Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A POD-based reduced-order finite difference extrapolating model for the non-stationary incompressible Boussinesq equations

Zhendong Luo*

*Correspondence:
zhdluo@ncepu.edu.cn
School of Mathematics and Physics, North China Electric Power University, No. 2, Bei Nong Road, Changping District, Beijing, 102206, China

Abstract

A proper orthogonal decomposition (POD) method is used to establish a POD-based reduced-order finite difference (FD) extrapolating model with fully second-order accuracy for the non-stationary incompressible Boussinesq equations (NSIBEs). The error estimates of the POD-based reduced-order FD solutions obtained from the POD-based reduced-order FD extrapolating model are provided. The algorithm implementation for the POD-based reduced-order FD extrapolating model is given. A numerical experiment shows that the numerical results are consistent with the theoretical conclusions. Moreover, it is shown that the POD-based reduced-order FD extrapolating model is feasible and efficient for finding the numerical solutions for NSIBEs.

MSC: 76M20; 65M12; 65M15

Keywords: proper orthogonal decomposition; POD-based reduced-order finite difference extrapolating model; non-stationary incompressible Boussinesq equations; error estimate; numerical simulation

1 Introduction

Let $\Omega \subset R^2$ be a bounded and connected polygonal domain. Consider the following non-stationary incompressible Boussinesq equations (NSIBEs) including the velocity vector field and the pressure field as well as the temperature field (see [1, 2]).

Problem I Find $\mathbf{U} = (u, v), p,$ and T such that for $t_N > 0$,

$$\begin{cases}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, & (x,y,t) \in \Omega \times (0, t_N), \\
\frac{\partial u}{\partial t} + \frac{u\partial u}{\partial x} + \frac{v\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \gamma(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}), & (x,y,t) \in \Omega \times (0, t_N), \\
\frac{\partial v}{\partial t} + \frac{u\partial v}{\partial x} + \frac{v\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \gamma(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) + T, & (x,y,t) \in \Omega \times (0, t_N), \\
\frac{\partial T}{\partial t} + \frac{u\partial T}{\partial x} + \frac{v\partial T}{\partial y} = \frac{1}{\gamma_0}(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}), & (x,y,t) \in \Omega \times (0, t_N), \\
u(x,y,t) = \varphi_u(x,y,t), \qquad v(x,y,t) = \varphi_v(x,y,t), & (x,y,t) \in \partial\Omega \times (0, t_N), \\
T(x,y,t) = \varphi_T(x,y,t), & (x,y,t) \in \partial\Omega \times (0, t_N), \\
u(x,y,0) = u^0(x,y), \qquad v(x,y,0) = v^0(x,y), & (x,y) \in \Omega, \\
T(x,y,0) = T^0(x,y), & (x,y) \in \Omega,
\end{cases}$$

where $\mathbf{U} = (u, v)$ is the velocity vector, p the pressure, T the temperature, t_N the total time, Re the Reynolds number, Pr the Prandtl number, $\gamma = \sqrt{Pr/Re}$, $\gamma_0 = \sqrt{RePr}$, and $\varphi_u(x, y, t)$, $\varphi_v(x, y, t)$, $\varphi_T(x, y, t)$, $u^0(x, y)$, $v^0(x, y)$, and $T^0(x, y)$ all are given functions.

Problem I constitutes an important nonlinear system of equations in fluid dynamics. Because this system of equations does not only contain the velocity vector field as well as the pressure field but also includes the temperature field, especially, due to its nonlinearity, it is not easy to find its analytical solutions. One has to rely on numerical solutions. A classical finite difference (FD) scheme with fully second-order accuracy is one of the most effective high accuracy numerical methods for finding the numerical solutions of NSIBEs (see [2]). However, the classical FD scheme with fully second-order accuracy for NSIBEs generally includes many degrees of freedom so that it could bring about many difficulties in practical engineering computing. Especially, due to the truncation error accumulation in the computational process, the classical FD scheme with fully second-order accuracy may also appear to have no convergence after some computing steps. Thus, an extremely meaningful work is how to establish a reduced-order FD scheme with fewer degrees of freedom and sufficiently high accuracy so that it can reduce the truncation error accumulation, alleviating the computational load as well as saving time for calculations and resource demands in the computational process such that it can continuously simulate the development of the fluid flow.

The proper orthogonal decomposition (POD; see [3]) technique is one of the primary methods to construct low-order bases for the model reduction of complex and nonlinear problems. It has its roots in statistics (see [4]) and was initially used in the study of coherent structures in turbulent flows (see [5, 6]). Over the past three decades, the POD technique has been steadily developed (see [4–16]). Especially, in the past decade, it is used for model reduction of the numerical computational methods (for example, the Galerkin method, the FD scheme, the finite element method) to bring about a very good efficiency for finding the numerical solutions to partial differential equations (for example, see [17–26]). However, almost all existing POD-based reduced-order numerical methods employ the numerical solutions obtained from classical numerical methods on the total time span $[0, t_N]$ to construct the POD basis and POD-based reduced-order models, and then recompute the solutions on the same time span $[0, t_N]$, which is actually belong to repeating computations. Therefore, in this article, we thoroughly improve the existing methods, namely we do only employ the first few given classical FD numerical solutions with fully second-order accuracy for NSIBEs as snapshots on the very short time span $[0, t_0]$ ($t_0 \ll t_N$) to construct the POD basis and establish a POD-based reduced-order FD extrapolating model with fully second-order accuracy for finding the numerical solutions of NSIBEs on total time span $[0, t_N]$. Thus, we sufficiently adopt the advantage of POD method, namely utilize the given data (on the very short time span $[0, t_0]$) to predict future physic phenomena (on the time span $[t_0, t_N]$), which is why we name this reduced-order method as the POD-based reduced-order FD extrapolating model. So, the idea here has a very important guiding role if it is extended to numerical weather forecast.

Though a POD-based reduced-order extrapolating model with fully second-order FD scheme for non-stationary Burgers equation (see [27]) and a POD-based reduced-order FD extrapolating model with first-order time accuracy for the non-stationary Navier-Stokes equations (see [28]) have been presented too, NSIBEs are different from the non-stationary Burgers equation and the non-stationary Navier-Stokes equations as is mentioned in [17]. Namely, they are more complex than the non-stationary Burgers equation in [27] and the non-stationary Navier-Stokes equations without energy equation in [28]. They include a temperature field except for fluid velocity and pressure fields and there is a nonlinear coupling for the temperature field and fluid velocity field. Thus, establishing the POD-based reduced-order FD extrapolating model with fully second-order accuracy for NSIBEs has far more difficulties, but, more important, it is more serviceable, and more challenging than those for the non-stationary Burgers equation and the non-stationary Navier-Stokes equations. Though a POD-based reduced-order FD scheme with first-order time accuracy without adopting the extrapolation technique has been developed for NSIBEs (see [17]), it is also to test and verify only the comparison of solutions on the same time span $[0, t_N]$ and it belongs to repeating computations. The POD-based reduced-order FD extrapolating model with fully second-order accuracy here utilizes the given data (on the very short time span $[0, t_0]$ and $t_0 \ll t_N$) to predict future physic phenomena (on time span $[t_0, t_N]$) and has no repeating computations. Especially, based on the error estimates, we give guidance on the choice of the number of POD basis functions and a suggestion on when to renew the POD basis. This is a main improvement and innovation for method in [17].

This article is organized as follows. In Section 2 we derive a classical FD scheme with fully second-order accuracy for NSIBEs and generate snapshots from the first few numerical solutions obtained from the classical FD scheme with fully second-order accuracy. In Section 3 we construct orthonormal POD bases from the elements of the snapshots by means of the POD technique and establish the POD-based reduced-order FD extrapolating scheme with fully second-order accuracy and lower dimensions for NSIBEs. In Section 4, the error estimates of the reduced-order FD solutions obtained from the POD reduced-order FD extrapolating model are provided as guidance to choose the number of POD basis functions and renew the POD basis, and the algorithm implementation for the POD-based reduced-order FD extrapolating model. In Section 5, a numerical experiment is used to show that the numerical results are consistent with the theoretical conclusions and we validate that the POD-based reduced-order FD extrapolating model is feasible and efficient for solving NSIBEs. In Section 6 we provide the main conclusions and present discussions.

