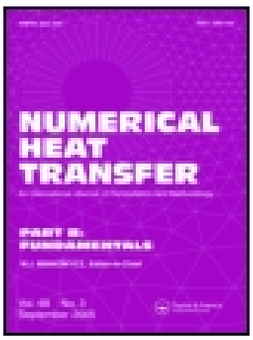

Taylor & Francis
Taylor & Francis Group

Numerical Heat Transfer, Part B: Fundamentals
An International Journal of Computation and Methodology

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: http://www.tandfonline.com/loi/unhb20

A time marching strategy for solving parabolic and elliptic equations with Neumann boundary conditions

Rex Kuan-Shuo Liu & Tony Wen-Hann Sheu

To cite this article: Rex Kuan-Shuo Liu & Tony Wen-Hann Sheu (2018): A time marching strategy for solving parabolic and elliptic equations with Neumann boundary conditions, Numerical Heat Transfer, Part B: Fundamentals, DOI: 10.1080/10407790.2018.1517551

To link to this article: https://doi.org/10.1080/10407790.2018.1517551

Published online: 26 Oct 2018.

NUMERICAL HEAT TRANSFER, PART B: FUNDAMENTALS
https://doi.org/10.1080/10407790.2018.1517551

Taylor & Francis
Taylor & Francis Group

A time marching strategy for solving parabolic and elliptic equations with Neumann boundary conditions

Rex Kuan-Shuo Liu[a] and Tony Wen-Hann Sheu[a,b,c]

[a]Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei, Taiwan; [b]Institute of Applied Mathematical Sciences, National Taiwan University, Taipei, Taiwan; [c]Center for Advanced Study in Theoretical Sciences, National Taiwan University, Taipei, Taiwan

ABSTRACT

In the community of computational fluid dynamics, pressure Poisson equation with Neumann boundary condition is usually encountered when solving the incompressible Navier–Stokes equations in a segregated approach such as SIMPLE, PISO, and projection methods. To deal with Neumann boundary conditions more naturally and to retain high order spatial accuracy as well, a sixth-order accurate combined compact difference scheme developed on staggered grids (NSCCD6) is adopted to solve the parabolic and elliptic equations subject to Neumann boundary conditions. The staggered grid system is usually used when solving the incompressible Navier–Stokes equations. By adopting the combined compact difference concept, there is no need to discretize Neumann boundary conditions with one-sided discretization scheme which is of lower accuracy order. The conventional Crank–Nicolson scheme is applied in this study for temporal discretization. For two-dimensional cases, D'yakonov alternating direction implicit scheme is adopted. A newly proposed time step changing strategy is adopted to improve convergence rate when solving the steady state solutions of the parabolic equation. High accuracy order of the currently proposed NSCCD6 scheme for one- and two-dimensional cases are shown in this article.

ARTICLE HISTORY
Received 10 June 2018
Accepted 25 August 2018

1. Introduction

In this research article, we are aiming at developing a sixth-order combined compact difference scheme (NSCCD6) on staggered grid to solve for the solutions of the parabolic and elliptic equation with Neumann boundary conditions. The elliptic Poisson equation can be regarded as the steady parabolic equation. The Poisson equation is usually shown in the computational electrostatics [1], potential flow [2], and ion-channel flow [3]. In the computation of incompressible flow with viscous fluid, the governing equations are the continuity equation and the momentum equation which form a velocity–pressure coupled system. Several segregated numerical algorithms are developed to solve the coupled system in a sequential manner such as SIMPLE [4], PISO [5], SOLA [6], and projection method [7] and [8]. In these algorithms, Poisson equation is solved for the pressure or pressure correction [9]–[11].

Numerical solution for the pressure Poisson equation (PPE) is of great importance in solving the incompressible Navier–Stokes equations to guarantee the divergence-free constraint on the velocity field [12]. The most difficult part is solving for the solution of the PPE with Neumann

CONTACT Tony Wen-Hann Sheu twhsheu@ntu.edu.tw Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/unhb.

Nomenclature

f	source term	μ_i	amplitude of ϕ_{ii}
h	grid spacing along each direction	η_i	amplitude of ϕ_i
i, j	nonstaggered grid indices	ϕ	variable of the governing equation
N	grid number along each direction	ϕ_x, ϕ_y	first derivative of ϕ along x - and y -direction
$NLevel$	number of time increments		
S_i	staggered grid index	ϕ_{xx}, ϕ_{yy}	second derivative of ϕ along x - and y -direction
L_{ii}	second-order spatial derivative operator along i -direction	ω_i	phase angle in i -direction
t	time (s)		
Δt	time increment		
Greek Symbols			
ζ	amplification factor		
ξ	amplitude of ϕ		

boundary condition. It is noted that Zhang [13] proposed a high order finite difference scheme with a compact correction term to solve the Poisson equation, but this scheme is applicable to Dirichlet boundary condition only. The resulting matrix from directly discretizing the governing equation and the Neumann boundary conditions with finite difference method is singular [14] and [15]. Furthermore, if high order combined compact difference (CCD) scheme [16] is adopted, a wider bandwidth and asymmetric matrix is constructed because the variable and its first- and second-order derivatives are treated as unknowns. Therefore, it is not easy to implement conventional linear system solvers such as the conjugate gradient (CG) method [17], Bi-CGSTAB method [18], or GMRES method [19]. One way to deal with this problem is to adopt truncated singular value decomposition (TSVD) method [20]–[22]. TSVD method computes the Moore–Penrose pseudo-inverse of the original matrix and is multiplied by the right-hand side vector to get the solutions. However, it is time consuming to perform the singular value decomposition when the matrix size is large. Therefore, it is not suitable to use TSVD method in multi-dimensional problems. An alternative is to solve the transient parabolic equation to steady state [23]–[26]. Since the resulting matrix by discretizing the parabolic equation with Neumann boundary conditions is not singular, general matrix solver can be adopted. More specifically, the resulting matrix is banded if the CCD scheme [16] is adopted, and an efficient direct matrix solver can be used to solve the linear system. On the other hand, alternating direction implicit (ADI) method is adopted to reduce matrix size to one-dimensional problem. However, it is time consuming to solve for the steady state solution of the parabolic equation. Several numerical treatments were proposed in [23], [25], and [26]. A time step changing strategy is proposed in this study to get rid of this problem.

This paper is organized as follows. Section 2 shows the governing equations and the staggered grid storage profile. The numerical schemes and the fundamental analysis including the proposed NSCCD6 scheme, ADI scheme for multidimensional problems, von Neumann linear stability analysis and the spectral analysis are detailed in Section 3. Several verification cases are conducted and the resulting results will be shown in Section 4. Finally, Section 5 draws some conclusions.

2. Governing equations

The elliptic Poisson equation in Eq. (1) and the parabolic heat equation in Eq. (2) will be investigated in the domain $(x, y) \in [0, 1]$.

$$0 = \nabla^2 \phi + f \tag{1}$$

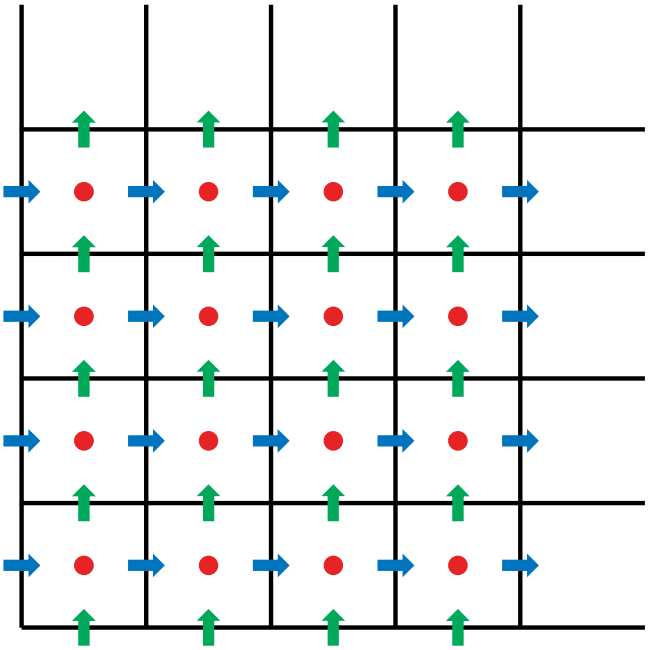

Figure 1. Staggered grid arrangement (red circle for ϕ, ϕ_{xx} *and* ϕ_{yy}; blue rightwards-arrow for ϕ_x; green upwards-arrow for ϕ_y).

$$\frac{\partial \phi}{\partial t} = \nabla^2 \phi + f \tag{2}$$

The solution ϕ is sought subjected to the following Neumann boundary conditions prescribed on the domain boundary $\partial\Omega$.

$$\begin{aligned} \phi_x(0,y,t) &= \phi_{x,0}(y,t) \quad \phi_x(1,y,t) = \phi_{x,1}(y,t) \\ \phi_y(x,0,t) &= \phi_{y,0}(x,t) \quad \phi_y(x,1,t) = \phi_{y,1}(x,t) \end{aligned} \tag{3}$$

It is remarked that the source term f and the four Neumann boundary conditions are time-dependent for the parabolic equation and are time-independent for the elliptic equation for which t is the fictitious time. The above variable ϕ is stored in the staggered grid system as shown in Figure 1 that is usually used in solving the incompressible Navier–Stokes equations. It is noted that $\phi, \phi_{xx},$ and ϕ_{yy} are stored at the cell center (i.e. grid point (i,j)) while ϕ_x and ϕ_y are stored at the cell faces (i.e. grid points $(i+\frac{1}{2},j)$ and $(i,j+\frac{1}{2})$, respectively).

