively:

$$A = \begin{pmatrix} 1 & \frac{1}{10} & 0 & \cdots & 0 & 0 \\ \frac{1}{10} & 1 & \frac{1}{10} & \cdots & 0 & 0 \\ 0 & \frac{1}{10} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{1}{10} \\ 0 & 0 & 0 & \cdots & \frac{1}{10} & 1 \end{pmatrix}, \quad B = \begin{pmatrix} -\frac{12}{5} & \frac{6}{5} & 0 & \cdots & 0 & 0 \\ \frac{6}{5} & -\frac{12}{5} & \frac{6}{5} & \cdots & 0 & 0 \\ 0 & \frac{6}{5} & -\frac{12}{5} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{12}{5} & \frac{6}{5} \\ 0 & 0 & 0 & \cdots & \frac{6}{5} & -\frac{12}{5} \end{pmatrix} \quad (12)$$

In addition, setting

$$A_1 = A_x \otimes I, A_2 = I \otimes A_y,$$

$$B_1 = B_x \otimes I, B_2 = I \otimes B_y.$$

Equations (10) and (11) can be written as

$$A_1 u_{xx} = \frac{6}{5(\Delta x)^2} B_1 u \quad (13)$$

$$A_2 u_{yy} = \frac{6}{5(\Delta y)^2} B_2 u \quad (14)$$

from which it follows that

$$u_{xx} = \frac{6}{5(\Delta x)^2} A_1^{-1} B_1 u \quad (15)$$

$$u_{yy} = \frac{6}{5(\Delta y)^2} A_2^{-1} B_2 u \quad (16)$$

Using the Crank–Nicolson implicit scheme for the 2D heat conduction equation in Equation (6), we have

$$\frac{u^{p+1} - u^p}{\Delta t} = \kappa \left(\frac{u_{xx}^p + u_{yy}^p}{2} + \frac{u_{xx}^{p+1} + u_{yy}^{p+1}}{2} \right) + \frac{f^p + f^{p+1}}{2} \quad (17)$$

where $u^p = u(x_i, y_j, t_p)$ and $f^p = f(x_i, y_j, t_p)$. Combining Equations (15)–(17), we have

$$\begin{aligned} \frac{u^{p+1} - u^p}{\Delta t} &= \kappa \left[\frac{3}{5(\Delta x)^2} A_1^{-1} B_1 + \frac{3}{5(\Delta y)^2} A_2^{-1} B_2 \right] u^p \\ &\quad + \kappa \left[\frac{3}{5(\Delta x)^2} A_1^{-1} B_1 + \frac{3}{5(\Delta y)^2} A_2^{-1} B_2 \right] u^{p+1} + \frac{f^p + f^{p+1}}{2} \end{aligned}$$

Having defined $M = \frac{3}{5(\Delta x)^2} A_1^{-1} B_1 + \frac{3}{5(\Delta y)^2} A_2^{-1} B_2$, the resulting linear system is obtained as follows

$$(I - \kappa \Delta t \cdot M) u^{p+1} = (I + \kappa \Delta t \cdot M) u^p + \Delta t \frac{f^p + f^{p+1}}{2} \quad (18)$$

Solving the linear system given by Equation (18) yields the numerical solution for the temperature distribution at the next time level.

3.2. Accelerated Strategy with DST

Traditional matrix inversion methods for solving Equation (18) face significant computational challenges when dealing with fine meshes. To overcome this limitation, we propose a more efficient approach that utilizes the DST in the x - and y -directions, which provides an accelerated solution method for the fourth-order compact finite-difference formulation.

It is noteworthy that the DST for $u(x_i, y_j, t_p)$ and the inverse DST for $\tilde{u}(k_i, k_j, t_p)$ are defined as follows

$$u(x_i, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (19)$$

$$\tilde{u}(k_i, k_j, t_p) = \frac{2}{N_x} \frac{2}{N_y} \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} u(x_i, y_j, t_p) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (20)$$

In addition, we note that

$$u(x_{i-1}, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \sin\left(\frac{(i-1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (21)$$

$$u(x_{i+1}, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \sin\left(\frac{(i+1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (22)$$

and

$$u_{xx}(x_i, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{xx}(k_i, k_j, t_p) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (23)$$

$$u_{xx}(x_{i-1}, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{xx}(k_i, k_j, t_p) \sin\left(\frac{(i-1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (24)$$

$$u_{xx}(x_{i+1}, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{xx}(k_i, k_j, t_p) \sin\left(\frac{(i+1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (25)$$

Substituting them into Equation (5) yields

$$\begin{aligned} & \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{xx}(k_i, k_j, t_p) \left\{ \frac{1}{10} \sin\left(\frac{(i-1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) + \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \right. \\ & \quad \left. + \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) + \frac{1}{10} \sin\left(\frac{(i+1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \right\} \\ &= \frac{6}{5(\Delta x)^2} \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \left\{ \sin\left(\frac{(i-1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) - 2 \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \right. \\ & \quad \left. + \sin\left(\frac{(i+1)k_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \right\} \end{aligned}$$

Applying elementary trigonometry, we have a simple form:

$$\begin{aligned} & \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{xx}(k_i, k_j, t_p) \left(\frac{1}{5} \cos\left(\frac{k_i\pi}{N_x}\right) + 1 \right) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \\ &= \frac{6}{5(\Delta x)^2} \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \left(2 \cos\left(\frac{k_i\pi}{N_x}\right) - 2 \right) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \end{aligned}$$

Finally, we have

$$\tilde{u}_{xx}(k_i, k_j, t_p) = \frac{\frac{6}{(\Delta x)^2} \left(2 \cos\left(\frac{k_i\pi}{N_x}\right) - 2 \right)}{\cos\left(\frac{k_i\pi}{N_x}\right) + 5} \tilde{u}(k_i, k_j, t_p)$$