2 A classical FD scheme with fully second-order accuracy

In this section, we establish a classical FD scheme with fully second-order accuracy for Problem I. For the sake of convenience, without loss of generality, we may as well assume that the computational field $\Omega = [0, M] \times [0, \tilde{M}]$. Let Δx and Δy be the spatial step increments in the x and y direction, respectively, Δt be the time step increment, and $u^n_{j+\frac{1}{2},k}$, $v^n_{j,k+\frac{1}{2}}$ (or $T^n_{j,k+\frac{1}{2}}$) and $p^n_{j,k}$ denote the value of functions u, v (or T) and p at the points $(x_{j+\frac{1}{2}}, y_k, t_n)$, $(x_j, y_{k+\frac{1}{2}}, t_n)$ and (x_j, y_k, t_n) $(0 \le j \le J = [M/\Delta x], 0 \le k \le K = [\tilde{M}/\Delta y], 0 \le n \le N = [t_N/\Delta t])$, respectively. Then the classical FD scheme with fully second-order accuracy for Problem I is established as follows:

$$\frac{u^{n+1}_{j+\frac{1}{2},k} - u^{n+1}_{j-\frac{1}{2},k}}{\Delta x} + \frac{v^{n+1}_{j,k+\frac{1}{2}} - v^{n+1}_{j,k-\frac{1}{2}}}{\Delta y} = 0, \tag{1}$$

$$u^{n+1}_{j+\frac{1}{2},k} = u^{n-1}_{j+\frac{1}{2},k} + 2F^n_{j+\frac{1}{2},k} - \frac{2\Delta t}{\Delta x}\left(p^n_{j+1,k} - p^n_{j,k}\right), \tag{2}$$

$$v_{j,k+\frac{1}{2}}^{n+1} = v_{j,k+\frac{1}{2}}^{n-1} + 2G_{j,k+\frac{1}{2}}^{n} - \frac{2\Delta t}{\Delta x}\left(p_{j,k+1}^{n} - p_{j,k}^{n}\right) + \Delta t T_{j,k+\frac{1}{2}}^{n}, \tag{3}$$

$$T_{j,k+\frac{1}{2}}^{n+1} = T_{j,k+\frac{1}{2}}^{n-1} + 2H_{j,k+\frac{1}{2}}^{n}, \tag{4}$$

where

$$\begin{aligned} F_{j+\frac{1}{2},k}^{n} &= \gamma\Delta t\left[\frac{u_{j+\frac{1}{2},k-1} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{1}{2},k+1}}{\Delta y^2} + \frac{u_{j-\frac{1}{2},k} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{3}{2},k}}{\Delta x^2}\right]^{n} \\ &\quad - \frac{\Delta t}{\Delta x}u_{j+\frac{1}{2},k}^{n}\left(u_{j+1,k}^{n} - u_{j,k}^{n}\right) - \frac{\Delta t}{\Delta y}v_{j+\frac{1}{2},k}^{n}\left(u_{j+\frac{1}{2},k+\frac{1}{2}}^{n} - u_{j+\frac{1}{2},k-\frac{1}{2}}^{n}\right); \\ G_{j,k+\frac{1}{2}}^{n} &= \gamma\Delta t\left[\frac{v_{j-1,k+\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j+1,k+\frac{1}{2}}}{\Delta x^2} + \frac{v_{j,k-\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j,k+\frac{3}{2}}}{\Delta y^2}\right]^{n} \\ &\quad - \frac{\Delta t}{\Delta x}u_{j,k+\frac{1}{2}}^{n}\left(v_{j+\frac{1}{2},k+\frac{1}{2}}^{n} - v_{j-\frac{1}{2},k+\frac{1}{2}}^{n}\right) - \frac{\Delta t}{\Delta y}v_{j,k+\frac{1}{2}}^{n}\left(v_{j,k+1}^{n} - v_{j,k}^{n}\right); \\ H_{j,k+\frac{1}{2}}^{n} &= \frac{\Delta t}{\gamma_0}\left(\frac{T_{j-1,k+\frac{1}{2}}^{n} - 2T_{j,k+\frac{1}{2}}^{n} + T_{j+1,k+\frac{1}{2}}^{n}}{\Delta x^2} + \frac{T_{j,k-\frac{1}{2}}^{n} - 2T_{j,k+\frac{1}{2}}^{n} + T_{j,k+\frac{3}{2}}^{n}}{\Delta y^2}\right) \\ &\quad - \frac{\Delta t}{\Delta x}u_{j,k+\frac{1}{2}}^{n}\left(T_{j+\frac{1}{2},k+\frac{1}{2}}^{n} - T_{j-\frac{1}{2},k+\frac{1}{2}}^{n}\right) - \frac{\Delta t}{\Delta y}v_{j,k+\frac{1}{2}}^{n}\left(T_{j,k+1}^{n} - T_{j,k}^{n}\right). \end{aligned}$$

Inserting (2) and (3) into (1) one could obtain the approximate FD scheme of the Poisson equation for p as follows:

$$\left[\frac{p_{j-1,k} - 2p_{j,k} + p_{j+1,k}}{\Delta x^2} + \frac{p_{j,k-1} - 2p_{j,k} + p_{j,k+1}}{\Delta y^2}\right]^{n+1} = R, \tag{5}$$

where $R = \frac{2}{\Delta t\Delta x}[F_{j+\frac{1}{2},k} - F_{j-\frac{1}{2},k}]^{n} + \frac{2}{\Delta t\Delta y}[G_{j,k+\frac{1}{2}} - G_{j,k-\frac{1}{2}} + \Delta t(T_{j,k+\frac{1}{2}} - T_{j,k-\frac{1}{2}})]^{n}$.

Put $s = u, v, p, T$. Then the boundary values of the FD schemes (1)-(4) are as follows:

$$\begin{aligned} &s_{i,\frac{1}{2}}^{n} = \varphi_s(x_i, y_{\frac{1}{2}}, t_n), \quad (x_i, y_{\frac{1}{2}}) \in [0, M] \times \{0\}, i = \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, J + \frac{1}{2}; \\ &s_{i,K+\frac{1}{2}}^{n} = \varphi_s(x_i, y_{K+\frac{1}{2}}, t_n), \quad (x_i, y_{K+\frac{1}{2}}) \in [0, M] \times \{\tilde{M}\}, i = \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, J + \frac{1}{2}; \\ &s_{\frac{1}{2},i}^{n} = \varphi_s(x_{\frac{1}{2}}, y_i, t_n), \quad (x_{\frac{1}{2}}, y_i) \in \{0\} \times [0, \tilde{M}], i = \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, K + \frac{1}{2}; \\ &s_{J+\frac{1}{2},i}^{n} = \varphi_s(x_{J+\frac{1}{2}}, y_i, t_n), \quad (x_{J+\frac{1}{2}}, y_i) \in \{M\} \times [0, \tilde{M}], i = \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, K + \frac{1}{2}; \\ &s_{i,0}^{n} = 2s_{i,\frac{1}{2}}^{n} - s_{i,1}^{n}, \qquad s_{i,K+1}^{n} = 2s_{i,K+\frac{1}{2}}^{n} - s_{i,K}^{n}, \quad i = 0, \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, J + \frac{1}{2}, J; \\ &s_{0,i}^{n} = 2s_{\frac{1}{2},i}^{n} - s_{1,i}^{n}, \qquad s_{J+1,i}^{n} = 2s_{J+\frac{1}{2},i}^{n} - s_{J,i}^{n}, \quad i = 0, \frac{1}{2}, 1, 1 + \frac{1}{2}, \ldots, K + \frac{1}{2}, K. \end{aligned}$$

If $(|u_{j+\frac{1}{2},k}^{n}| + |v_{j,k+\frac{1}{2}}^{n}|)\Delta t \leq 8\gamma$ and $8\Delta t \leq \max\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\}$, by using the same approaches as the proof of the convergence and local stability of the FD equations of the non-stationary Navier-Stokes equation in [2], it is not difficult to prove the convergence and local stability for the FD equations (2)-(5). We conclude with the following result, whose proof is provided in the Appendix.