The steady state solutions for Eq. (2) are seen as the solutions for Eq. (1) [23], [25], and [27]. Therefore, the proposed NSCCD6 scheme is capable of simulating both elliptic Poisson equation and parabolic heat equation with Neumann boundary conditions within the same formulation. When calculating the steady state solutions for Eq. (2), it usually takes a lot of computation time. To overcome this problem, a time step changing strategy is proposed to quickly eliminate errors relevant to different frequencies.

3. Numerical schemes

3.1. The sixth-order CCD scheme on staggered grids for spatial discretization

In this section, derivation of the adopted sixth-order CCD scheme (NSCCD6) will be detailed in staggered grids. For simplicity, the storage strategy for $\phi, \phi_x,$ and ϕ_{xx} is schematically shown in

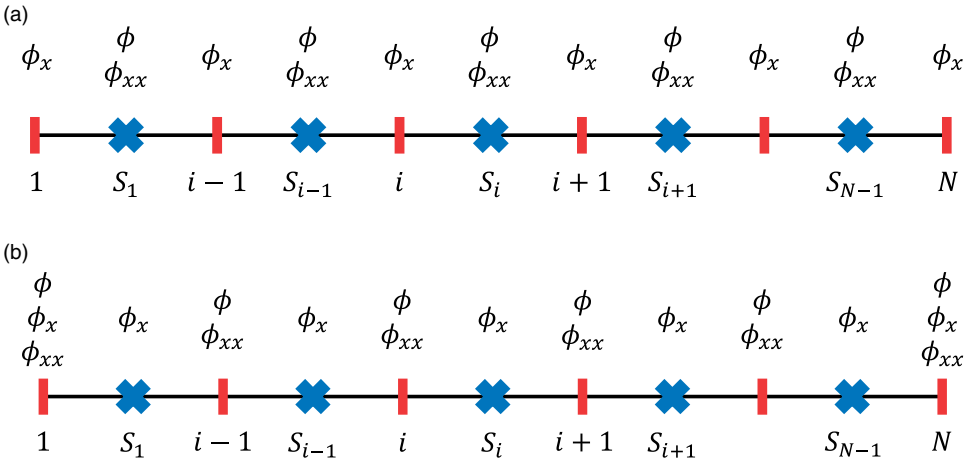

Figure 2. The staggered grids (a) used in this paper and (b) proposed in [28].

Figure 2a for the proposed scheme involving N grids and $N-1$ staggered grids and the grid spacing is h. It is worth noting that this staggered grid arrangement is different from that proposed by Chu [28] as shown in Figure 2b. In both grid systems, grid indices 1 and N are boundary points, S_1 and S_{N-1} are near-boundary staggered points, $i-1\sim N-1$ are interior nonstaggered points, and $S_2\sim S_{N-2}$ are interior staggered points. Our staggered grid arrangement is the same as that normally used in solving the incompressible Navier–Stokes equations. In other words, the function value and the second-order derivative are located at the cell center while the first derivative is located at the cell face. As a result, the NSCCD6 scheme is applicable to other Navier–Stokes solvers.

First of all, the equation for calculating the first order derivative on node i is assumed to take the form of

$$\left[a_{-1}(\phi_x)_{i-1}+(\phi_x)_i+a_1(\phi_x)_{i+1}\right]+h\left[b_{-1}(\phi_{xx})_{S_{i-1}}+b_1(\phi_{xx})_{S_i}\right]=\frac{1}{h}\left[c_{-1}(\phi)_{S_{i-1}}+c_1(\phi)_{S_i}\right] \tag{4}$$

In this formulation, the first derivative at grid point i is computed simultaneously with those at the grids $i-1$ and $i+1$ and the second derivatives at the staggered grids S_{i-1} and S_i. The R.H.S. contains the known function values at staggered grids S_{i-1} and S_i. By conducting the modified equation analysis, we can get the following equation which contains six unknowns.

$$\begin{aligned}(\phi_x)_i=&\ \phi_i\frac{1}{h}(c_{-1}+c_1)\\&+(\phi_x)_i\left(-a_{-1}-a_1-\frac{1}{2}c_{-1}+\frac{1}{2}c_1\right)\\&+(\phi_{xx})_i h\left(a_{-1}-a_1-b_{-1}-b_1+\frac{1}{8}c_{-1}+\frac{1}{8}c_1\right)\\&+(\phi_{xxx})_i h^2\left(-\frac{1}{2}a_{-1}-\frac{1}{2}a_1+\frac{1}{2}b_{-1}-\frac{1}{2}b_1-\frac{1}{48}c_{-1}+\frac{1}{48}c_1\right)\\&+(\phi_{xxxx})_i h^3\left(\frac{1}{6}a_{-1}-\frac{1}{6}a_1-\frac{1}{8}b_{-1}-\frac{1}{8}b_1+\frac{1}{384}c_{-1}+\frac{1}{384}c_1\right)\\&+(\phi_{xxxxx})_i h^4\left(-\frac{1}{24}a_{-1}-\frac{1}{24}a_1+\frac{1}{48}b_{-1}-\frac{1}{48}b_1-\frac{1}{3840}c_{-1}+\frac{1}{3840}c_1\right)\end{aligned} \tag{5}$$

These six unknowns can then be solved from the following linear system of algebraic equations.

$$\begin{aligned}
c_{-1}+c_1&=0\\
-a_{-1}-a_1-\frac{1}{2}c_{-1}+\frac{1}{2}c_1&=1\\
a_{-1}-a_1-b_{-1}-b_1+\frac{1}{8}c_{-1}+\frac{1}{8}c_1&=0\\
-\frac{1}{2}a_{-1}-\frac{1}{2}a_1+\frac{1}{2}b_{-1}-\frac{1}{2}b_1-\frac{1}{48}c_{-1}+\frac{1}{48}c_1&=0\\
\frac{1}{6}a_{-1}-\frac{1}{6}a_1-\frac{1}{8}b_{-1}-\frac{1}{8}b_1+\frac{1}{384}c_{-1}+\frac{1}{384}c_1&=0\\
-\frac{1}{24}a_{-1}-\frac{1}{24}a_1+\frac{1}{48}b_{-1}-\frac{1}{48}b_1-\frac{1}{3840}c_{-1}+\frac{1}{3840}c_1&=0
\end{aligned}$$

The six coefficients can then be uniquely derived as

$$a_{-1}=a_1=\frac{-7}{254}\qquad b_{-1}=-b_1=\frac{-17}{254}\qquad c_{-1}=-c_1=\frac{-120}{127} \tag{6}$$

The truncation error in Eq. (5) is obtained as $\frac{1}{7!}\frac{457}{2032}h^6\phi^{(7)}$ after substituting Eq. (6) into Eq. (5).

By applying the same procedures, the second-order derivatives at near-boundary staggered grids S_1 and S_{N-1} and interior staggered grids S_i can be derived as

$$\begin{aligned}
&\frac{1}{h}\left[\frac{1792}{4307}(\phi_x)_1-\frac{62432}{4307}(\phi_x)_2\right]+\left[(\phi_{xx})_{S_1}-\frac{3551}{4307}(\phi_{xx})_{S_2}\right]\\
&\quad=\frac{1}{h^2}\left[\frac{2449}{177}(\phi)_{S_1}-\frac{58533}{4307}(\phi)_{S_2}-\frac{1071}{4307}(\phi)_{S_3}+\frac{35}{12921}(\phi)_{S_4}\right]
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\frac{1}{h}\left[\frac{62432}{4307}(\phi_x)_{N-1}-\frac{1792}{4307}(\phi_x)_N\right]+\left[-\frac{3551}{4307}(\phi_{xx})_{S_{N-2}}+(\phi_{xx})_{S_{N-1}}\right]\\
&\quad=\frac{1}{h^2}\left[\frac{35}{12921}(\phi)_{S_{N-4}}-\frac{1071}{4307}(\phi)_{S_{N-3}}-\frac{58533}{4307}(\phi)_{S_{N-2}}+\frac{2449}{177}(\phi)_{S_{N-1}}\right]
\end{aligned} \tag{8}$$

$$\begin{aligned}
&\frac{144}{47}\frac{1}{h}\left[(\phi_x)_i-(\phi_x)_{i+1}\right]+(\phi_{xx})_{S_i}-\frac{5}{94}\left[(\phi_{xx})_{S_{i-1}}+(\phi_{xx})_{S_{i+1}}\right]\\
&\quad=-\frac{102}{47}\frac{1}{h^2}\left[(\phi)_{S_{i-1}}-2(\phi_{xx})_{S_i}+(\phi_{xx})_{S_{i+1}}\right]
\end{aligned} \tag{9}$$