Here, we define the modified wavenumber $\lambda_{xi} = \frac{6}{(\Delta x)^2} \frac{2 \cos\left(\frac{k_i \pi}{N_x}\right) - 2}{\cos\left(\frac{k_i \pi}{N_x}\right) + 5}$, then

$$\tilde{u}_{xx}(k_i, k_j, t_p) = \lambda_{xi} \cdot \tilde{u}(k_i, k_j, t_p) \quad (26)$$

Similarly, we know that

$$u(x_i, y_{j-1}, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \sin\left(\frac{ik_i \pi}{N_x}\right) \sin\left(\frac{(j-1)k_j \pi}{N_y}\right) \quad (27)$$

$$u(x_i, y_{j+1}, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}(k_i, k_j, t_p) \sin\left(\frac{ik_i \pi}{N_x}\right) \sin\left(\frac{(j+1)k_j \pi}{N_y}\right) \quad (28)$$

and

$$u_{yy}(x_i, y_j, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{yy}(k_i, k_j, t_p) \sin\left(\frac{ik_i \pi}{N_x}\right) \sin\left(\frac{jk_j \pi}{N_y}\right) \quad (29)$$

$$u_{yy}(x_i, y_{j-1}, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{yy}(k_i, k_j, t_p) \sin\left(\frac{ik_i \pi}{N_x}\right) \sin\left(\frac{(j-1)k_j \pi}{N_y}\right) \quad (30)$$

$$u_{yy}(x_i, y_{j+1}, t_p) = \sum_{k_i=1}^{N_x-1} \sum_{k_j=1}^{N_y-1} \tilde{u}_{yy}(k_i, k_j, t_p) \sin\left(\frac{ik_i \pi}{N_x}\right) \sin\left(\frac{(j+1)k_j \pi}{N_y}\right) \quad (31)$$

Substituting these into Equation (5) will yield

$$\tilde{u}_{yy}(k_i, k_j, t_p) = \lambda_{yj} \cdot \tilde{u}(k_i, k_j, t_p) \quad (32)$$

$$\text{where } \lambda_{yj} = \frac{6}{(\Delta y)^2} \frac{2 \cos\left(\frac{k_j \pi}{N_y}\right) - 2}{\cos\left(\frac{k_j \pi}{N_y}\right) + 5}.$$

Applying 2D DST in the x -direction and y -direction, Equation (17) can be written as

$$\begin{aligned} & \frac{\tilde{u}(k_i, k_j, t_{p+1}) - \tilde{u}(k_i, k_j, t_p)}{\Delta t} \\ &= \kappa \cdot \left[\frac{\tilde{u}_{xx}(k_i, k_j, t_p) + \tilde{u}_{xx}(k_i, k_j, t_{p+1})}{2} + \frac{\tilde{u}_{yy}(k_i, k_j, t_p) + \tilde{u}_{yy}(k_i, k_j, t_{p+1})}{2} \right] \\ & \quad + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \\ &= \kappa \cdot \left[\lambda_{xi} \frac{\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})}{2} + \lambda_{yj} \frac{\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})}{2} \right] \\ & \quad + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \\ &= \kappa \cdot \frac{\lambda_{xi} + \lambda_{yj}}{2} [\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})] + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \\ & \quad \frac{\tilde{u}(k_i, k_j, t_{p+1}) - \tilde{u}(k_i, k_j, t_p)}{\Delta t} \\ &= \kappa \cdot \left[\frac{\tilde{u}_{xx}(k_i, k_j, t_p) + \tilde{u}_{xx}(k_i, k_j, t_{p+1})}{2} + \frac{\tilde{u}_{yy}(k_i, k_j, t_p) + \tilde{u}_{yy}(k_i, k_j, t_{p+1})}{2} \right] \\ & \quad + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \\ &= \kappa \cdot \left[\lambda_{xi} \frac{\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})}{2} + \lambda_{yj} \frac{\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})}{2} \right] \\ & \quad + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \\ &= \kappa \cdot \frac{\lambda_{xi} + \lambda_{yj}}{2} [\tilde{u}(k_i, k_j, t_p) + \tilde{u}(k_i, k_j, t_{p+1})] + \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2} \end{aligned}$$

where $\tilde{f}(k_i, k_j, t_p)$ can be calculated from

$$\tilde{f}(k_i, k_j, t_p) = \frac{2}{N_x} \frac{2}{N_y} \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} f(x_i, y_j, t_p) \sin\left(\frac{ik_i\pi}{N_x}\right) \sin\left(\frac{jk_j\pi}{N_y}\right) \quad (33)$$

Finally, we have

$$\tilde{u}(k_i, k_j, t_{p+1}) = \frac{2 + \kappa\Delta t(\lambda_{xi} + \lambda_{yj})}{2 - \kappa\Delta t(\lambda_{xi} + \lambda_{yj})} \tilde{u}(k_i, k_j, t_p) + \Delta t \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2 - \kappa\Delta t(\lambda_{xi} + \lambda_{yj})} \quad (34)$$

Thus, from Equation (34), we can obtain $\tilde{u}(k_i, k_j, t_{p+1})$. Additionally, using the 2D inverse DST calculation, we can derive the temperature distribution $u(x_i, y_j, t_p)$ in the spatial domain, for all $i = 2, 3, \dots, N_x$, $j = 2, 3, \dots, N_y$, and $p = 1, 2, \dots, N_t + 1$, as defined in Equation (6).

4. Numerical Experiments and Discussion

To illustrate the application of the fourth-order compact finite-difference scheme with DST acceleration for solving the two-dimensional heat conduction equation subject to homogeneous DBCs, we present two numerical examples in this section that demonstrate the accuracy and efficiency of the proposed method. For our numerical experiments, we set $N_x = N_y = N$ and then define the mesh size $\Delta x = \Delta y$. We calculate both the maximum absolute error and the relative root-mean-square error, which is defined using the discrete L^2 norm as follows [37]:

$$E = \frac{\sqrt{\sum_{i=1}^{N_x+1} \sum_{j=1}^{N_y+1} (u_{i,j}^p - \tilde{u}_{i,j}^p)^2}}{\sqrt{\sum_{i=1}^{N_x+1} \sum_{j=1}^{N_y+1} (\tilde{u}_{i,j}^p)^2}} \times 100 \% \quad (35)$$

where $u_{i,j}^p$ and $\tilde{u}_{i,j}^p$ denote the 2D approximated solution and analytical solutions at the point (x_i, y_j) with $t = t_p$, respectively.