Theorem 1 *The classical FD scheme* (1)-(4) *with fully second-order accuracy for NSIBEs have following error estimates*:

$$\begin{aligned}&\bigl\|\bigl(u(x_{j+\frac{1}{2}},y_k,t_n),v(x_j,y_{k+\frac{1}{2}},t_n),T(x_j,y_{k+\frac{1}{2}},t_n),p(x_j,y_k,t_n)\bigr)-\bigl(u^n_{j+\frac{1}{2},k},v^n_{j,k+\frac{1}{2}},T^n_{j,k+\frac{1}{2}},p^n_{j,k}\bigr)\bigr\|\\&\quad=O\bigl(\Delta t^2,\Delta x^2,\Delta y^2\bigr),\quad n=1,2,\ldots,N,\end{aligned}\tag{6}$$

where $\|\cdot\|$ *denotes usual norm of vector.*

If only the Reynolds number Re; the Prandtl number Pr; the boundary value functions $\varphi_u(x,y,t)$, $\varphi_v(x,y,t)$, and $\varphi_T(x,y,t)$; the initial value functions $u^0(x,y)$, $v^0(x,y)$, and $T^0(x,y)$, and the time step increment Δt and the spatial step increments Δx and Δy are given, by solving the FD schemes (2)-(5), we can obtain the classical FD solutions $u^n_{j+\frac{1}{2},k}$, $v^n_{j,k+\frac{1}{2}}$, $T^n_{j,k+\frac{1}{2}}$, and $p^n_{j,k}$ ($0\le j\le J$, $0\le k\le K$, $1\le n\le N$) of Problem I.

Put $u^n_i=u^n_{j+\frac{1}{2},k}$, $v^n_i=v^n_{j,k+\frac{1}{2}}$, $T^n_i=T^n_{j,k+\frac{1}{2}}$, and $p^n_i=p^n_{j,k}$ ($i=kJ+j+1$, $1\le i\le m$, $m=JK$, $0\le j\le J-1$, $0\le k\le K-1$), respectively. We may choose the first L group of solutions to construct a set $\{u^l_i,v^l_i,T^l_i,p^l_i\}^L_{l=1}$ ($1\le i\le m$, $L\ll N$) including $L\times m$ elements from the set $\{u^n_i,v^n_i,T^n_i,p^n_i\}^N_{n=1}$ ($1\le i\le m$) including $N\times m$ elements, which are known as snapshots.

3 POD-based reduced-order FD extrapolating model

In this section, we use snapshots extracted from the first fewer numerical solutions of the classical FD schemes (2)-(5) to reconstruct orthonormal POD bases and establish the POD-based reduced-order FD extrapolating model with fully second-order accuracy and lower dimensions.

The sets of snapshots $\{u^l_i,v^l_i,T^l_i,p^l_i\}^L_{l=1}$ ($1\le i\le m$) in Section 2 can constitute four $m\times L$ matrices $\mathbf{A}_s=(s^l_i)_{m\times L}$ ($s=u,v,p,T$) denoted by

$$\mathbf{A}_s=\begin{pmatrix}s^1_1&s^2_1&\cdots&s^L_1\\s^1_2&s^2_2&\cdots&s^L_2\\\vdots&\vdots&\ddots&\vdots\\s^1_m&s^2_m&\cdots&s^L_m\end{pmatrix}.$$

Since the order m for matrices $\mathbf{A}_s\mathbf{A}_s^T$ is far larger than the order L for matrices $\mathbf{A}_s^T\mathbf{A}_s$, however, their positive eigenvalues are identical, therefore, we may first solve the eigenequation corresponding to matrices $\mathbf{A}_s^T\mathbf{A}_s$ to find the eigenvalues $\lambda_{s1}\ge\lambda_{s2}\ge\cdots\ge\lambda_{s\tilde{M}_s}>0$ ($\tilde{M}_s=\operatorname{rank}\mathbf{A}_s$) and the corresponding eigenvectors $\boldsymbol{\varphi}_{sj}$, and then by the relationship

$$\boldsymbol{\phi}_{sj}=\mathbf{A}_s\boldsymbol{\varphi}_{sj}/\sqrt{\lambda_{sj}},\quad j=1,2,\ldots,\tilde{M}_s,s=u,v,p,T,$$

we may obtain the eigenvectors $\boldsymbol{\phi}_{sj}$ ($j=1,2,\ldots,\tilde{M}_s$, $s=u,v,p,T$) corresponding to the nonzero eigenvalues for the matrix $\mathbf{A}_s\mathbf{A}_s^T$.

Taking the first M_s ($0<M_s\le\tilde{M}_s\le L$) columns from four eigenmatrices $\mathbf{U}_s=(\boldsymbol{\phi}_{s1},\boldsymbol{\phi}_{s2},\ldots,\boldsymbol{\phi}_{s\tilde{M}_s})$, we construct the four orthonormal POD basis functions (see [17]) $\boldsymbol{\Phi}_s=(\boldsymbol{\phi}_{s1},\boldsymbol{\phi}_{s2},\ldots,\boldsymbol{\phi}_{sM_s})$ ($s=u,v,p,T$).

Write

$$\mathbf{s}^n_m=\bigl(s^n_1,s^n_2,\ldots,s^n_m\bigr)^T,\quad n=1,2,\ldots,N,s=u,v,p,T.\tag{7}$$

Then we have the following error estimates (see [17]):

$$\left\|\mathbf{s}_m^l - \boldsymbol{\Phi}_s\boldsymbol{\Phi}_s^T\mathbf{s}_m^l\right\| \le \sqrt{\lambda_{s(M_s+1)}}, \quad l = 1,2,\ldots,L, s = u,v,p,T. \tag{8}$$

Thus, (2)-(5) are written as the following vector scheme:

$$\begin{aligned}&\left(\mathbf{u}_m^{n+1},\mathbf{v}_m^{n+1},\mathbf{T}_m^{n+1},\mathbf{p}_m^{n+1}\right)^T\\ &\quad = \left(\mathbf{u}_m^{n-1},\mathbf{v}_m^{n-1},\mathbf{T}_m^{n-1},\mathbf{p}_m^{n-1}\right)^T + \tilde{F}\left(\mathbf{u}_m^n,\mathbf{v}_m^n,\mathbf{T}_m^n,\mathbf{p}_m^n\right), \quad n = 1,2,\ldots,N-1,\end{aligned} \tag{9}$$

where $\tilde{F}$ is determined from (2)-(5). Put

$$\left(\mathbf{u}_m^{*n},\mathbf{v}_m^{*n},\mathbf{T}_m^{*n},\mathbf{p}_m^{*n}\right)^T = \left(\boldsymbol{\Phi}_u\boldsymbol{\alpha}_{M_u}^n,\boldsymbol{\Phi}_v\boldsymbol{\beta}_{M_v}^n,\boldsymbol{\Phi}_T\boldsymbol{\delta}_{M_T}^n,\boldsymbol{\Phi}_p\boldsymbol{\epsilon}_{M_p}^n\right)^T, \tag{10}$$

where $\mathbf{u}_m^{*n} = (u_1^{*n},u_2^{*n},\ldots,u_m^{*n})^T$, $\mathbf{v}_m^{*n} = (v_1^{*n},v_2^{*n},\ldots,v_m^{*n})^T$, $\mathbf{T}_m^{*n} = (T_1^{*n},T_2^{*n},\ldots,T_m^{*n})^T$, and $\mathbf{p}_m^{*n} = (p_1^{*n},p_2^{*n},\ldots,p_m^{*n})^T$ are four column vectors corresponding to u, v, T, and p, respectively. If $\mathbf{u}_m^n$, $\mathbf{v}_m^n$, $\mathbf{T}_m^n$, and $\mathbf{p}_m^n$ in (9) are approximately replaced with $\mathbf{u}_m^{*n}$, $\mathbf{v}_m^{*n}$, $\mathbf{T}_m^{*n}$, and $\mathbf{p}_m^{*n}$ in (10) ($n = 0,1,2,\ldots,N$), by noting that four matrices $\boldsymbol{\Phi}_u$, $\boldsymbol{\Phi}_v$, $\boldsymbol{\Phi}_T$, and $\boldsymbol{\Phi}_p$ are formed with the orthonormal vectors, the POD-based reduced-order FD extrapolating model with fully second-order accuracy for Problem I, which includes only $M_u + M_v + M_T + M_p$ ($M_u,M_v,M_T,M_p \ll L \ll m$) unknown values on each time level and has no repeating computations, is denoted by

$$\begin{aligned}&\boldsymbol{\alpha}_{M_u}^n = \boldsymbol{\Phi}_u^T\mathbf{u}_m^n, \qquad \boldsymbol{\beta}_{M_v}^n = \boldsymbol{\Phi}_v^T\mathbf{v}_m^n, \qquad \boldsymbol{\delta}_{M_T}^n = \boldsymbol{\Phi}_T^T\mathbf{T}_m^n,\\ &\boldsymbol{\epsilon}_{M_p}^n = \boldsymbol{\Phi}_p^T\mathbf{p}_m^n, \quad 1 \le n \le L;\end{aligned} \tag{11}$$