The truncation errors of the above three equations are $\frac{29}{43925}h^5\phi^{(7)}$ and $\frac{1}{20160}h^6\phi^{(8)}$ for the near-boundary and the interior staggered grids, respectively. The purpose of deriving a fifth-order accuracy scheme is to preserve the rate of convergence as sixth-order for solving the parabolic equation with Neumann boundary conditions. For example, as documented in [29], a second-order (two-order less) scheme along with a fourth-order scheme for the near-boundary and interior points leads to third-order numerical solutions for solving the parabolic equation with Neumann boundary conditions. Finally, in order to compute the first and second-order derivatives at nonstaggered and staggered grids, the following matrix equation shall be solved under $a_1=\frac{-7}{254}, a_2=\frac{144}{47h}, b_1=\frac{17h}{254}, b_2=\frac{-5}{94}, c_1=\frac{1792}{4307h}, c_2=\frac{-62432}{4307h}, c_3=\frac{-3551}{4307}$.

$$
\left[\begin{array}{ccccccc|ccccccc}
1 & & & & & & & & & & & & & \\
a_1 & 1 & a_1 & & & & & -b_1 & b_1 & & & & & \\
 & \ddots & \ddots & \ddots & & & & & \ddots & \ddots & & & & \\
 & & \ddots & \ddots & \ddots & & & & & \ddots & \ddots & & & \\
 & & & a_1 & 1 & a_1 & & & & & -b_1 & b_1 & & \\
 & & & & & 1 & & & & & & & & \\
\hline
c_1 & c_2 & & & & & & 1 & c_3 & & & & & \\
 & a_2 & -a_2 & & & & & b_2 & 1 & b_2 & & & & \\
 & & \ddots & \ddots & & & & & \ddots & \ddots & \ddots & & & \\
 & & & a_2 & -a_2 & & & & & b_2 & 1 & b_2 & & \\
 & & & & -c_2 & -c_1 & & & & & c_3 & 1 & &
\end{array}\right]
\left\{\begin{array}{c}
(\phi_x)_1 \\ (\phi_x)_2 \\ \vdots \\ \vdots \\ (\phi_x)_{N-1} \\ (\phi_x)_N \\ \hline (\phi_{xx})_{S_1} \\ (\phi_{xx})_{S_2} \\ \vdots \\ (\phi_{xx})_{S_{N-2}} \\ (\phi_{xx})_{S_{N-1}}
\end{array}\right\}
$$

$$
= \frac{1}{h^2}
\left[\begin{array}{ccccc}
0 & & & & \\
\frac{-120h}{127} & \frac{120h}{127} & & & \\
 & \ddots & \ddots & \ddots & \\
 & & \ddots & \ddots & \ddots \\
 & & & \frac{-120h}{127} & \frac{120h}{127} \\
\hline
\frac{2449}{177} & \frac{-58533}{4307} & \frac{-1071}{4307} & \frac{35}{12921} & 0 \\
\frac{-102}{47} & \frac{204}{47} & \frac{-102}{47} & & \\
 & \ddots & \ddots & & \\
 & & \frac{-102}{47} & \frac{204}{47} & \frac{-102}{47} \\
 & \frac{35}{12921} & \frac{-1071}{4307} & \frac{-58533}{4307} & \frac{2449}{177}
\end{array}\right]
\left\{\begin{array}{c}
\phi_{S_1} \\ \phi_{S_2} \\ \vdots \\ \phi_{S_{N-2}} \\ \phi_{S_{N-1}}
\end{array}\right\}
+
\left\{\begin{array}{c}
(\phi_x)_1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ (\phi_x)_N \\ \hline 0 \\ 0 \\ \vdots \\ 0 \\ 0
\end{array}\right\}
\tag{10}
$$

The matrix shown in Eq. (10) is a penta-diagonal matrix and the computationally efficient direct matrix solver can be applied to obtain the solution from the above linear system.

The key feature of the adopted NSCCD6 scheme is that there is no need of using any low-order one-sided discretization scheme to discretize ϕ_x on the boundary points ($i = 1$ *and* N) when dealing with Neumann boundary condition for ϕ as seen in Eq. (10). Because the first- and second-order derivatives are computed simultaneously, Neumann boundary conditions for ϕ are just set as the known values for ϕ_x at the two boundary points.

3.2. Crank–Nicolson and D'yakonov ADI schemes for temporal discretization of two-dimensional parabolic equation

When solving multidimensional Eq. (2), D'yakonov ADI scheme [30] and [31] is adopted. Here the second-order accurate Crank–Nicolson scheme is applied first to Eq. (2), thereby leading to

$$
\frac{\phi^{n+1}-\phi^n}{\Delta t} = \frac{1}{2}(L_{xx}+L_{yy})\left(\phi^{n+1}+\phi^n\right) + f^{n+\frac{1}{2}} + O(\Delta t^2) \tag{11}
$$

In the above, the superscript n is the time level such that $t = n\Delta t$, and L_{xx} and L_{yy} are the second-order spatial differential operators with respect to x and y, respectively. Then, the terms at the time levels $n+1$ and n are collected on the L.H.S. and R.H.S., respectively.

$$
(1 - rL_{xx} - rL_{yy})\phi^{n+1} = (1 + rL_{xx} + rL_{yy})\phi^{n+1} + \Delta t f^{n+\frac{1}{2}} + O(\Delta t^3) \tag{12}
$$

where $r = \Delta t/2$. By adding the perturbation term $r^2 L_{xx}L_{yy}\phi^{n+1}$ to both hand sides and $r^2 L_{xx}L_{yy}\phi^n - r^2 L_{xx}L_{yy}\phi^n$ to the R.H.S., respectively, the final equation can be obtained after some algebraic manipulations.

$$(1 - rL_{xx})(1 - rL_{yy})\phi^{n+1} = (1 + rL_{xx})(1 + rL_{yy})\phi^n + \Delta t f^{n+\frac{1}{2}} + r^2 L_{xx}L_{yy}\left(\phi^{n+1} - \phi^n\right) + O(\Delta t^3) \tag{13}$$

The term $(\phi^{n+1} - \phi^n)$ on the R.H.S. of the above equation is of the order $O(\Delta t)$, so the third term $r^2 L_{xx}L_{yy}(\phi^{n+1} - \phi^n)$ is of the order $O(\Delta t^3)$ and can be absorbed into the fourth term as the error term. To solve Eq. (13), the D'yakonov ADI procedure is first applied to solve for an intermediate solution ϕ^* and, then, the solution ϕ^{n+1}.

$$(1 - rL_{xx})\phi^* = (1 + rL_{xx})(1 + rL_{yy})\phi^n + \Delta t f^{n+\frac{1}{2}} \tag{14}$$

$$(1 - rL_{yy})\phi^{n+1} = \phi^* \tag{15}$$

The differential operators on the L.H.S. of the above two equations take the same form. Therefore, we can solve ϕ^* and ϕ^{n+1} by combining Eqs. (14) or (15) with Eq. (10). The resulting matrix equation for solving Eqs. (10) and (14) or (15) is as follows.

$$\left[\begin{array}{cccccc|cccccc|cccccc}
1 & & & & & & & & & & & & a_1 & & & & & \\
 & 1 & & & & & & & & & & & & a_1 & & & & \\
 & & \ddots & & & & & & & & & & & & \ddots & & & \\
 & & & 1 & & & & & & & & & & & & a_1 & & \\
 & & & & 1 & & & & & & & & & & & & a_1 & \\
\hline
 & & & & & & 1 & & & & & & & & & & & \\
b_2 & -b_2 & & & & & b_1 & 1 & b_1 & & & & -b_3 & b_3 & & & & \\
 & \ddots & \ddots & & & & & \ddots & \ddots & \ddots & & & & \ddots & \ddots & & & \\
 & & \ddots & \ddots & & & & & \ddots & \ddots & \ddots & & & & \ddots & \ddots & & \\
 & & & b_2 & -b_2 & & & & & b_1 & 1 & b_1 & & & & -b_3 & b_3 & \\
 & & & & & & & & & & & 1 & & & & & & \\
\hline
c_2 & c_3 & c_4 & c_5 & & & c_6 & c_7 & & & & & 1 & c_1 & & & & \\
d_2 & -2d_2 & d_2 & & & & & d_3 & -d_3 & & & & d_1 & 1 & d_1 & & & \\
 & \ddots & \ddots & \ddots & & & & & \ddots & \ddots & & & & \ddots & \ddots & \ddots & & \\
 & & d_2 & -2d_2 & d_2 & & & & & d_3 & -d_3 & & & & d_1 & 1 & d_1 & \\
 & & c_5 & c_4 & c_3 & c_2 & & & & & -c_7 & -c_6 & & & & c_1 & 1 & \\
\end{array}\right]
\left\{\begin{array}{c}
\phi_{S_1} \\ \phi_{S_2} \\ \vdots \\ \phi_{S_{N-2}} \\ \phi_{S_{N-1}} \\ \hline \phi'_1 \\ \phi'_2 \\ \vdots \\ \phi'_{N-1} \\ \phi'_N \\ \hline \phi_{S_1}'' \\ \phi_{S_2}'' \\ \vdots \\ \phi_{S_{N-2}}'' \\ \phi_{S_{N-1}}''
\end{array}\right\}
=
\left\{\begin{array}{c}
SC_1 \\ SC_2 \\ \vdots \\ SC_{N-2} \\ SC_{N-1} \\ \hline \phi'_1 \\ 0 \\ \vdots \\ 0 \\ \phi'_N \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0
\end{array}\right\} \tag{16}$$