Additionally, the asymptotic convergence rate is estimated by [38]:

$$\text{Order} := \frac{\log(E_{N_1}/E_{N_2})}{\log(N_1/N_2)} \quad (36)$$

where N_1 and N_2 represent distinct numbers of discrete meshes.

4.1. Example 1: 2D Homogeneous Heat Conduction Equation

In this numerical example, we consider the 2D homogeneous heat conduction equation subject to homogeneous DBCs:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{1}{8\pi^2} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), & 0 < x < 1, 0 < y < 1, 0 < t \leq 1 \\ u(0, t) = u(1, t) = 0, & 0 \leq t \leq 1 \\ u(x, y, 0) = \sin(2\pi x) \sin(2\pi y), & 0 \leq x \leq 1, 0 \leq y \leq 1 \end{cases}$$

The analytical solution of this problem is known as $u(x, y, t) = \sin(2\pi x) \sin(2\pi y) e^{-t}$. Obviously, the analytical solution $u(x, y, t)$ becomes very small when t is large. Its temperature distribution at $t = 1$ is shown in Figure 1.

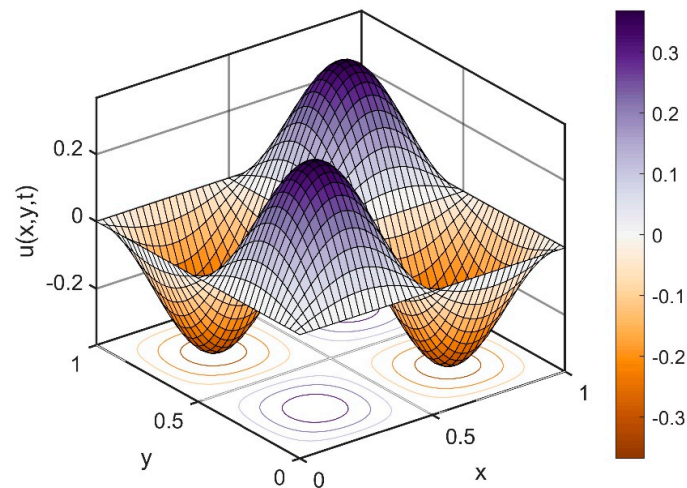


Figure 1. The temperature distribution of the 2D heat conduction homogeneous equation with homogeneous DBCs at $t = 1$.

In our computation, we take 41 equally spaced sampling points along the x - and y -directions, respectively, and the time step is taken as 0.001. The numerical results show excellent agreement between the DST-CNCFD numerical solution and the corresponding analytical solution for $t = 1$ at all discrete nodes (x_i, y_j) ($i, j = 1, 2, \dots, 41$), as depicted in Figure 2. For this test case, a maximum absolute error of 9.03×10^{-7} is achieved.

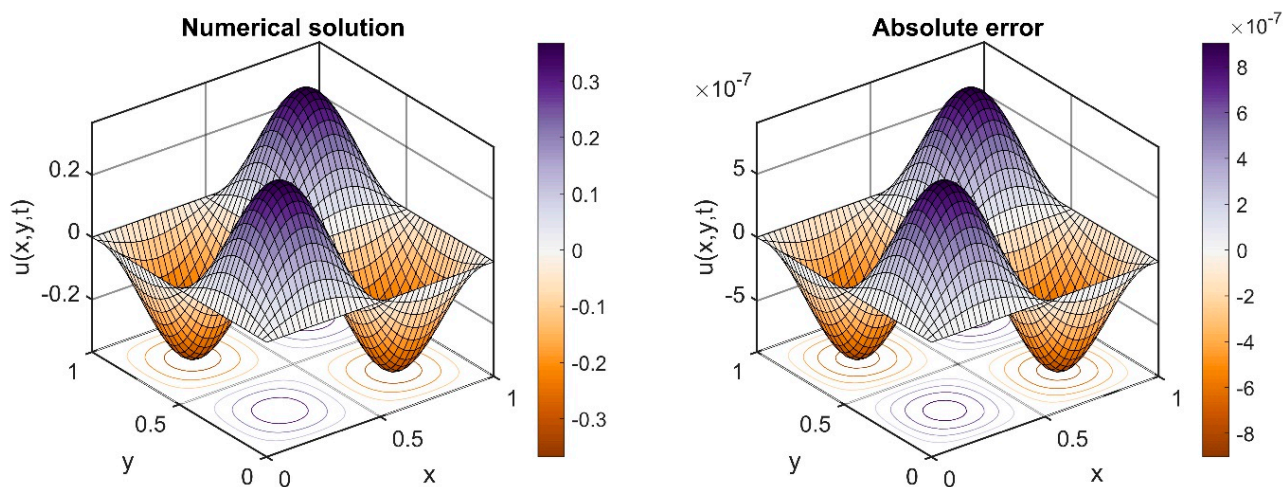


Figure 2. DST-CNCFD numerical solution for the 2D heat conduction homogeneous equation subject to homogeneous DBCs with 41×41 at $t = 1$.

Moreover, the numerical error analysis for this 2D heat conduction equation has been conducted. We computed this problem with the present DST-CNCFD solver and compared the results with the traditional CNFD method [39]. Table 1 presents the numerical results with $\Delta t = 0.001$, $N = 10, 20, 40, 80$, and $t = 1$. The results confirm that fourth-order accuracy is achieved. Refining the spatial mesh in this numerical experiment leads to a corresponding decrease in the relative root-mean-square error, thereby confirming the stability of both the proposed scheme and the conventional CNFD method. Furthermore, the DST-CNCFD method exhibits higher numerical accuracy compared to the conventional CNFD approach. These numerical findings demonstrate that our DST-CNCFD technique achieves high accuracy for solving the two-dimensional heat conduction equation subject to homogeneous DBCs.