$$\begin{aligned}&\left(\boldsymbol{\alpha}_{M_u}^{n+1},\boldsymbol{\beta}_{M_v}^{n+1},\boldsymbol{\delta}_{M_T}^{n+1},\boldsymbol{\epsilon}_{M_p}^{n+1}\right)^T = \left(\boldsymbol{\alpha}_{M_u}^{n-1},\boldsymbol{\beta}_{M_v}^{n-1},\boldsymbol{\delta}_{M_T}^{n-1},\boldsymbol{\epsilon}_{M_p}^{n-1}\right)^T + \left(\boldsymbol{\Phi}_u^T,\boldsymbol{\Phi}_v^T,\boldsymbol{\Phi}_T^T,\boldsymbol{\Phi}_p^T\right)^T\\ &\qquad\qquad \times \tilde{F}\left(\boldsymbol{\Phi}_u\boldsymbol{\alpha}_{M_u}^n,\boldsymbol{\Phi}_v\boldsymbol{\beta}_{M_v}^n,\boldsymbol{\Phi}_T\boldsymbol{\delta}_{M_T}^n,\boldsymbol{\Phi}_p\boldsymbol{\epsilon}_{M_p}\right), \quad L \le n \le N-1.\end{aligned} \tag{12}$$

If $\boldsymbol{\alpha}_{M_u}^n$, $\boldsymbol{\beta}_{M_v}^n$, $\boldsymbol{\delta}_{M_T}^n$, and $\boldsymbol{\epsilon}_{M_p}^n$ are obtained from (11) and (12), the solutions for the POD-based reduced-order FD extrapolating model are given by

$$\begin{aligned}&\mathbf{u}_m^{*n} = \boldsymbol{\Phi}_u\boldsymbol{\alpha}_{M_u}^n, \qquad \mathbf{v}_m^{*n} = \boldsymbol{\Phi}_v\boldsymbol{\beta}_{M_v}^n, \qquad \mathbf{T}_m^{*n} = \boldsymbol{\Phi}_T\boldsymbol{\delta}_{M_T}^n,\\ &\mathbf{p}_m^{*n} = \boldsymbol{\Phi}_p\boldsymbol{\epsilon}_{M_p}, \quad n = 1,2,\ldots,N.\end{aligned} \tag{13}$$

Further, the component forms of the solutions for the POD-based reduced-order FD extrapolating model are denoted by $u_{j+\frac{1}{2},k}^{*n} = u_i^{*n}$, $v_{j,k+\frac{1}{2}}^{*n} = v_i^{*n}$, $T_{j,k+\frac{1}{2}}^{*n} = T_i^{*n}$, and $p_{j,k}^{*n} = p_i^{*n}$ ($0 \le j \le J-1$, $0 \le k \le K-1$, $i = k(J+1)+j+1$, $1 \le i \le m = KJ$).

Remark 1 Since the classical FD schemes (2)-(5) on each time level include $4m$ unknown quantities, while the system of equations (11)-(13) on each time level (when $n > L$) only includes $M_u + M_v + M_T + M_p$ unknown quantities ($M_u,M_v,M_T,M_p \ll L \ll m$, for example, in Section 5, $L = 20$, $M_u = M_v = M_T = M_p = 6$, but $m = 10^4$), so the system of equations (11)-(13) is the POD-based reduced-order FD extrapolating model with very few degrees of freedom but without repeating computations and is completely different from existing POD-based reduced-order models (for example, see [17–26]). Though we draw the snapshots from the first L classical FD solutions with fully second-order accuracy in this article, in fact, if one computes actual problems, one may obtain the ensemble of snapshots by drawing samples from experiments of physical system trajectories.

4 Error analysis and algorithm implementation for the POD-based reduced-order FD extrapolating algorithm

In this section, the error estimates of the solutions of the POD-based reduced-order FD extrapolating model are first provided for guidance to choose the number of POD basis functions and renew the POD basis, and then the algorithm implementation for the POD-based reduced-order FD extrapolating model is given.

4.1 Error estimates and criterion of renewing POD basis

It is obvious that when $n = L+1, L+2, \ldots, N$, by (13), (12) can be written as the following form like (1)-(4):

$$\frac{u^{*n+1}_{j+\frac{1}{2},k} - u^{*n+1}_{j-\frac{1}{2},k}}{\Delta x} + \frac{v^{*n+1}_{j,k+\frac{1}{2}} - v^{*n+1}_{j,k-\frac{1}{2}}}{\Delta y} = 0, \tag{14}$$

$$u^{*n+1}_{j+\frac{1}{2},k} = u^{*n-1}_{j+\frac{1}{2},k} + 2F^{*n}_{j+\frac{1}{2},k} - \frac{2\Delta t}{\Delta x}\left(p^{*n}_{j+1,k} - p^{*n}_{j,k}\right), \tag{15}$$

$$v^{*n+1}_{j,k+\frac{1}{2}} = v^{*n-1}_{j,k+\frac{1}{2}} + 2G^{*n}_{j,k+\frac{1}{2}} - \frac{2\Delta t}{\Delta x}\left(p^{*n}_{j,k+1} - p^{*n}_{j,k}\right) + \Delta t T^{*n}_{j,k+\frac{1}{2}}, \tag{16}$$

$$T^{*n+1}_{j,k+\frac{1}{2}} = T^{*n-1}_{j,k+\frac{1}{2}} + H^{*n}_{j,k+\frac{1}{2}}, \tag{17}$$

where

$$\begin{aligned}
F^{*n}_{j+\frac{1}{2},k} = {}& \gamma\,\Delta t\left[\frac{u_{j+\frac{1}{2},k-1} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{1}{2},k+1}}{\Delta y^2} + \frac{u_{j-\frac{1}{2},k} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{3}{2},k}}{\Delta x^2}\right]^{*n} \\
& - \frac{\Delta t}{\Delta x} u^{*n}_{j+\frac{1}{2},k}\left(u^{*n}_{j+1,k} - u^{*n}_{j,k}\right) - \frac{\Delta t}{\Delta y} v^{*n}_{j+\frac{1}{2},k}\left(u^{*n}_{j+\frac{1}{2},k+\frac{1}{2}} - u^{*n}_{j+\frac{1}{2},k-\frac{1}{2}}\right); \\
G^{*n}_{j,k+\frac{1}{2}} = {}& \gamma\,\Delta t\left[\frac{v_{j-1,k+\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j+1,k+\frac{1}{2}}}{\Delta x^2} + \frac{v_{j,k-\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j,k+\frac{3}{2}}}{\Delta y^2}\right]^{*n} \\
& - \frac{\Delta t}{\Delta x} u^{*n}_{j,k+\frac{1}{2}}\left(v^{*n}_{j+\frac{1}{2},k+\frac{1}{2}} - v^{*n}_{j-\frac{1}{2},k+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta y} v^{*n}_{j,k+\frac{1}{2}}\left(v^{*n}_{j,k+1} - v^{*n}_{j,k}\right); \\
H^{*n}_{j,k+\frac{1}{2}} = {}& \frac{\Delta t}{\gamma_0}\left(\frac{T^{*n}_{j-1,k+\frac{1}{2}} - 2T^{*n}_{j,k+\frac{1}{2}} + T^{*n}_{j+1,k+\frac{1}{2}}}{\Delta x^2} + \frac{T^{*n}_{j,k-\frac{1}{2}} - 2T^{*n}_{j,k+\frac{1}{2}} + T^{*n}_{j,k+\frac{3}{2}}}{\Delta y^2}\right) \\
& - \frac{\Delta t}{\Delta x} u^{*n}_{j,k+\frac{1}{2}}\left(T^{*n}_{j+\frac{1}{2},k+\frac{1}{2}} - T^{*n}_{j-\frac{1}{2},k+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta y} v^{*n}_{j,k+\frac{1}{2}}\left(T^{*n}_{j,k+1} - T^{*n}_{j,k}\right).
\end{aligned}$$

It is obvious that the stability condition of (14)-(17) is also $(|u^{*n}_{j+\frac{1}{2},k}| + |v^{*n}_{j,k+\frac{1}{2}}|)\Delta t \le 8\gamma$ and $8\Delta t \le \max\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\}$.