where $a_1 = -\frac{\Delta t}{2}, b_1 = \frac{-7}{254}, b_2 = \frac{120}{127}\frac{1}{h}, b_3 = \frac{17}{254}h, c_1 = \frac{-3551}{4307}, c_2 = \frac{-2449}{177}\frac{1}{h^2}, c_3 = \frac{58533}{4307}\frac{1}{h^2}, c_4 = \frac{1071}{4307}\frac{1}{h^2}, c_5 = \frac{-35}{12921}\frac{1}{h^2}, c_6 = \frac{1792}{4307}\frac{1}{h}, c_7 = \frac{-62432}{4307}\frac{1}{h}, d_1 = \frac{-5}{94}, d_2 = \frac{102}{47}\frac{1}{h}, d_3 = \frac{144}{47}\frac{1}{h}$. It can be seen that the final matrix is a banded matrix and the efficient direct solver can be applied to solve the linear system. The source terms $SC_1 \sim SC_{N-1}$ are calculated from the R.H.S. of Eqs. (14) or (15).

There are some additional efforts should be taken into account when using the proposed NSCCD6 scheme. First of all, the Neumann boundary condition for ϕ^* is needed to be calculated before solving Eq. (14). Differentiating Eq. (15) with respect to x, we can get the following equation to calculate the Neumann boundary condition for ϕ^*.

$$\phi_x^* = \phi_x^{n+1} - \frac{\Delta t}{2}\left(\phi_{yy}^{n+1}\right)_x \tag{17}$$

It is noted that $(\phi_{yy}^{n+1})_x = (\phi_x^{n+1})_{yy}$ where ϕ_x^{n+1} is prescribed as the Neumann boundary condition for ϕ^{n+1}. Therefore, the term $(\phi_x^{n+1})_{yy}$ on the left and right boundaries ($x = 0$ and 1) can be calculated with the high order CCD scheme given in [16] by differentiating $(\phi_x)^{n+1}$ twice with respect to y.

Second, there is a mixed derivative term $L_{xx}L_{yy}\phi^n$ shown on the R.H.S. when solving Eq. (14). Because $L_{yy}\phi^n = \phi_{yy}^n$ has been calculated at the previous time step together with $\phi^n, L_{xx}L_{yy}\phi^n$ can

be calculated by taking the second derivative of ϕ^n_{yy} with respect to x. As shown in Eq. (10), the NSCCD6 scheme calculates first and second derivatives simultaneously, implying that $(\phi^n_{yy})_x$ and $(\phi^n_{yy})_{xx}$ will be calculated at the same moment. As a result, the boundary value for $(\phi^n_{yy})_x$ should be calculated before calculating the interior values of $(\phi^n_{yy})_x$ and $(\phi^n_{yy})_{xx}$. The method for calculating $(\phi^n_{yy})_x$ on the left and right boundaries is the same used for calculating $(\phi^{n+1}_{yy})_x$ on the two boundaries mentioned above.

3.3. Von Neumann linear stability analysis of the proposed spatial discretization scheme

In this subsection, we investigate the stability of the proposed NSCCD6 scheme by von Neumann linear stability analysis. Assuming that the variable ϕ is periodic both in x- and y-direction and the source term f is zero. At the grid (j,k), the variable ϕ and its derivatives can be expressed as

$$\begin{aligned} \phi^n_{j,k} &= \xi^n e^{i(\omega_x j+\omega_y k)} \\ (\phi_x)^n_{j,k} &= \eta^n_x e^{i(\omega_x j+\omega_y k)} \qquad (\phi_{xx})^n_{j,k} = \mu^n_x e^{i(\omega_x j+\omega_y k)} \\ (\phi_y)^n_{j,k} &= \eta^n_y e^{i(\omega_x j+\omega_y k)} \qquad (\phi_{yy})^n_{j,k} = \mu^n_y e^{i(\omega_x j+\omega_y k)} \end{aligned}$$

where $i \equiv \sqrt{-1}, \xi^n, \eta^n_x, \mu^n_x, \eta^n_y, \mu^n_y$ are the amplitudes and ω_x, ω_y are the phase angles in x- and y-direction, respectively. The above relations can be substituted into Eqs. (4) and (9), thereby leading to

$$\begin{aligned} &-\frac{7}{254}\eta^n_x e^{i(\omega_x j+\omega_y k)}(e^{i\omega_x}+e^{-i\omega_x}) + \eta^n_x e^{i(\omega_x j+\omega_y k)} + \frac{17h}{254}\mu^n_x e^{i(\omega_x j+\omega_y k)}(e^{i\frac{1}{2}\omega_x}-e^{-i\frac{1}{2}\omega_x}) \\ &= \frac{120}{127h}\xi^n e^{i(\omega_x j+\omega_y k)}(e^{i\frac{1}{2}\omega_x}-e^{-i\frac{1}{2}\omega_x}) \end{aligned} \tag{18}$$

$$\begin{aligned} &-\frac{144}{47h}\eta^n_x e^{i(\omega_x j+\omega_y k)}(e^{i\frac{1}{2}\omega_x}-e^{-i\frac{1}{2}\omega_x}) + \mu^n_x e^{i(\omega_x j+\omega_y k)} - \frac{5}{94}\mu^n_x e^{i(\omega_x j+\omega_y k)}(e^{i\omega_x}+e^{-i\omega_x}) \\ &= -\frac{102}{47h^2}\xi^n e^{i(\omega_x j+\omega_y k)}(e^{i\omega_x}-2+e^{-i\omega_x}) \end{aligned} \tag{19}$$

According to Euler's formula, $\exp(i\omega) = \cos\omega + i\sin\omega$, the following two equations are obtained after doing some algebraic manipulations.

$$\left(1-\frac{7}{127}\cos\omega_x\right)\eta^n_x + i\frac{17h}{127}\sin\left(\frac{1}{2}\omega_x\right)\mu^n_x = i\frac{240}{127h}\sin\left(\frac{1}{2}\omega_x\right)\xi^n \tag{20}$$

$$i\frac{-288}{47h}\sin\left(\frac{1}{2}\omega_x\right)\eta^n_x + \left(1-\frac{5}{47}\cos\omega_x\right)\mu^n_x = \frac{204}{47h^2}(1-\cos\omega_x)\xi^n \tag{21}$$

By solving the above linear system, the amplitudes, η^n_x and μ^n_x, can be obtained as

$$\eta^n_x = i\frac{\xi^n B_x}{\Delta x A_x} = i\frac{\xi^n 252\sin\left(\frac{1}{2}\omega_x\right)(9\cos\omega_x+31)}{\Delta x(35\cos^2\omega_x+1484\cos\omega_x+3521)} \tag{22}$$

$$\mu^n_x = \frac{\xi^n C_x}{\Delta x^2 A_x} = \frac{\xi^n 12(119\cos^2\omega_x+602\cos\omega_x-721)}{\Delta x^2(35\cos^2\omega_x+1484\cos\omega_x+3521)} \tag{23}$$

Within the semi-discrete framework, the resulting equation without considering the source term becomes

$$\left(1-\frac{\Delta t}{2}L_{xx}\right)\left(1-\frac{\Delta t}{2}L_{yy}\right)\phi^{n+1}_{j,k} = \left(1+\frac{\Delta t}{2}L_{xx}\right)\left(1+\frac{\Delta t}{2}L_{yy}\right)\phi^n_{j,k} \tag{24}$$

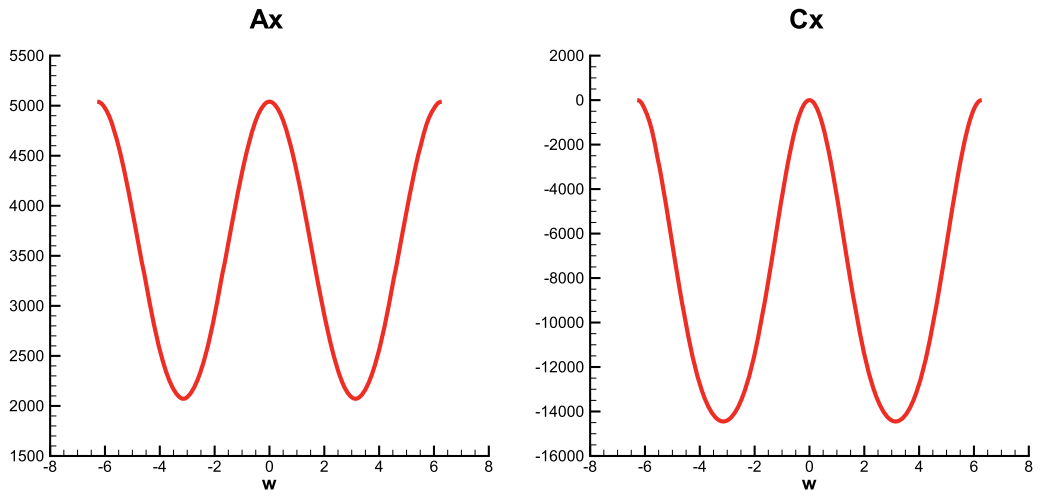

Figure 3. Profiles of A_x (left) and C_x (right).