Table 1. Relative root-mean-square error of our proposed DST-CNCFD scheme and the traditional CNFD scheme for Example 1.

Mesh Size	DST-CNCFD Scheme		Traditional CNFD Scheme	
	Error (Per Cent)	Order	Error (Per Cent)	Order
10×10	6.59×10^{-2}	—	3.30	—
20×20	4.06×10^{-3}	4.021	8.23×10^{-1}	2.004
40×40	2.46×10^{-4}	4.045	2.06×10^{-1}	1.998
80×80	7.52×10^{-6}	4.008	5.14×10^{-2}	2.003
160×160	7.32×10^{-6}	4.003	1.28×10^{-2}	2.006

Furthermore, we compare the computational efficiency of the proposed method with that of the traditional CNFD method in Table 2. From this table, it is evident that the DST-CNCFD method requires significantly less computational time than the mesh $N_x = N_y = 160$ is refined. The traditional CNFD method, however, requires additional computational time to solve the 2D heat conduction equation.

Table 2. Comparison of computational time between the DST-CNCFD scheme and the traditional CNFD technique for the 2D heat conduction equation.

Mesh Size	DST-CNCFD Scheme	Traditional CNFD Scheme
10×10	0.03 s	2.15 s
20×20	0.06 s	2.65 s
40×40	0.18 s	7.28 s
80×80	0.30 s	54.14 s
160×160	0.99 s	278.19 s

4.2. Example 2: 2D Non-Homogeneous Heat Conduction Equation

In this numerical example, the DST-CNCFD solver is employed to solve the 2D non-homogeneous heat conduction equation subject to homogeneous DBCs. The 2D initial-boundary value problem is formulated as follows:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{1}{\pi^2} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \sin(\pi x) \sin(\pi y) e^{-t}, & 0 < x < 1, 0 < t < 1 \\ u(0, t) = u(1, t) = 0, & 0 \leq t \leq 1 \\ u(x, y, 0) = \sin(\pi x) \sin(\pi y), & 0 \leq x \leq 1 \end{cases}$$

The analytical solution to this problem is known as $u(x, y, t) = \sin(\pi x) \sin(\pi y) e^{-t}$. Its temperature distribution at $t = 1$ is shown in Figure 3.

In our computation, we take 41 equally spaced sampling points along the x - and y -directions, respectively, and the time step is taken as 0.001. The results demonstrate excellent agreement between the numerical solutions obtained using the DST-CNCFD solver and the analytical solutions for time $t = 1$ at all discrete grid points $(x_i, y_j) (i, j = 1, 2, \dots, 41)$, as depicted in Figure 4. A maximum absolute error of 5.44×10^{-8} is achieved.

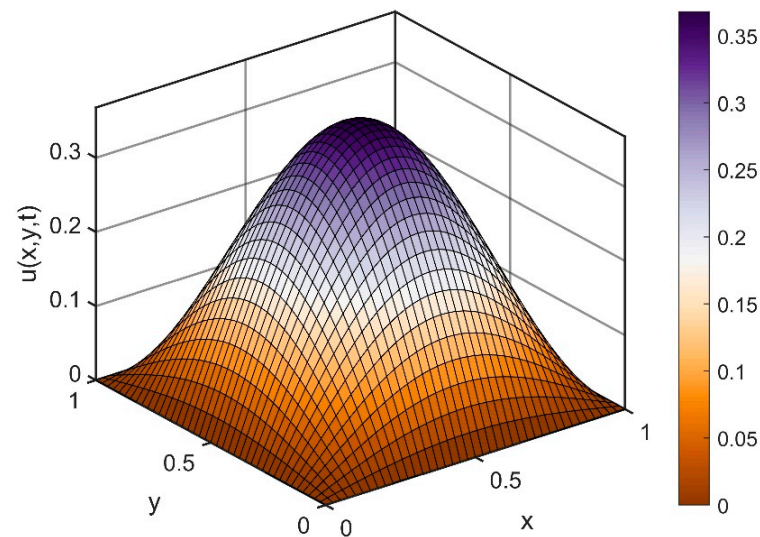


Figure 3. Temperature distribution obtained from the 2D heat conduction non-homogeneous equation subject to homogeneous DBCs at time $t = 1$.

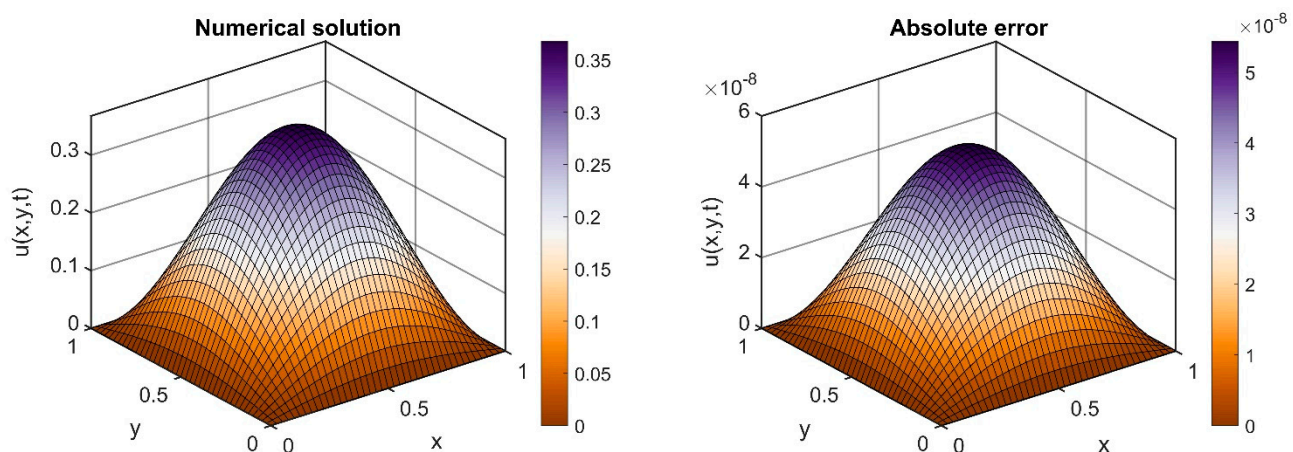


Figure 4. DST-CNCFD numerical solution for the 2D heat conduction non-homogeneous equation with homogeneous DBCs with 41×41 at $t = 1$.