Equations (12) and (13) can also be rewritten as vector form denoted by

$$\begin{aligned}
\left(\mathbf{u}^{*n+1}_m, \mathbf{v}^{*n+1}_m, \mathbf{T}^{*n+1}_m, \mathbf{p}^{*n+1}_m\right)^T = {}& \left(\mathbf{u}^{*n-1}_m, \mathbf{v}^{*n-1}_m, \mathbf{T}^{*n-1}_m, \mathbf{p}^{*n-1}_m\right)^T \\
& + \tilde{F}\left(\mathbf{u}^{*n}_m, \mathbf{v}^{*n}_m, \mathbf{T}^{*n}_m, \mathbf{p}^{*n}_m\right), \quad n = L, L+1, L+2, \ldots, N.
\end{aligned} \tag{18}$$

Let $\mathbf{e}_n = (\mathbf{u}^n_m, \mathbf{v}^n_m, \mathbf{T}^n_m, \mathbf{p}^n_m)^T - (\mathbf{u}^{*n}_m, \mathbf{v}^{*n}_m, \mathbf{T}^{*n}_m, \mathbf{p}^{*n}_m)^T$. Subtracting (11) and (13) from (9) yields

$$\mathbf{e}_n = \left(\mathbf{u}^n, \mathbf{v}^n, \mathbf{T}^n_m, \mathbf{p}^n\right)^T - \left(\mathbf{\Phi}_u\mathbf{\Phi}_u^T\mathbf{u}^n, \mathbf{\Phi}_v\mathbf{\Phi}_v^T\mathbf{v}^n, \mathbf{\Phi}_T\mathbf{\Phi}_T^T\mathbf{T}^n, \mathbf{\Phi}_p\mathbf{\Phi}_p^T\mathbf{p}^n\right)^T, \quad 1 \le n \le L. \tag{19}$$

Subtracting (18) from (9) yields

$$\mathbf{e}_{n+1} = \mathbf{e}_{n-1} + \tilde{F}\big(\mathbf{u}_m^n, \mathbf{v}_m^n, \mathbf{T}_m^n, \mathbf{p}_m^n\big) - \tilde{F}\big(\mathbf{u}_m^{*n}, \mathbf{v}_m^{*n}, \mathbf{T}_m^{*n}, \mathbf{p}_m^{*n}\big), \quad L \le n \le N-1. \tag{20}$$

By (8) and (19), we have the following error estimates:

$$\|\mathbf{e}_n\| \le \sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}, \quad 1 \le n \le L. \tag{21}$$

Let $M = \max\{\Delta t(|u^n_{j+\frac{1}{2},k}| + |v^n_{j,k+\frac{1}{2}}|)/(8\gamma), \Delta t(|u^{*n}_{j+\frac{1}{2},k}| + |v^{*n}_{j,k+\frac{1}{2}}|)/(8\gamma), 8\Delta t/\min\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\}\}$. Based on the stability conditions of (1)-(4) and (14)-(17), we have $M \le 1/2$. Therefore, from (1)-(4) and (14)-(17), we obtain

$$\begin{aligned} \|\mathbf{e}_{n+1}\| &\le \|\mathbf{e}_{n-1}\| + \big\|\tilde{F}\big(\mathbf{u}_m^n, \mathbf{v}_m^n, \mathbf{T}_m^n, \mathbf{p}_m^n\big) - \tilde{F}\big(\mathbf{u}_m^{*n}, \mathbf{v}_m^{*n}, \mathbf{T}_m^{*n}, \mathbf{p}_m^{*n}\big)\big\| \\ &\le \|\mathbf{e}_{n-1}\| + M\|\mathbf{e}_n\|, \quad L \le n \le N-1. \end{aligned} \tag{22}$$

Summing (22) from L to $n-1$ yields

$$\|\mathbf{e}_n\| \le \|\mathbf{e}_{L-1}\| + \|\mathbf{e}_L\| + M\sum_{i=L}^{n-1}\|\mathbf{e}_i\|, \quad L \le n \le N-1. \tag{23}$$

Put $\xi_n = M\sum_{i=L}^{n-1}\|\mathbf{e}_i\| + \|\mathbf{e}_{L-1}\| + \|\mathbf{e}_L\|$. Then, from (23), we obtain $\|\mathbf{e}_n\| \le \xi_n$ and $\xi_n - \xi_{n-1} = M\|\mathbf{e}_{n-1}\|$ ($n \ge L \ge 2$). Thus, we have from (21) that

$$\begin{aligned} \|\mathbf{e}_n\| &\le \xi_n \le (M+1)\xi_{n-1} \le (M+1)^2\xi_{n-2} \le \cdots \le (M+1)^{n-L}\xi_L \\ &= (M+1)^{n-L}\big(\|\mathbf{e}_{L-1}\|_2 + \|\mathbf{e}_L\|_2\big) \\ &\le 2(M+1)^{n-L}\big[\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}\big], \\ &\quad L+1 \le n \le N. \end{aligned} \tag{24}$$

Since the absolute value of each component for vector is not more than its norm, by combining Theorem 1 with (21) and (24), we have the following result.

Theorem 2 *Based on the stability conditions* $(|u^n_{j+\frac{1}{2},k}| + |v^n_{j,k+\frac{1}{2}}|)\Delta t \le 8\gamma$, $(|u^{*n}_{j+\frac{1}{2},k}| + |v^{*n}_{j,k+\frac{1}{2}}|)\Delta t \le 8\gamma$, *and* $8\Delta t \le \max\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\}$ *of* (1)-(4) *and* (14)-(17), *the error estimates between the solution for NSIBEs and the solutions obtained from the POD-based reduced-order FD extrapolating model* (11)-(13) *are denoted by*

$$\begin{aligned} &\big|u(x_{j+\frac{1}{2}}, y_k, t_n) - u^{*n}_{j+\frac{1}{2},k}\big| + \big|v(x_j, y_{k+\frac{1}{2}}, t_n) - v^{*n}_{j,k+\frac{1}{2}}\big| \\ &\quad + \big|T(x_j, y_{k+\frac{1}{2}}, t_n) - T^{*n}_{j,k+\frac{1}{2}}\big| + \big|p(x_j, y_k, t_n) - p^{*n}_{j,k}\big| \\ &= O\big(C_n(M), \Delta t^2, \Delta x^2, \Delta y^2\big), \quad 1 \le n \le N, \end{aligned} \tag{25}$$

where $C_n(M) = \sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}$ $(1 \le n \le L)$, $C_n(M) = 2(1+M)^{n-L}[\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}]$ $(L+1 \le n \le N)$, *and* $M = \max\{(|u^n_{j+\frac{1}{2},k}| + |v^n_{j,k+\frac{1}{2}}|)\Delta t\gamma^{-1}/8, (|u^{*n}_{j+\frac{1}{2},k}| + |v^{*n}_{j,k+\frac{1}{2}}|)\Delta t\gamma^{-1}/8, 8\Delta t/\min\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\}\}$.

Remark 2 The error estimates in Theorem 2 provide guidance for choosing the number of POD basis functions, namely, we should take M_u, M_v, M_T, and M_p such that $\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}} = O(\Delta t^2, \Delta x^2, \Delta y^2)$. $C_n(M)$ $(L+1 \le n \le N)$ are caused by extrapolating iteration and may act as the criterion for renewing the POD basis, namely, if $C_n(M) > \max(\Delta t^2, \Delta x^2, \Delta y^2)$, the old POD basis is substituted with the new POD basis regenerated from new snapshots.

4.2 Algorithm implementation for the POD-based reduced-order FD extrapolating model

The algorithm implementation for the POD-based reduced-order FD extrapolating model (11)-(13) consists of the following five steps.