The corresponding amplification factor ζ can be expressed as

$$\begin{aligned}\zeta &= \frac{u_{j,k}^{n+1}}{u_{j,k}^{n}} = \left(\frac{1+\frac{\Delta t}{2}L_{xx}}{1-\frac{\Delta t}{2}L_{xx}}\right)\left(\frac{1+\frac{\Delta t}{2}L_{yy}}{1-\frac{\Delta t}{2}L_{yy}}\right)\\ &= \left(\frac{1+\frac{\Delta t}{2}\frac{C_x}{\Delta x^2 A_x}}{1-\frac{\Delta t}{2}\frac{C_x}{\Delta x^2 A_x}}\right)\left(\frac{1+\frac{\Delta t}{2}\frac{C_y}{\Delta y^2 A_y}}{1-\frac{\Delta t}{2}\frac{C_y}{\Delta y^2 A_y}}\right)\\ &= g_x(\omega_x)g_y(\omega_y)\end{aligned} \tag{25}$$

The profiles of A_x and C_x (defined in Eq. (23)) are shown in Figure 3 and it can be observed that A_x is always positive while C_x is less than or equal to zero. Thus, the absolute values of functions g_x and g_y are always equal to or lower than one from the following deductions.

$$\begin{gathered}1+\frac{\Delta t}{2}\frac{C_x}{\Delta x^2 A_x} \le 1-\frac{\Delta t}{2}\frac{C_x}{\Delta x^2 A_x}\\ |g_x(\omega_x)| \le 1\\ |\zeta| \le |g_x(\omega_x)||g_y(\omega_y)| \le 1\end{gathered}$$

Therefore, through the von Neumann linear stability analysis, the newly proposed NSCCD6 scheme is unconditionally stable under the periodic boundary conditions.

3.4. Spectral analysis of the proposed spatial discretization scheme

Because the von Neumann linear stability analysis considers periodic boundary conditions, the interior schemes are considered excluding the near-boundary schemes. In order to check the stability condition when taking both the interior and near-boundary schemes into account, the spectral analysis is adopted in this study as well. Here, we first assume that there are two different solutions, $\vec{\phi}_1^{\,n}$ and $\vec{\phi}_2^{\,n}$, corresponding to two different initial conditions under the same Neumann boundary conditions. The difference between these two solutions is denoted as $\vec{\theta}^{\,n} = \vec{\phi}_1^{\,n} - \vec{\phi}_2^{\,n}$. The governing equation for $\vec{\theta}^{\,n}$ can be derived from Eq. (2) as

Table 1. Minimum and maximum eigenvalues of matrix $[D]$ for different grid numbers.

Grid no.	Min.	Max.
11	0.5232	1.0000
101	0.9327	1.0000
1001	0.9931	1.0000
10001	0.9993	1.0000

$$\frac{\{\theta\}^{n+1}-\{\theta\}^{n}}{\Delta t}=\frac{1}{2}[C_1]\left(\{\theta\}^{n+1}+\{\theta\}^{n}\right) \tag{26}$$

With a further deduction, we can get the solution of $\vec{\theta}^{n+1}$ in terms of its initial condition, $\vec{\theta}^{0}$ given below

$$\begin{aligned}\{\theta\}^{n+1} &= \left([I]-r[C_1]\right)^{-1}\left([I]+r[C_1]\right)\{\theta\}^{n}\\ &= [D]\{\theta\}^{n}\\ &= [D]^{n+1}\{\theta\}^{0}\end{aligned} \tag{27}$$

The derivation of the matrices $[C_1]$ and $[D]$ are shown in the Appendix. Thanks to the spectral analysis, the numerical scheme is stable provided that the spectral radius or the maximum absolute eigenvalue of the matrix $[D]$ is less than or equal to 1. Note that the computation of the matrix $[D]$ is analytically infeasible, the maximum absolute eigenvalue of it cannot be computed exactly. Therefore, eigenvalues can be obtained numerically for the four cases involving different grid numbers. Table 1 shows the minimum and maximum eigenvalues of the matrix $[D]$ obtained under different grid numbers. It can be observed that all the maximum eigenvalues are nominally equal to 1 and the numerical scheme is said to be stable.

3.5. The newly proposed time step changing strategy for solving the steady state solutions

Since Peacemann and Rachford [23] who first proposed the pioneer work of the ADI scheme to solve for the solutions of the parabolic and elliptic equations, few works have been done focusing on calculating the steady state solutions of the parabolic equation. Some previous works are shown in [24]–[27], and [32]. As stated in the previous work [23], a considerable computational effort needs to be made to reach the steady state solutions when using a single time increment. The errors corresponding to different frequencies cannot be eliminated efficiently. Peacemann and Rachford [23] showed that errors corresponding to different wave numbers can be rapidly eliminated when using different time step sizes which are derived from the amplification factors by forcing them equal to zero. However, if every time step size was utilized only once, the error could be reduced by a factor of 10^{-3} only. Still, it was far away from the steady state solution. Therefore, we propose to solve the transient parabolic equation to get the steady state solutions with a V-cycle time step changing strategy. The time increment is iteratively changed in the order of $\Delta t_{min}=\Delta t_1\rightarrow\Delta t_2\rightarrow\cdots\rightarrow\Delta t_{NLevel-1}\rightarrow\Delta t_{NLevel}=\Delta t_{max}\rightarrow\cdots\rightarrow\Delta t_1$. The series of the time increments, $\Delta t_1\sim\Delta t_{NLevel}$, can be calculated from the amplification factors. According to [23], the number of different time increments is *NLevel* which is the same as the cell number, say $N-1$. However, for large number of grid points, most of the time increments are clustered closely as shown in Figure 4. As can be seen, the time increment grows slowly between indices 1 and 300 but rapidly changes from 396 to 400 if the number of grid points is $N=401$. Peacemann and Rachford also showed that for some clustered values of Δt, an average time increment could be used. For example, $\Delta t=0.003$ could be used instead of using $\Delta t=0.00229, 0.00287, 0.003809$, and 0.005451. The most important idea is that the range of the numerical time increments (time increments used in numerical simulation) should cover the theoretical time increments (time increments derived from the amplification factors).

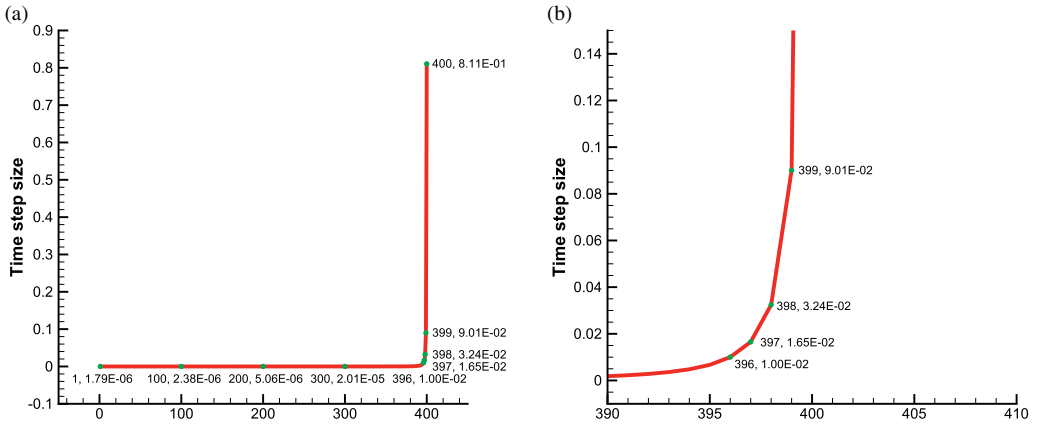

Figure 4. Theoretical time increment distribution for the case with the grid number of $N = 401$. (a) Global view and (b) Zoomed view.