4.3. Example 3: 2D Heat Conduction Equation with Variable Thermal Diffusivity

To evaluate the adaptability of the proposed DST-CNCFD solver, a geothermal model was designed. We consider the parabolic heat equation with variable thermal diffusivity ($\kappa = k/\rho c_p$), see Equation (6). Here, ρ represents the density of the medium, c_p is the thermal capacity under constant pressure, and k is the thermal conductivity. Then, the solution in the wavenumber domain for the DST-CNCFD scheme can be written as follows:

$$\tilde{u}(k_i, k_j, t_{p+1}) = \frac{2 + \kappa_{i,j} \Delta t (\lambda_{xi} + \lambda_{yj})}{2 - \kappa_{i,j} \Delta t (\lambda_{xi} + \lambda_{yj})} \tilde{u}(k_i, k_j, t_p) + \Delta t \frac{\tilde{f}(k_i, k_j, t_p) + \tilde{f}(k_i, k_j, t_{p+1})}{2 - \kappa_{i,j} \Delta t (\lambda_{xi} + \lambda_{yj})} \quad (37)$$

In this example, the geometric computational domain is designed as $1000 \text{ km} \times 1000 \text{ km}$, and the density $\rho = 3000 \text{ kg/m}^3$ and the heat capacity $c_p = 1000 \text{ J/kg/K}$ are constant for the entire model. The DBCs at the right, the left, the bottom, and the top boundaries of the model are all zero. The initial condition represents a background temperature of 1000 K , with a rectangular thermal wave at 1300 K positioned in the center (where ‘wave’ means perturbation in the temperature field), as illustrated in Figure 5. The thermal conductivity is defined as $k = 10 \text{ W/m/K}$ within the region of the thermal wave and $k = 3 \text{ W/m/K}$ for the surrounding medium.

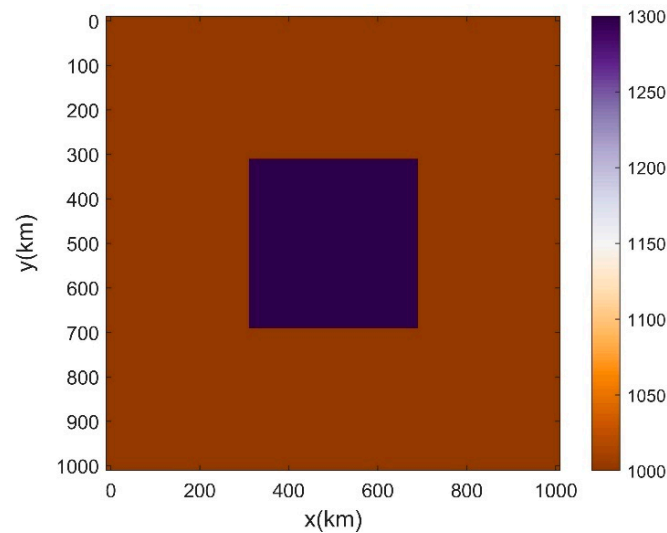


Figure 5. The initial temperature distribution of Example 3.

For solving this heat initial-boundary value problem with the DST-CNCFD solver, the time step is fixed at $\Delta t = 5$ Myr. We set discrete elements in a given domain as 51×51 (i.e., $N_x = N_y = 51$). The numerical temperature distributions of this model are shown in Figure 6. Modeling results show that the temperature decreases as time increases, which is consistent with the theoretical expectation.

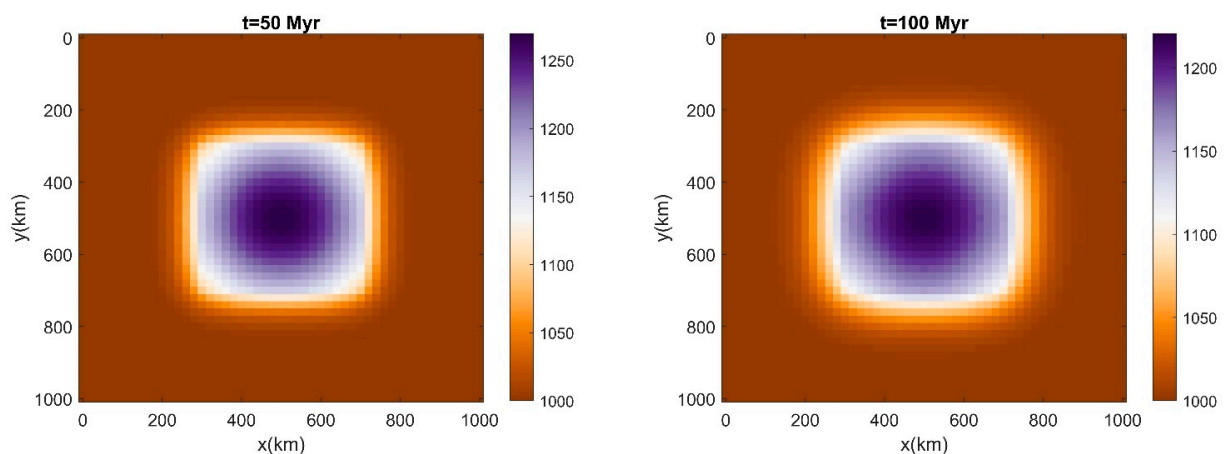


Figure 6. Cont.