Step 1. For given the Reynolds number Re, the Prandtl number Pr, boundary value functions $\varphi_u(x,y,t)$, $\varphi_v(x,y,t)$, and $\varphi_T(x,y,t)$, initial value functions $u^0(x,y)$, $v^0(x,y)$, and $T^0(x,y)$, the time step increment Δt, and the spatial step increments Δx and Δy, let $u^1(x,y) = u^0(x,y)$, $v^1(x,y) = v^0(x,y)$, and $T^1(x,y) = T^0(x,y)$, solving the following classical FD scheme, for $n = 1, 2, \ldots, L$:

$$
\begin{aligned}
F^n_{j+\frac{1}{2},k} &= \gamma\,\Delta t\left[\frac{u_{j+\frac{1}{2},k-1} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{1}{2},k+1}}{\Delta y^2} + \frac{u_{j-\frac{1}{2},k} - 2u_{j+\frac{1}{2},k} + u_{j+\frac{3}{2},k}}{\Delta x^2}\right]^n \\
&\quad - \frac{\Delta t}{\Delta x} u^n_{j+\frac{1}{2},k}\left(u^n_{j+1,k} - u^n_{j,k}\right) - \frac{\Delta t}{\Delta y} v^n_{j+\frac{1}{2},k}\left(u^n_{j+\frac{1}{2},k+\frac{1}{2}} - u^n_{j+\frac{1}{2},k-\frac{1}{2}}\right); \\
G^n_{j,k+\frac{1}{2}} &= \gamma\,\Delta t\left[\frac{v_{j-1,k+\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j+1,k+\frac{1}{2}}}{\Delta x^2} + \frac{v_{j,k-\frac{1}{2}} - 2v_{j,k+\frac{1}{2}} + v_{j,k+\frac{3}{2}}}{\Delta y^2}\right]^n \\
&\quad - \frac{\Delta t}{\Delta x} u^n_{j,k+\frac{1}{2}}\left(v^n_{j+\frac{1}{2},k+\frac{1}{2}} - v^n_{j-\frac{1}{2},k+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta y} v^n_{j,k+\frac{1}{2}}\left(v^n_{j,k+1} - v^n_{j,k}\right); \\
H^n_{j,k+\frac{1}{2}} &= \frac{\Delta t}{\gamma_0}\left(\frac{T^n_{j-1,k+\frac{1}{2}} - 2T^n_{j,k+\frac{1}{2}} + T^n_{j+1,k+\frac{1}{2}}}{\Delta x^2} + \frac{T^n_{j,k-\frac{1}{2}} - 2T^n_{j,k+\frac{1}{2}} + T^n_{j,k+\frac{3}{2}}}{\Delta y^2}\right) \\
&\quad - \frac{\Delta t}{\Delta x} u^n_{j,k+\frac{1}{2}}\left(T^n_{j+\frac{1}{2},k+\frac{1}{2}} - T^n_{j-\frac{1}{2},k+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta y} v^n_{j,k+\frac{1}{2}}\left(T^n_{j,k+1} - T^n_{j,k}\right), \\
R &= \frac{F^n_{j+\frac{1}{2},k} - F^n_{j-\frac{1}{2},k}}{\Delta t\Delta x} + \frac{G^n_{j,k+\frac{1}{2}} - G^n_{j,k-\frac{1}{2}} + \Delta t(T^n_{j,k+\frac{1}{2}} - T^n_{j,k-\frac{1}{2}})}{\Delta t\Delta y}, \\
&\left[\frac{p_{j-1,k} - 2p_{j,k} + p_{j+1,k}}{\Delta x^2} + \frac{p_{j,k-1} - 2p_{j,k} + p_{j,k+1}}{\Delta y^2}\right]^n = R, \\
u^{n+1}_{j+\frac{1}{2},k} &= u^{n-1}_{j+\frac{1}{2},k} + 2F^n_{j+\frac{1}{2},k} - \frac{2\Delta t}{\Delta x}\left(p^n_{j+1,k} - p^n_{j,k}\right), \\
v^{n+1}_{j,k+\frac{1}{2}} &= v^{n-1}_{j,k+\frac{1}{2}} + 2G^n_{j,k+\frac{1}{2}} - \frac{2\Delta t}{\Delta x}\left(p^n_{j,k+1} - p^n_{j,k}\right) + \Delta t T^n_{j,k+\frac{1}{2}}, \\
T^{n+1}_{j,k+\frac{1}{2}} &= T^{n-1}_{j,k+\frac{1}{2}} + H^n_{j,k+\frac{1}{2}}
\end{aligned}
$$

yields L (with usually $L = 20$) groups of classical FD solutions $u^n_{j+\frac{1}{2},k}$, $v^n_{j,k+\frac{1}{2}}$, $T^n_{j,k+\frac{1}{2}}$, and $p^n_{j,k}$ $(0 \le j \le J, 0 \le k \le K, 1 \le n \le L)$, further, constructing a set of snapshots $\{u^l_i, v^l_i, T^l_i, p^l_i\}^L_{l=1}$ $(1 \le i \le m)$ with $L \times m$ elements (for actual engineering problems, the ensemble of snapshots is obtained from physical system trajectories by drawing samples from experiments), where $u^n_i = u^n_{j+\frac{1}{2},k}$, $v^n_i = v^n_{j,k+\frac{1}{2}}$, $T^n_i = T^n_{j,k+\frac{1}{2}}$, and $p^n_i = p^n_{j,k}$ $(i = kJ + j + 1, 1 \le i \le m, m = JK, 0 \le j \le J - 1, 0 \le k \le K - 1)$, respectively.

Step 2. Form the snapshot matrices $\mathbf{A}_s = (s_i^l)_{m\times L}$ $(s = u, v, T, p)$ and solve the linear systems of equations $(\mathbf{A}_s^T\mathbf{A}_s - \lambda_s\mathbf{I}_L)\boldsymbol{\varphi}_s = \mathbf{0}$ obtaining the eigenvalues $\lambda_{s1} \geq \lambda_{s2} \geq \cdots \geq \lambda_{s\tilde{M}_s} > 0$ $(\tilde{M}_s = \operatorname{rank}\mathbf{A}_s,\ s = u, v, T, p)$ and corresponding eigenvectors $\boldsymbol{\varphi}_{sj}$ $(j = 1, 2, \ldots, \tilde{M}_s,\ s = u, v, T, p)$.

Step 3. For the error $\mu = O(\Delta t^2, \Delta x^2, \Delta y^2)$ needed, determine the numbers M_s $(M_s \leq \tilde{M}_s,\ s = u, v, T, p)$ of POD basis functions such that $\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}} \leq \mu$, and construct the POD basis $\boldsymbol{\Phi}_s = (\boldsymbol{\phi}_{s1}, \boldsymbol{\phi}_{s2}, \ldots, \boldsymbol{\phi}_{sM_s})$ (where $\boldsymbol{\phi}_{sj} = \mathbf{A}_s\boldsymbol{\varphi}_{sj}/\sqrt{\lambda_{sj}}$, $j = 1, 2, \ldots, M_s$, $s = u, v, T, p$).

Step 4. Write $\mathbf{u}_m^n = (u_1^n, u_2^n, \ldots, u_m^n)^T$, $\mathbf{v}_m^n = (v_1^n, v_2^n, \ldots, v_m^n)^T$, $\mathbf{T}_m^n = (T_1^n, T_2^n, \ldots, T_m^n)^T$, and $\mathbf{p}_m^n = (p_1^n, p_2^n, \ldots, p_m^n)^T$ $(0 \leq n \leq N)$. Solve the following POD-based reduced-order FD extrapolating model with fully second-order accuracy:

$$\begin{aligned}
&\boldsymbol{\alpha}_{M_u}^n = \boldsymbol{\Phi}_u^T\mathbf{u}_m^n, \qquad \boldsymbol{\beta}_{M_v}^n = \boldsymbol{\Phi}_v^T\mathbf{v}_m^n, \qquad \boldsymbol{\delta}_{M_T}^n = \boldsymbol{\Phi}_T^T\mathbf{T}_m^n, \qquad \boldsymbol{\epsilon}_{M_p}^n = \boldsymbol{\Phi}_p^T\mathbf{p}_m^n, \quad n = 1, 2, \ldots, L; \\
&\bigl(\boldsymbol{\alpha}_{M_u}^{n+1}, \boldsymbol{\beta}_{M_v}^{n+1}, \boldsymbol{\delta}_{M_p}^{n+1}, \boldsymbol{\epsilon}_{M_p}^{n+1}\bigr)^T = \bigl(\boldsymbol{\alpha}_{M_u}^{n-1}, \boldsymbol{\beta}_{M_v}^{n-1}, \boldsymbol{\delta}_{M_p}^{n-1}, \boldsymbol{\epsilon}_{M_p}^{n-1}\bigr)^T \\
&\qquad\qquad\qquad + \tilde{G}\bigl(\boldsymbol{\alpha}_{M_u}^n, \boldsymbol{\beta}_{M_v}^n, \boldsymbol{\delta}_{M_p}^n, \boldsymbol{\epsilon}_{M_p}^n\bigr), \quad n = L, L+1, L+2, \ldots, N; \\
&\mathbf{u}_m^{*n} = \boldsymbol{\Phi}_u\boldsymbol{\alpha}_{M_u}^n, \qquad \mathbf{v}_m^{*n} = \boldsymbol{\Phi}_v\boldsymbol{\beta}_{M_v}^n, \qquad \mathbf{T}_m^{*n} = \boldsymbol{\Phi}_T\boldsymbol{\delta}_{M_T}, \\
&\mathbf{p}_m^{*n} = \boldsymbol{\Phi}_p\boldsymbol{\epsilon}_{M_p}, \quad n = 1, 2, \ldots, N,
\end{aligned}$$