According to Peacemann and Rachford, they only showed that the theoretical time increments can be determined from the corresponding amplification factors. However, the procedure to do sampling, for example, how to determine the averaged time increment $\Delta t = 0.003$ shown above, was not shown in the paper [23]. On the other hand, Douglas [33] showed that if the exact solution ϕ is in the form of continued product, for example $\phi = \exp(-\pi^2 t) \times \cos(\pi x) \times \cos(\pi y)$ in multidimensional problems, the perturbation terms (i.e. the third term on the R.H.S. of Eq. (13)) introduced in deriving ADI scheme could cause a larger numerical error. Because the perturbation term can be approximated as the cross-derivative term shown below and is not zero, so that the splitting error may make numerical solutions unable to converge when solving for steady state solutions if larger time steps are utilized.

$$\frac{1}{4}\Delta t^3 \frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}\frac{\partial}{\partial t}\phi \tag{28}$$

In our numerical tests, if the Peacemann and Rachford type [23] of time increments (Δt_{PR}) is adopted, we cannot get the steady state solutions. The criteria for obtaining a steady state solution is that the root-mean-square of the difference between two consecutive time step solutions is less than a tolerance ϵ. Therefore, we need to provide a time increment series such that the smallest time increment is small enough to avoid large splitting error in the first few iterations and the largest time increment is large enough to get the steady state solution quickly. Also, a sampling procedure is adopted to decrease the number of different time increments.

In our proposed time step series, the smallest time increment is calculated by the squares of the stability constraint of the forward-time central-space scheme while the largest time step is set to be equal to one.

$$\Delta t_{min,1D} = \left(\frac{h^2}{2}\right)^2, \quad \Delta t_{min,2D} = \left(\frac{h^2}{4}\right)^2, \quad \Delta t_{max} = 1.0 \tag{29}$$

Second, the value of *NLevel* is determined by a multigrid-like method. The value of h used to calculate Δt_{min} is considered as the grid size corresponding to the finest mesh, say h_1. The coarsest mesh size, as a result, is $h_{NLevel} = 2^{NLevel-1}h_1$ if the mesh size h_i is assumed to be twice of the mesh size h_{i-1}. By keeping h_{NLevel} smaller than one, we can obtain the following relation to determine the value of *NLevel*, such that

$$NLevel \leq 1 + \frac{\log(N-1)}{\log 2} \tag{30}$$

Table 2. The minimum and maximum time increments of the PR-type and the newly proposed method for 1D and 2D cases.

	N	11	51	101	201
AF	Δt_{min}	2.9032E−03	1.1478E−04	2.8686E−05	7.1708E−06
	Δt_{max}	8.1057E−01	8.1057E−01	8.1057E−01	8.1057E−01
1D	Δt_{min}	2.5000E−05	4.0000E−08	2.5000E−09	1.5625E−10
	Δt_{max}	1.0000E−00	1.0000E−00	1.0000E−00	1.0000E−00
2D	Δt_{min}	6.2500E−06	1.0000E−08	6.2500E−10	3.9063E−11
	Δt_{max}	1.0000E−00	1.0000E−00	1.0000E−00	1.0000E−00

Thus, the *NLevel* used in our simulation is the largest integer that is smaller or equal to the R.H.S. of the above equation. Finally, after getting $\Delta t_{min}, \Delta t_{max},$ and NLevel, we should decide the distribution profile of the time increments between these two extreme values with a control parameter. A simple way to achieve this goal is to make the time step series as a geometric sequence. Therefore, the time increment series become

$$\Delta t_{min} = \Delta t_1 < \Delta t_2 < \cdots < \Delta t_{NLevel-1} < \Delta t_{NLevel} = \Delta t_{max} \tag{31}$$

where $\Delta t_i / \Delta t_{i-1}$ is a constant. Table 2 shows the minimum and the maximum time increments of PR-type (Δt_{PR}) as proposed in [23] and the newly proposed method for the one- and two-dimensional cases. It is worth noting that even we have assumed there are several levels of grid size, they are only used to determine the value of *NLevel*. In other words, under the cases with different values of Δt, the mesh size used in our simulation is h_1 in time marching to steady state.

4. Verification study

In this section, several verification problems are conducted. Numerical results for heat equation and Poisson equation with Neumann boundary conditions are shown for one- and two-dimensional cases. The spatial rate of convergence (sroc) for each case is also calculated. In this section, the spatial rate of convergence is calculated by the following equation.

$$\text{sroc}_i = \frac{\log\left(L_2 - \text{error}_{N_{i-1}} / L_2 - \text{error}_{N_i}\right)}{\log\left(h_{i-1}/h_i\right)} \tag{32}$$

In the above, h is the grid size and it is assumed to be $h = \Delta x = \Delta y$ and the error L_2-norms are calculated by the following two equations for one- and two-dimensional cases, respectively.

$$(L_2 - \text{norm})_{1D} = \sqrt{\frac{1}{N-1}\sum_{i=S_1}^{S_{N-1}} \left(\phi_{numerical} - \phi_{exact}\right)^2} \tag{33}$$

$$(L_2 - \text{norm})_{2D} = \sqrt{\frac{1}{(N-1)^2}\sum_{i=S_1}^{S_{N-1}}\sum_{j=S_1}^{S_{N-1}} \left(\phi_{numerical} - \phi_{exact}\right)^2} \tag{34}$$

4.1. One-dimensional heat equation

In this test case, the computational domain is $x \in [0, 1]$. The exact solution is $\phi_{exact} = \exp\left(-\frac{1}{2}\pi^2 t\right)\cos(\pi x) + x^2 + xt$. In other words, the nonhomogeneous Neumann boundary conditions are $\phi_x(x=0) = \tan d\phi_x(x=1) = t+2$ and the source term is $f = \frac{1}{2}\pi^2 \exp\left(-\frac{1}{2}\pi^2 t\right)\cos(\pi x) + x - 2$. To get the spatial rate of convergence, time increment is set to 10^{-5} to decrease temporal discretization error. Three different grid numbers are chosen as $N = 11, 21, 41$. The simulation stops at $T = 1.0$s. The numerical results computed by the scheme

Table 3. Sroc for solving the one-dimensional heat equation.

N	NSCCD6		Reference	
	Error L_2-norm	sroc	Error L_2-norm	sroc
11	3.9189E–007	–	2.3385E–004	–
21	3.2868E–009	6.90	3.2681E–005	2.84
41	3.9269E–011	6.39	4.2533E–006	2.94

Table 4. Sroc for solving the two-dimensional heat equation.

N	NSCCD6		Reference	
	Error L_2-norm	sroc	Error L_2-norm	sroc
11^2	1.0525E–006	–	1.5042E–006	–
21^2	8.3696E–009	6.97	2.2505E–007	2.74
41^2	6.6341E–011	6.98	3.0160E–008	2.90

reported in [34] are compared with our results as shown in Table 3. Higher accuracy order can be seen for the cases using the proposed NSCCD6 scheme.

4.2. Two-dimensional heat equation

The computational domain is $(x, y) \in [0, 1]$ for the two-dimensional case. The exact solution is $\phi = \exp(-\pi^2 t)\cos(\pi x)\cos(\pi y) + (x^2 + y^2) + (x + y)t$ and the Neumann boundary conditions are $\phi_x(0, y, t) = \phi_y(x, 0, t) = \tan d\phi_x(1, y, t) = \phi_y(x, 1, t) = t + 2$. The source term in this case is $f = \pi^2 \exp(-\pi^2 t)\cos(\pi x)\cos(\pi y) + x + y - 4$. The time increment is 10^{-5} for decreasing the temporal discretization error. Grid numbers, $N^2 = 11^2, 21^2$ and 41^2, are chosen to compute the sroc. From the numerical results shown in Table 4, we can get much better spatial rates of convergences (sroc) than those calculated from the reference numerical methods detailed in [35].

4.3. One-dimensional Poisson equation

The first steady state problem under investigation is the one-dimensional Poisson equation with Neumann boundary conditions. The solution domain is $x \in [0, 1]$ and the exact solution is $\phi(x) = -\cos(2\pi x)$. The corresponding Neumann boundary condition is $\phi_x = 2\pi \sin(2\pi x)$ which is zero at $x = 0$ and 1. The source term can be calculated from Eq. (1). For the steady state problem, both sroc at small number of grid points and speed-up at large number of grid points are calculated. It is worth noting that zero initial condition is usually used when utilizing iterative methods, but the numerical solution calculated by CG method with the second-order central difference scheme is used as the initial condition for the proposed NSCCD6 scheme. The convergence criterion for the time stepping NSCCD6 scheme is that the root-mean-squares of the difference between two consecutive time step solutions is less than 10^{-14}. First, for the cases of calculating sroc, the minimum time increments calculated from Eq. (29) are used to decrease temporal errors with the grid numbers $N = 11, 21$, and 41. The numerical results are shown in Table 5. Numerical solutions obtained by CG and NSCCD6 methods are shown. Here the superconvergent solutions can be obtained from the proposed NSCCD6 scheme. An interesting phenomenon can be observed that the larger the *NLevel*, the smaller the CPU time and the iteration number. Even though *NLevel* can be enlarged to any larger number, but the calculation strategy for *NLevel* shown in Eq. (30) was adopted here. Here, the computation times for CG and NSCCD6 are monitored separately. Therefore, the total computation time is the sum of the computation times of CG and NSCCD6. It is interesting that the computation times for NSCCD6 are almost the same as those for the CG method, thus the NSCCD6 scheme with time increment changing strategy is an efficient method for solving the Poisson equation with Neumann

Table 5. Numerical results for calculating the sroc of the 1D Poisson equation.