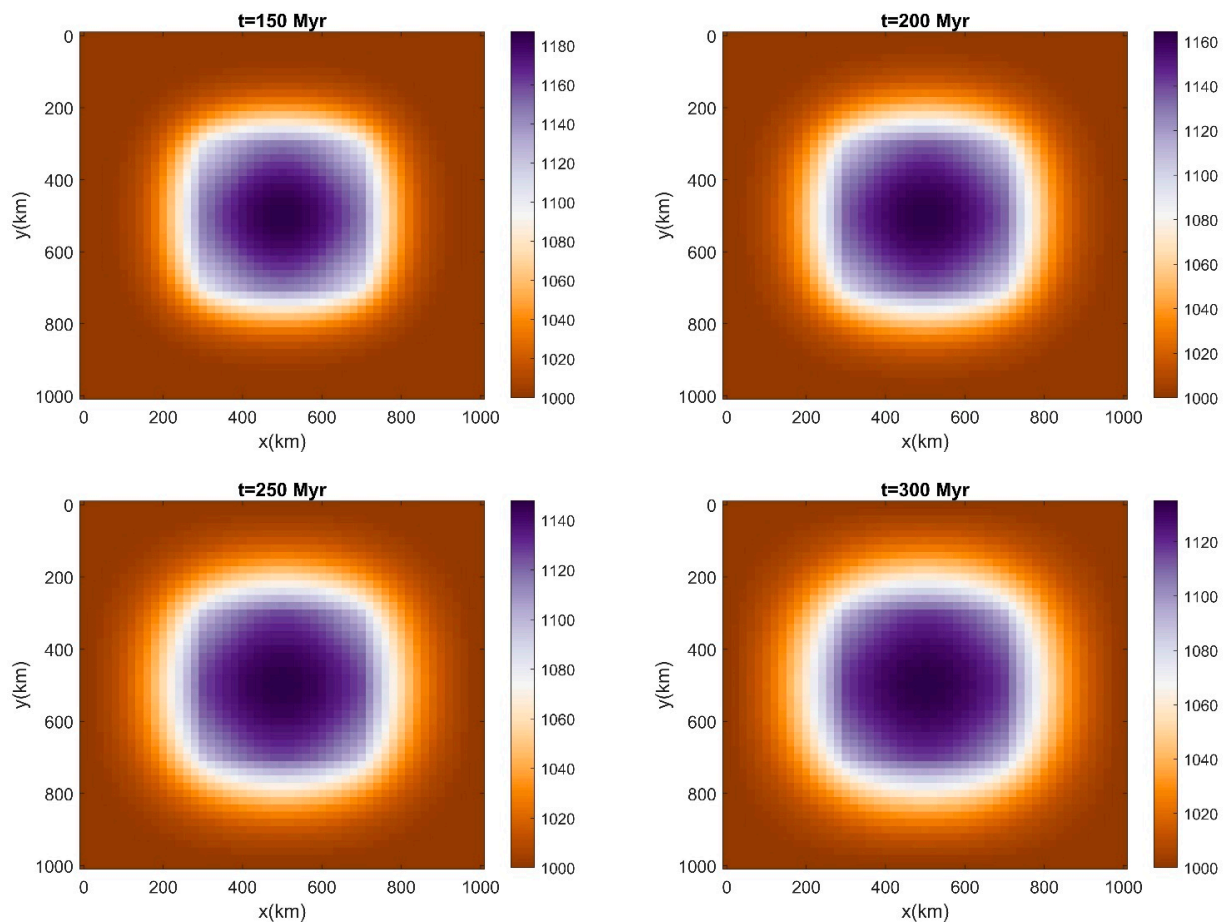


Figure 6. Numerical temperature distributions simulated by the DST-CNCFD method for Example 3.

5. Conclusions

The present work introduces and applies an innovative numerical technique, referred to as the DST-CNCFD method, aimed at overcoming the limitations of the traditional CNFD approximation when addressing high-dimensional heat conduction equations. The proposed approach utilizes DST technology to convert 2D heat conduction problems into algebraic problems independently, contrasting with the standard finite-difference methodology. Although this technique requires an inverse DST step for dimensionality reduction, it successfully circumvents the computational challenges associated with large-scale matrix inversion or iterative solvers. The accuracy and efficiency of the DST-CNCFD method are demonstrated through the resolution of 2D heat conduction equations with homogeneous DBCs. Numerical experiments confirm that this method substantially reduces computational costs while delivering highly accurate approximate results, which are in excellent agreement with the exact analytical solutions.

Although the DST-CNCFD algorithm presented in this paper primarily addresses the 2D heat conduction equation with homogeneous DBCs, our fast solver can be readily adapted to tackle heat conduction equations with non-homogeneous DBCs through appropriate modifications [40]. This versatility is particularly advantageous for compact finite-difference schemes used in the discretization of PDEs, such as the wave equation. As an illustration, we provide the MATLAB (R2024a) code for Example 1 in Appendix A. The numerical solutions generated using this code are achieved in a reduced timeframe.

Author Contributions: Conceptualization, C.L. and X.T.; formal analysis, C.L. and X.T.; funding acquisition, C.L.; methodology, C.L. and X.T.; project administration, X.T.; visualization, C.L. and