where $\tilde{G}(\boldsymbol{\alpha}_{M_u}^n, \boldsymbol{\beta}_{M_v}^n, \boldsymbol{\delta}_{M_T}^n, \boldsymbol{\epsilon}_{M_p}^n) = (\boldsymbol{\Phi}_u^T, \boldsymbol{\Phi}_v^T, \boldsymbol{\Phi}_T^T, \boldsymbol{\Phi}_p^T)^T\tilde{F}(\boldsymbol{\Phi}_u\boldsymbol{\alpha}_{M_u}^n, \boldsymbol{\Phi}_v\boldsymbol{\beta}_{M_v}^n, \boldsymbol{\Phi}_T\boldsymbol{\delta}_{M_T}, \boldsymbol{\Phi}_p\boldsymbol{\epsilon}_{M_p})$ and $\tilde{F}(\cdot, \cdot, \cdot)$ is decided by the classical FD equations (2)-(5) writing as vector form, obtaining the reduced-order solution vectors $\mathbf{u}_m^{*n} = (u_1^{*n}, u_2^{*n}, \ldots, u_m^{*n})^T$, $\mathbf{v}_m^{*n} = (v_1^{*n}, v_2^{*n}, \ldots, v_m^{*n})^T$, $\mathbf{T}_m^{*n} = (T_1^{*n}, T_2^{*n}, \ldots, T_m^{*n})^T$, and $\mathbf{p}_m^{*n} = (p_1^{*n}, p_2^{*n}, \ldots, p_m^{*n})^T$, further, obtaining the component forms $u_{j+\frac{1}{2},k}^{*n} = u_i^{*n}$, $v_{j,k+\frac{1}{2}}^{*n} = v_i^{*n}$, $T_{j,k+\frac{1}{2}}^{*n} = T_i^{*n}$, and $p_{j,k+\frac{1}{2}}^{*n} = p_i^{*n}$ $(0 \leq j \leq J,\ 0 \leq k \leq K,\ i = k(J+1) + j + 1,\ 1 \leq i \leq m = KJ)$.

Step 5. Put

$$M = \max\bigl\{8\Delta t/\min\bigl\{\gamma^{-1}\Delta x^2, \gamma^{-1}\Delta y^2, \gamma_0\Delta x^2, \gamma_0\Delta y^2\bigr\}, \bigl(|u_{j+\frac{1}{2},k}^{*n}| + |v_{j,k+\frac{1}{2}}^{*n}|\bigr)\Delta t\gamma^{-1}/8\bigr\}.$$

If $2(1+M)^{n-L}[\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}] \leq \mu$ $(L+1 \leq n \leq N)$, then $\mathbf{u}_m^{*n} = (u_1^{*n}, u_2^{*n}, \ldots, u_m^{*n})^T$, $\mathbf{v}_m^{*n} = (v_1^{*n}, v_2^{*n}, \ldots, v_m^{*n})^T$, $\mathbf{T}_m^{*n} = (T_1^{*n}, T_2^{*n}, \ldots, T_m^{*n})^T$, and $\mathbf{p}_m^{*n} = (p_1^{*n}, p_2^{*n}, \ldots, p_m^{*n})^T$ $(n = 1, 2, \ldots, N)$ are just solutions satisfying accuracy needed. Else, namely, if $2(1+M)^{n-L}[\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}] > \mu$ $(L+1 \leq n \leq N)$, put $(u_1^l, u_2^l, \ldots, u_m^l) = (u_1^{*l}, u_2^{*l}, \ldots, u_m^{*l})$, $(v_1^l, v_2^l, \ldots, v_m^l) = (v_1^{*l}, v_2^{*l}, \ldots, v_m^{*l})$, $(T_1^l, T_2^l, \ldots, T_m^l) = (T_1^{*l}, T_2^{*l}, \ldots, T_m^{*l})$, and $(p_1^l, p_2^l, \ldots, p_m^l) = (p_1^{*l}, p_2^{*l}, \ldots, p_m^{*l})$ $(l = n-L, n-L-1, \ldots, n-1)$, return to Step 2.

Remark 3 If the classical FD equations (2)-(5) are used to solve NSIBEs, since it includes a large number of degrees of freedom and the truncation error is accumulated in the computational process, it may appear to have no convergence after some computing steps; while if the POD-based reduced-order FD extrapolating model (11)-(13) is used to find the numerical solutions for NSIBEs, since it includes fewer degrees of freedom, it can lessen the truncation error accumulation in the computational process and continuously simulate the development of the fluid flow.

5 A numerical example

In this section, we present a numerical example with a physical model of square cavity non-stationary flow to validate the feasibility and efficiency of the POD-based reduced-order FD extrapolating model with fully second-order accuracy.

Let the computational field $\Omega = (0,1) \times (0,1)$, $Re = 10^3$, $Pr = 0.1$, $\Delta x = \Delta y = 0.01$, $\Delta t = 0.01$, $u^0 = v^0 = \varphi_u = \varphi_v = 0$, and $\varphi_T = 0$ on $\bar{\Omega}$ except that $T^0 = \varphi_T = 2y(1.5 - y)$ (if $x = 1$) and $T^0 = \varphi_T = x$ (if $y = 1$) (see Figure 1). We find the numerical solutions $(u^n_{j+\frac{1}{2},k}, v^n_{j,k+\frac{1}{2}})$, $T^n_{j,k+\frac{1}{2}}$, and $p^n_{j,k}$ by the classical FD scheme (2)-(5) when $n = 300$ (for example, $t = 3$), which are depicted graphically on the left charts in Figures 2, 3, and 4, respectively.

When we take $L = 20$ and use Step 5 in Section 4.2 of the algorithm implementation for the POD-based reduced-order FD extrapolating model (11)-(13), we obtain from computing that $\sqrt{\lambda_{u7}} + \sqrt{\lambda_{v7}} + \sqrt{\lambda_{T7}} + \sqrt{\lambda_{p7}} \leq 4 \times 10^{-4}$. It is shown that it is only necessary to choose the first six POD basis functions. But when $n = 201$, the error $2(1+M)^{n-L}[\sqrt{\lambda_{u(M_u+1)}} + \sqrt{\lambda_{v(M_v+1)}} + \sqrt{\lambda_{T(M_T+1)}} + \sqrt{\lambda_{p(M_p+1)}}]$ exceeds 4×10^{-4}, and it is necessary to renew the POD basis once at $t = 2$ ($n = 200$). The reduced-order solutions $(u^{*n}_{j+\frac{1}{2},k}, v^{*n}_{j,k+\frac{1}{2}})$, $T^{*n}_{j,k+\frac{1}{2}}$, and $p^{*n}_{j,k+\frac{1}{2}}$ obtained by the POD reduced-order FD extrapolating model at $t = 3$ ($n = 300$) do not exceed 4×10^{-4} and are depicted graphically on the right charts in Figures 2, 3, and 4. Every two charts in Figures 2, 3, and 4 exhibit a

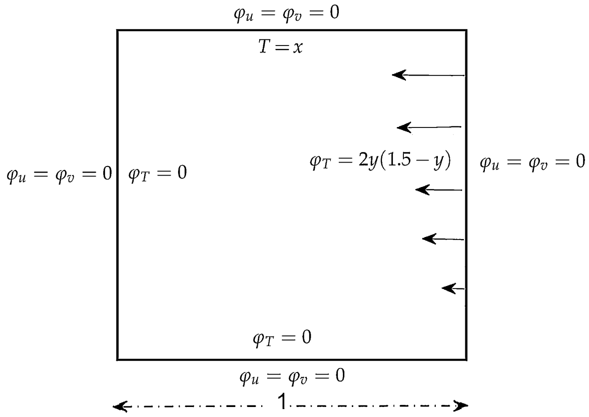

$\varphi_u = \varphi_v = 0$
$T = x$
$\varphi_u = \varphi_v = 0$
$\varphi_T = 0$
$\varphi_T = 2y(1.5 - y)$
$\varphi_u = \varphi_v = 0$
$\varphi_T = 0$
$\varphi_u = \varphi_v = 0$
1

Figure 1 Physics model of the cavity flows.