N	NSCCD6					CG	
	Error L_2-norm	sroc	*NLevel*	CPU time (s)	Iterative no.	Error L_2-norm	CPU time (s)
11	1.5607E–003	–	4	2.2482E–002	331	2.3729E–002	1.7074E–002
21	1.7293E–005	6.50	5	1.8359E–002	193	5.8445E–003	1.7471E–002
41	1.4911E–007	6.86	6	1.7879E–002	91	1.6981E–003	1.7719E–002

Table 6. Numerical results for calculating the speed-up of the 1D Poisson equation.

N	*NLevel*	CPU time for CG (s)	CPU time for NSCCD6 (s)	Iterative no.	total CPU time (s)	Speed-up
101	1	1.6369E–002	1.1891E–000	6237	1.2055E–000	–
	7	1.6369E–002	1.7267E–002	37	3.3636E–002	35.84
201	1	1.7499E–002	2.9637E + 001	21235	2.9654E + 001	–
	8	1.7499E–002	7.3015E–002	42	9.0514E–002	327.61
401	1	1.7628E–002	4.2054E + 002	69923	4.2055E + 002	–
	9	1.7628E–002	1.9804E–001	33	2.1567E–001	1949.97

boundary conditions. By comparing L_2-norms of error, NSCCD6 scheme can significantly improve the solutions obtained from the CG method. In summary, NSCCD6 scheme can reduce errors with a factor of $10^{-1}\sim10^{-4}$ using only twice of the computation time of the CG method.

Secondly, for the cases with larger number of grid points, the efficiency of applying the proposed method is clearly shown. Here the speed-up is defined as following.

$$\text{speed}-\text{up} = \frac{\text{total computation time for } NLevel = 1}{\text{total computation time for } Nlevel \text{ calculated from Eq. } (30)} \tag{35}$$

Grid numbers $N = 101, 201$, and 401 are considered. From Table 6, we can observe that the computation time and the iteration number for the cases using the time increment changing strategy can be significantly reduced. The speed-up is more than two thousand for the case of $N = 401$. It is believed that the speed-up can be even higher for the cases involving a larger value of N.

4.4. Two-dimensional Poisson equation

The last test case is the two-dimensional Poisson equation with Neumann boundary conditions. The solution domain is $(x, y) \in [0, 1]$ with the exact solution $\phi = -\cos(2\pi x)\cos(2\pi y)$. The Neumann boundary conditions are $\phi_x = 2\pi \sin(2\pi x)\cos(2\pi y)$ and $\phi_y = 2\pi \cos(2\pi x)\sin(2\pi y)$ which are zero at the four boundaries. The source term can be calculated from Eq. (1). Grid numbers $N^2 = 21^2, 41^2$, and 81^2 are used for calculating both sroc and speed-up. The sroc shows super-convergent results as detailed in Table 7. The solution errors can be reduced by a factor of 10^{-3} *to* 10^{-6} from the CG solutions to the NSCCD6 solutions. It can be seen that the speed-up for solving two-dimensional steady state Poisson equation with the proposed NSCCD6 scheme and time increment changing strategy can be tremendously increased to the orders of 10^3 *to* 10^6 as shown in Table 8. The key to success is that for two-dimensional cases, errors of different frequencies can be quickly damped under different time increments.

5. Concluding remarks

In this paper, we are aiming at solving parabolic and elliptic equations with Neumann boundary conditions with sixth-order accurate CCD schemes on staggered grids. First, the NSCCD6 scheme is derived on the staggered grid system that is usually used in computational fluid dynamics society. In doing so, Neumann boundary condition is not needed to be discretized with one-sided scheme which is of lower accuracy order. On the other hand, this Poisson solver can be adopted by any other Navier–Stokes solver. Second, when solving the transient parabolic equation to the

Table 7. Numerical results for calculating the sroc of the 2D Poisson equation.

N	NSCCD6					CG	
	Error L_2-norm	sroc	*NLevel*	CPU time (s)	Iterative no.	Error L_2-norm	CPU time (s)
21	7.5819E–006	–	5	3.6107E–002	33	4.1327E–003	2.8218E–002
41	6.3328E–008	6.90	6	1.3990E–001	30	1.0293E–003	3.4552E–002
81	4.5043E–010	7.14	7	5.2296E–001	24	2.5710E–004	5.0559E–002

Table 8. Numerical results for calculating the speed-up of the 2D Poisson equation.

N	*NLevel*	CPU time for CG (s)	CPU time for NSCCD6 (s)	Iterative no.	total CPU time (s)	Speed-up
21	1	2.8218E–002	2.8809E + 002	585777	2.8812E + 002	–
	5	2.8218E–002	3.6107E–002	33	6.4325E–002	4.4791E + 003
41	1	3.4552E–002	2.2164E + 004	6327918	2.2164E + 004	–
	6	3.4552E–002	1.3990E–001	30	1.7445E–001	1.2705E + 005
81	1	5.0559E–002	1.1125E + 006	66719498	1.1125E + 006	–
	7	5.0559E–002	5.2296E–001	24	5.7352E–001	1.9398E + 006

steady state, errors of different frequencies are hardly and slowly to be eliminated if only a single time increment is utilized. Therefore, a time increment changing strategy is proposed to accelerate convergence. The underlying idea is to provide a series of time increments which cover the range of the amplification factors and reducing splitting errors introduced in the course of ADI procedures when solving the two-dimensional equation. Through numerical verifications, superconvergent property is seen and a huge computation time is saved by using the proposed NSCCD6 scheme and time increment changing strategy.

Funding

This work was supported by the Ministry of Science and Technology (MOST) of Republic of China (R.O.C.) under the Grants MOST105-2221-E-002-066 and MOST106-2221-E-002-107-MY2.

References

[1] D. Xie and Y. Jiang, "A nonlocal modified Poisson-Boltzmann equation and finite element solver for computing electrostatics of biomolecules," J. Comput. Phys., vol. 322, pp. 1–20, 2016.

[2] O. M. Faltinsen, R. Firoozkoohi, and A. N. Timokha, "Analytical modeling of liquid sloshing in a two-dimensional rectangular tank with a slat screen," J. Eng. Math., vol. 70, nos. 1–3, pp. 93–109, 2011.

[3] T. W. H. Sheu, Y. G. Bhumkar, S. T. Yuan, and S. C. Syue, "Development of a high-resolution scheme for solving the PNP-NS equations in curved channels," Commun. Comput. Phys., vol. 19, no. 2, pp. 496–533, 2016.

[4] S. Patankar, Numerical Heat Transfer and Fluid Flow. New York, NY; Boca Raton, FL, USA: Hemisphere Publishing Corporation; CRC Press (Taylor & Francis Group), 1980.

[5] R. I. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," J. Comput. Phys., vol. 62, no. 1, pp. 40–65, 1986.

[6] C. W. Hirt, B. D. Nichols, and N. C. Romero, SOLA-A Numerical Solution Algorithm for Transient Fluid Flows. NM, USA: Los Alamos Scientific Lab, 1975.

[7] A. J. Chorin, "The numerical solution of the Navier-Stokes equations for an incompressible fluid," Bull. Am. Math. Soc., vol. 73, no. 6, pp. 928–931, 1967.

[8] A. J. Chorin, "Numerical solution of the Navier-Stokes equations," Math. Comput., vol. 22, no. 104, pp. 745–762, 1968.

[9] A. Ashrafizadeh, B. Alinia, and P. Mayeli, "A new co-located pressure-based discretization method for the numerical solution of incompressible Navier-Stokes equations," Numer. Heat Transfer, Part B, vol. 67, no. 6, pp. 563–589, 2015.

[10] Y.-H. Hwang, K.-C. Ng, and T. Wen-Hann Sheu, "An improved particle smoothing procedure for Laplacian operator in a randomly scattered cloud," Numer. Heat Transfer, Part B, vol. 70, no. 2, pp. 111–135, 2016.