X.T.; supervision, X.T.; writing-original draft preparation, C.L.; writing-review and editing, C.L. and X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was partly supported by the National Major Scientific and Technological Project on Deep Earth (2025ZD1007800) and partly by the Guangzhou Construction Group Science and Technology Plan Project (2022-KJ005).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank Ying Liu from Ocean University of China, who modified this manuscript to improve the English writing quality and took part in helpful discussions about the results of the models. We would also like to thank the editors and the two anonymous reviewers for providing comments that substantially improved the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. DST-CNCFD Code

```
function Example1 ()
t1 = cputime;
xsize = 1;
ysize = 1;
tsize = 1;
xnum = 41;
ynum = 41;
tnum = 41;
dx = xsize/(xnum-1);
dy = ysize/(ynum-1);
dt = tsize/(tnum-1);
x = 0:dx:xsize;
y = 0:dy:ysize;
t = 0:dt:tsize;
Nx = xnum-1;
Ny = ynum-1;
M = tnum-1;
kx = 1:Nx-1;
ky = 1:Ny-1;
% Modified wavenumber
mwX = ((6/dx^2)*(2*cos(kx*pi/Nx)-2))./(cos(kx*pi/Nx) + 5);
mwY = ((6/dy^2)*(2*cos(ky*pi/Ny)-2))./(cos(ky*pi/Ny) + 5);
[MWX, MWY] = meshgrid(mwX,mwY);
Lambda = 1/8/pi/pi;
u = zeros(ynum,xnum,tnum);
% Initial conditions
for i = 1:ynum
    for j = 1:xnum
        u(i,j,1) = sin(2*pi*x(j))*sin(2*pi*y(i));
    end
end
u_in = u(2:end-1,2:end-1,1);
uhat(:,1) = dst(dst(u_in).').';
for k = 1:M
    uhat(:,k + 1) = uhat(:,k).*(2 + Lambda*dt*(MWX + MWY)). . .
```

```

        ./ (2-Lambda*dt*(MWX + MWY));
    end
    for k = 1:M + 1
        uu(:,k) = idst(idst(uhat(:,k)).').';
    end
    u(2:end-1,2:end-1,:) = uu;
    consuming_time = cputime-t1;
    figure(1)
    for k = 1:tnum
        surf(x,y,u(:,k));
        colorbar;
        title(['Numerical solution: t = ',num2str((k-1)*dt)]);
        xlabel('x');
        ylabel('y');
        zlabel('u');
        colormap('jet');
        drawnow;
        pause(0.1);
    end
    figure(2)
    [X,Y] = meshgrid(x,y);
    tt = 1;
    u_exact = sin(2*pi*X).*sin(2*pi*Y)*exp(-tt);
    surf(x,y,u_exact);
    xlabel('x');
    ylabel('y');
    zlabel('u');
    colormap('jet');
    colorbar;
    title('Analytical solution');
    figure(3)
    mesh(x,y,u(:,tnum)-u_exact);
    xlabel('x');
    ylabel('y');
    zlabel('u');
    colormap('jet');
    colorbar;
    title('Absolute error');

```

References

1. Su, L.D. A radial basis function (RBF)-finite difference (FD) method for the backward heat conduction problem. *Appl. Math. Comput.* **2019**, *354*, 232–247. [\[CrossRef\]](#)
2. Yuan, Y.; Li, C.; Song, H. An upwind-block-centered finite difference method for a semiconductor device for heat conduction and its numerical analysis. *J. Comput. Appl. Math.* **2023**, *425*, 114978. [\[CrossRef\]](#)
3. Sun, P.; Luo, Z.; Zhou, Y. Some reduced finite difference schemes based on a proper orthogonal decomposition technique for parabolic equations. *Appl. Numer. Math.* **2010**, *60*, 154–164. [\[CrossRef\]](#)
4. Beckers, P.; Beckers, B. A 66 line heat transfer finite element code to highlight the dual approach. *Comput. Math. Appl.* **2015**, *70*, 2401–2413. [\[CrossRef\]](#)
5. Agrawal, M.; Adlakha, N.; Pardasani, K.R. Three dimensional finite element model to study heat flow in dermal regions of elliptical and tapered shape human limbs. *Appl. Math. Comput.* **2010**, *217*, 4129–4140. [\[CrossRef\]](#)

6. Kaneko, H.; Bey, K.S.; Lenbury, Y.; Toghaw, P. Numerical experiments using hierarchical finite element method for nonlinear heat conduction in plates. *Appl. Math. Comput.* **2008**, *201*, 414–430. [\[CrossRef\]](#)
7. Čiegis, R.; Panasencko, G.; Pileckas, K.; Šumskas, V. ADI scheme for partially dimension reduced heat conduction models. *Comput. Math. Appl.* **2020**, *80*, 1275–1286. [\[CrossRef\]](#)
8. Mishra, S.C.; Roy, H.K. Solving transient conduction and radiation heat transfer problems using the lattice Boltzmann method and the finite volume method. *J. Comput. Phys.* **2007**, *223*, 89–107. [\[CrossRef\]](#)
9. Wang, P.; Yu, B.; Li, J.; Zhao, Y.; Shao, Q. A novel finite volume method for cylindrical heat conduction problems. *Int. Commun. Heat Mass Transf.* **2015**, *63*, 8–16. [\[CrossRef\]](#)
10. Jiang, G.; Liu, H.; Yang, K.; Gao, X. A fast reduced-order model for radial integral boundary element method based on proper orthogonal decomposition in nonlinear transient heat conduction problems. *Comput. Methods Appl. Mech. Eng.* **2020**, *368*, 113190. [\[CrossRef\]](#)
11. Yao, W.; Yu, B.; Gao, X.; Gao, Q. A precise integration boundary element method for solving transient heat conduction problems. *Int. J. Heat Mass Transf.* **2014**, *78*, 883–891. [\[CrossRef\]](#)
12. Yu, B.; Cao, G.; Meng, Z.; Gong, Y.; Dong, C. Three-dimensional transient heat conduction problems in FGMs via IG-DRBEM. *Comput. Method Appl. Mech. Eng.* **2021**, *384*, 113958. [\[CrossRef\]](#)
13. Sladek, J.; Sladek, V.; Zhang, C. Transient heat conduction analysis in functionally graded materials by the meshless local boundary integral equation method. *Comp. Mater. Sci.* **2003**, *28*, 494–504. [\[CrossRef\]](#)
14. Bartwal, N.; Shahane, S.; Roy, S.; Vanka, S.P. Simulation of heat conduction in complex domains of multi-material composites using a meshless method. *Appl. Math. Comput.* **2023**, *457*, 128208. [\[CrossRef\]](#)
15. Meng, Z.; Ma, Y.; Ma, L. A fast interpolating meshless method for 3D heat conduction equations. *Eng. Anal. Bound. Elem.* **2022**, *145*, 352–362. [\[CrossRef\]](#)
16. Li, X.; Zhao, S.; Qu, W. Localized Fourier collocation method for 2D transient heat conduction problems. *Appl. Math. Lett.* **2025**, *160*, 109331. [\[CrossRef\]](#)
17. Zogheib, B.; Tohidi, E. An accurate space-time pseudospectral method for solving nonlinear multi-dimensional heat transfer problems. *Mediterr. J. Math.* **2017**, *14*, 30. [\[CrossRef\]](#)
18. Onyejekwe, O. A comparison of time discretization schemes for the green element solution of the transient heat conduction equation. *Numer. Heat Transf. Part B Fundam.* **2000**, *38*, 405–422. [\[CrossRef\]](#)
19. Martín-Vaquero, J.; Sajavičius, S. The two-level finite difference schemes for the heat equation with nonlocal initial condition. *Appl. Math. Comput.* **2019**, *342*, 166–177. [\[CrossRef\]](#)
20. Goona, N.K.; Parne, S.R.; Sashidhar, S. Distributed source scheme to solve the classical form of Poisson equation using 3-D finite-difference method for improved accuracy and unrestricted source position. *Math. Comput. Simulat.* **2021**, *190*, 965–975. [\[CrossRef\]](#)
21. Harfash, A.J. High accuracy finite difference scheme for three-dimensional microscale heat equation. *J. Comput. Appl. Math.* **2008**, *220*, 335–346. [\[CrossRef\]](#)
22. Domínguez-Mota, F.J.; Armenta, S.M.; Tinoco-Guerrero, G.; Tinoco-Ruiz, J.G. Finite difference schemes satisfying an optimality condition for the unsteady heat equation. *Math. Comput. Simulat.* **2014**, *106*, 76–83. [\[CrossRef\]](#)
23. Zhang, J.; Zhao, J.J. Unconditionally stable finite difference scheme and iterative solution of 2D microscale heat transport equation. *J. Comput. Phys.* **2001**, *170*, 261–275. [\[CrossRef\]](#)
24. Chen, C.; Zhang, X.; Liu, Z.; Liu, Z.; Zhang, Y. A new high-order compact finite difference scheme based on precise integration method for the numerical simulation of parabolic equations. *Adv. Differ. Equ.* **2020**, *2020*, 15. [\[CrossRef\]](#)
25. Jodra, J.L.; Gurrutxaga, I.; Muguerza, J.; Yera, A. Solving Poisson's equation using FFT in a GPU cluster. *J. Parallel Distrib. Comput.* **2017**, *102*, 28–36. [\[CrossRef\]](#)
26. Schumann, U.; Sweet, R.A. Fast Fourier transforms for direct solution of Poisson's equation with staggered boundary conditions. *J. Comput. Phys.* **1988**, *75*, 123–137. [\[CrossRef\]](#)
27. Ren, Y.; Feng, H.; Zhao, S. A FFT accelerated high order finite difference method for elliptic boundary value problems over irregular domains. *J. Comput. Phys.* **2022**, *448*, 110762. [\[CrossRef\]](#)
28. Feng, H.; Zhao, S. FFT-based high order central difference schemes for three-dimensional Poisson's equation with various types of boundary conditions. *J. Comput. Phys.* **2020**, *410*, 109391. [\[CrossRef\]](#)
29. Costa, P. A FFT-accelerated multi-block finite-difference solver for massively parallel simulations of incompressible flows. *Comput. Phys. Commun.* **2022**, *271*, 108194. [\[CrossRef\]](#)
30. Gélébart, L. FFT-based simulations of heterogeneous conducting materials with combined non-uniform Neumann, periodic and Dirichlet boundary conditions. *Eur. J. Mech. A Solid.* **2024**, *105*, 105248. [\[CrossRef\]](#)
31. Wang, H.; Zhang, Y.; Ma, X.; Qiu, J.; Liang, Y. An efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions. *Computer. Math. Appl.* **2016**, *71*, 1843–1860. [\[CrossRef\]](#)

32. Gatiso, A.H.; Belachew, M.T.; Wolle, G.A. Sixth-order compact finite difference scheme with discrete sine transform for solving Poisson equations with Dirichlet boundary conditions. *Results Math.* **2021**, *10*, 100148. [[CrossRef](#)]
33. Morin, L.; Paxus, J. A fast numerical method for the conductivity of heterogeneous media with Dirichlet boundary conditions based on discrete sine-cosine transforms. *Compu. Methods Appl. Mech. Eng.* **2024**, *421*, 116772. [[CrossRef](#)]
34. Paux, J.; Morin, L.; Gélébart, L. A Discrete Sine-Cosine Transforms Galerkin Method for the Conductivity of Heterogeneous Materials with Mixed Dirichlet/Neumann Boundary Conditions. *Int. J. Numer. Meth. Eng.* **2024**, *26*, e7615. [[CrossRef](#)]
35. Paux, J.; Morin, L.; Gélébart, L.; Sanoko, A.M.A. A discrete sine-cosine based method for the elasticity of heterogeneous materials with arbitrary boundary conditions. *Comput. Methods Appl. Mech. Eng.* **2025**, *433*, 117488. [[CrossRef](#)]
36. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42. [[CrossRef](#)]
37. Wang, X.; Zhao, D.; Liu, J.; Zhang, Q. Efficient 2D modelling of magnetic anomalies using NUFFT in the Fourier domain. *Pure Appl. Geophys.* **2022**, *179*, 2311–2325. [[CrossRef](#)]
38. Balam, R.I.; Zapata, M.U. A new eighth-order implicit finite difference method to solve the three-dimensional Helmholtz equation. *Comput. Math. Appl.* **2020**, *80*, 1176–1200. [[CrossRef](#)]
39. Dehghan, M. Crank–Nicolson finite difference method for two-dimensional diffusion with an integral condition. *Appl. Math. Comput.* **2001**, *124*, 17–27. [[CrossRef](#)]
40. Pei, J.; Tong, X. A hybrid DST-accelerated finite-difference solver for 2D and 3D Poisson equation with Dirichlet boundary conditions. *Mathematics* **2025**, *13*, 2776. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.