[11] Q. Luo, T. Ren, and D. Liang, "Discretized pressure Poisson algorithms for the steady incompressible flow on a nonstaggered grid," Numer. Heat Transfer, Part B, vol. 71, no. 6, pp. 549–559, 2017.
[12] K.-S. Liu, T. W.-H. Sheu, Y.-H. Hwang, and K.-C. Ng, "High-order particle method for solving incompressible Navier-Stokes equations within a mixed Lagrangian-Eulerian framework," Comput. Methods Appl. Mech. Eng., vol. 325, pp. 77–101, 2017.
[13] K. Zhang, L. Wang, and Y. Zhang, "Improved finite difference method with a compact correction term for solving Poisson's equations," Numer. Heat Transfer, Part B, vol. 70, no. 5, pp. 393–405, 2016.
[14] J. M. Tang and C. Vuik, "On deflation and singular symmetric positive semi-definite matrices," J. Comput. Appl. Math., vol. 206, no. 2, pp. 603–614, 2007.
[15] M. Yoon, G. Yoon, and C. Min, "On solving the singular system arisen from Poisson equation with Neumann boundary condition," J. Sci. Comput., vol. 69, no. 1, pp. 391–405, 2016.
[16] P. C. Chu and C. Fan, "A three-point combined compact difference scheme," J. Comput. Phys., vol. 140, no. 2, pp. 370–399, 1998.
[17] O. Axelsson, "A generalized conjugate direction method and its application on a singular perturbation problem," Numer. Anal. Lect. Notes Math., vol. 773, pp. 1–11, 1980.
[18] H. A. Van Der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems," SIAM J. Sci. Stat. Comput., vol. 13, no. 2, pp. 631–644, 1992.
[19] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," SIAM J. Sci. Stat. Comput., vol. 7, no. 3, pp. 856–869, 1986.
[20] T. F. Chan and P. C. Hansen, "Computing truncated singular value decomposition least squares solution by rank revealing QR-factorizations," SIAM J. Sci. Stat. Comput., vol. 11, no. 3, pp. 519–530, 1990.
[21] P. C. Hansen, "Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank," SIAM J. Sci. Stat. Comput., vol. 11, no. 3, pp. 503–518, 1990.
[22] E. Onunwor and L. Reichel, "On the computation of a truncated SVD of a large linear discrete ill-posed problem," Numer. Algorithms, vol. 75, no. 2, pp. 359–380, 2017.
[23] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," J. Soc. Ind. Appl. Math., vol. 3, no. 1, pp. 28–41, 1955.
[24] A. R. Mitchell and G. Fairweather, "Improved forms of the alternating direction methods of Douglas, Peaceman, and Rachford for solving parabolic and elliptic equations," Numer. Math., vol. 6, no. 1, pp. 285–292, 1964.
[25] S. Doss and K. Miller, "Dynamic ADI methods for elliptic equations," SIAM J. Numer. Anal., vol. 16, no. 5, pp. 837–856, 1979.
[26] S. S. Abarbanel, D. L. Dwoyer, and D. Gottlieb, "Improving the convergence rate to steady state of parabolic ADI methods," J. Comput. Phys., vol. 67, no. 1, pp. 1–18, 1986.
[27] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," Trans. Am. Math. Soc., vol. 82, no. 2, pp. 421–439, 1956.
[28] P. C. Chu and C. Fan, "A three-point sixth-order staggered combined compact difference scheme," Math. Comput. Modell., vol. 32, nos. 3–4, pp. 323–340, 2000.
[29] Z.-Z. Sun, "Compact difference schemes for heat equation with Neumann boundary conditions," Numer. Methods Partial Differ. Equations, vol. 25, no. 6, pp. 1320–1341, 2009.
[30] Y. G. D'yakonov, "Some difference schemes for solving boundary problems," USSR Comput. Math. Math. Phys., vol. 2, no. 1, pp. 55–77, 1963.
[31] H. W. Sun and L. Z. Li, "A CCD-ADI method for unsteady convection diffusion equations," Comput. Phys. Commun., vol. 185, no. 3, pp. 790–797, 2014.
[32] G. Fairweather and A. R. Mitchell, "Some computational results of an improved A.D.I. method for the Dirichlet problem," Comput. J., vol. 9, no. 3, pp. 298–303, 1966.
[33] J. Douglas and S. Kim, "Improved accuracy for locally one-dimensional methods for parabolic equations," Math. Models Methods Appl. Sci., vol. 11, no. 9, pp. 1563–1579, 2001.
[34] J. Zhao, W. Dai, and T. Niu, "Fourth-order compact schemes of a heat conduction problem with Neumann boundary conditions," Numer. Methods Partial Differ. Equations, vol. 23, no. 5, pp. 949–959, 2007.
[35] J. Zhao, W. Dai, and S. Zhang, "Fourth-order compact schemes for solving multidimensional heat problems with Neumann boundary conditions," Numer. Methods Partial Differ. Equations, vol. 24, no. 1, pp. 165–178, 2008.

Appendix

A derivation of the matrices used for spectral analysis

For the deduction of the matrices $[C_1]$ and $[D]$, the Eq. (10) is first considered in the form of

$$[A]_{(2N-1)\times(2N-1)}\left\{\begin{array}{c}\phi'\\ \phi''\end{array}\right\}_{(2N-1)} = [B]_{(2N-1)\times(N-1)}\{\phi\}_{(N-1)} + \left\{\begin{array}{ccccccc}\phi'_1 & \ldots & \phi'_N & 0 & \ldots & 0\end{array}\right\}^T$$

where the superscript T denotes the transpose. Therefore, we know that the solution vector is

$$\left\{\begin{array}{c}\phi'\\ \phi''\end{array}\right\} = [A]^{-1}[B]\{\phi\} + [A]^{-1}\{\alpha\}$$

where the vector $\{\alpha\}$ includes the Neumann boundary conditions for ϕ only. With the aid of a transform matrix $[T]$, we can calculate only the second-order derivatives from the following equation.

$$\begin{aligned}\{\phi''\}_{(N-1)} &= [T]_{(N-1)\times(2N-1)}\left\{\begin{array}{c}\phi'\\ \phi''\end{array}\right\}_{(2N-1)}\\ &= [T][A]^{-1}[B]\{\phi\} + [T][A]^{-1}\{\alpha\}\\ &= [C_1]\{\phi\} + [C_2]\{\alpha\}\end{aligned} \tag{A.36}$$

The form of the transformation matrix $[T]$ is shown below.

$$\left\{\begin{array}{c}\phi'_1{}'\\ \phi'_2{}'\\ \vdots\\ \phi'_{N-2}{}'\\ \phi'_{N-1}{}'\end{array}\right\} = \left[\begin{array}{cccccc|cccccc}0&0&\ldots&\ldots&0&0&1&0&\ldots&0&0\\ 0&0&\ldots&\ldots&0&0&0&1&\ldots&0&0\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots\\ 0&0&\ldots&\ldots&0&0&0&0&\ldots&1&0\\ 0&0&\ldots&\ldots&0&0&0&0&\ldots&0&1\end{array}\right]\left\{\begin{array}{c}\phi'_1\\ \phi'_2\\ \vdots\\ \vdots\\ \phi'_{N-1}\\ \phi'_N\\ \hline \phi'_1{}'\\ \phi'_2{}'\\ \vdots\\ \phi'_{N-2}{}'\\ \phi'_{N-1}{}'\end{array}\right\}$$

Hereafter, the one-dimensional parabolic equation is discretized with the Crank–Nicolson scheme to get the semi-discrete form.

$$\phi^{n+1} - r\phi^{n+1}_{xx} = \phi^n + r\phi^n_{xx} + \Delta t f^{n+\frac{1}{2}}$$

where $r = \Delta t/2$. The second-order derivative terms can be replaced with the relation shown in Eq. (A.36), thereby yielding

$$\left\{[I] - r[C_1]\right\}\{\phi\}^{n+1} - r[C_2]\{\alpha\}^{n+1} = \left\{[I] + r[C_1]\right\}\{\phi\}^n + r[C_2]\{\alpha\}^n + \Delta t\{f\}^{n+\frac{1}{2}}$$

Here, we introduce two auxiliary matrices $[C^-]$ and $[C^+]$ to simplify the equation, thereby yielding

$$[C^-]\{\phi\}^{n+1} = [C^+]\{\phi\}^n + r[C_2]\left\{\{\alpha\}^{n+1} + \{\alpha\}^n\right\} + \Delta t\{f\}^{n+\frac{1}{2}}$$

We can get finally the matrix form for the solution vector $\{\phi\}^{n+1}$ as follows.

$$\begin{aligned}\{\phi\}^{n+1} &= [C^-]^{-1}[C^+]\{\phi\}^n + r[C^-]^{-1}[C_2]\left\{\{\alpha\}^{n+1} + \{\alpha\}^n\right\} + \Delta t[C^-]^{-1}\{f\}^{n+\frac{1}{2}}\\ &= [D]\{\phi\}^n + r[E]\left\{\{\alpha\}^{n+1} + \{\alpha\}^n\right\} + \Delta t[C^-]^{-1}\{f\}^{n+\frac{1}{2}}\\ &= [D]\{\phi\}^n\end{aligned}$$