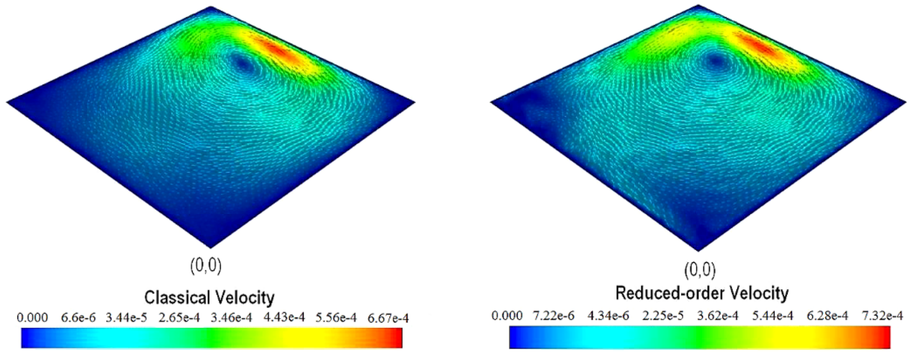


Figure 2 Classical FD solution (left chart) and reduced-order FD solution (right chart) of the velocity U at the time level *t* = 3 and when *Re* = 10^3 and *Pr* = 0.1.

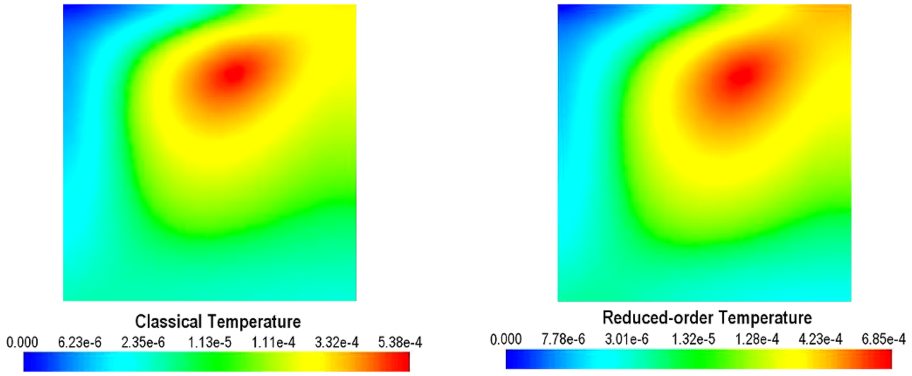


Figure 3 Classical FD solution (left chart) and reduced-order FD solution (right chart) of the temperature field *T* at the time level *t* = 3 and when *Re* = 10^3 and *Pr* = 0.1.

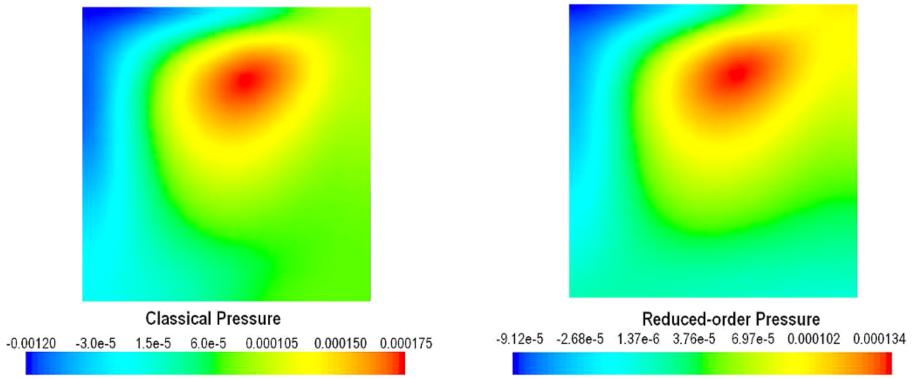


Figure 4 Classical FD solution (left chart) and reduced-order FD solution (right chart) of the pressure field *p* at the time level *t* = 3 and when *Re* = 10^3 and *Pr* = 0.1.

quasi-identical similarity, respectively. The errors of the POD-based reduced-order FD solutions on the starting time span are slightly larger than those of the classical FD solutions, since the POD-based reduced-order FD extrapolation model on each time level includes only 24 degrees of freedom and the classical FD scheme has 4×10^4 degrees of freedom. Namely, the degrees of freedom for the POD-based reduced-order FD extrapolation model are far fewer than those for classical FD scheme, so that the POD-based reduced-order FD extrapolation model could greatly lessen the truncation error accumulation in the computational process. It could also alleviate the calculating load, save time-consuming of calculations, and improve actual computational accuracy. Therefore, after some time span, the numerical errors of the POD-based reduced-order FD extrapolation model are fewer than those of the classical FD scheme (see Figures 5, 6, and 7). In fact, Figures 5, 6, and 7 show the cases of truncation error accumulation on $0 \leq t \leq 6$. It has been shown that the relation errors of classical FD solutions are far larger than those of reduced-order solutions obtained from the POD-based reduced-order FD extrapolating model. From the grow trends of relation errors of the classical FD solutions, in the classical FD scheme with fully second-order accuracy will appear no convergence after some computing steps, while the error accumulation of the POD-based reduced-order FD extrapolating model is very slow such that it can continuously simulate the development of

Figure 5 **Change charts of relation errors of reduced-order solution (red) and classical FD solution (green) of the velocity U with *Re* = 10^3 and *Pr* = 0.1 on 0 ≤ *t* ≤ 6.**

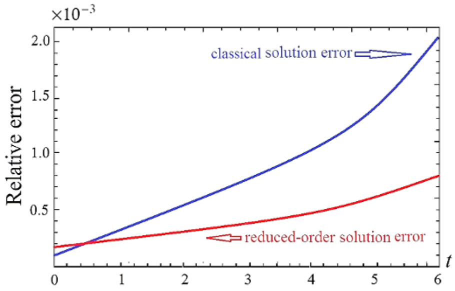


Figure 6 **Change charts of relation errors of reduced-order solution (red) and classical FD solution (green) of the pressure *p* with *Re* = 10^3 and *Pr* = 0.1 on 0 ≤ *t* ≤ 6.**

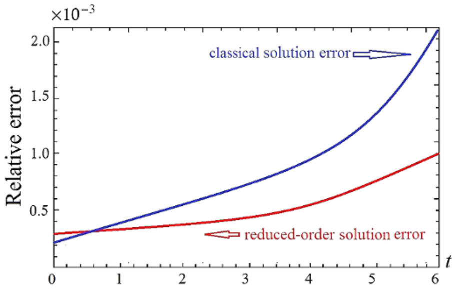


Figure 7 **Change charts of relation errors of reduced-order solution (red) and classical FD solution (green) of the temperature *T* with *Re* = 10^3 and *Pr* = 0.1 on 0 ≤ *t* ≤ 6.**

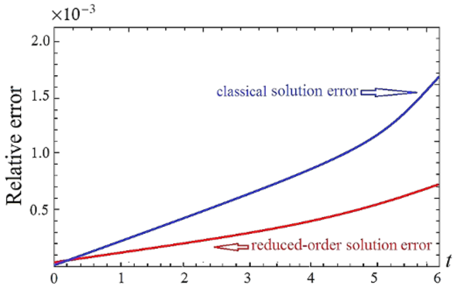
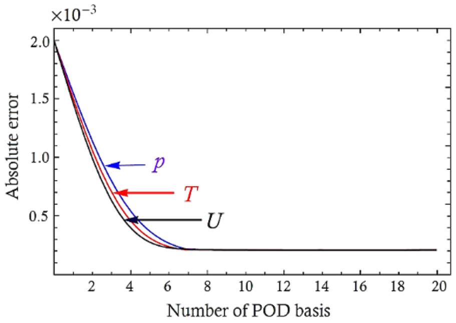


Figure 8 **Absolute errors for *Re* = 10^3 and *Pr* = 0.1 when the POD basis is different and at the time level *t* = 3.**



the fluid flow. It is also shown that the POD-based reduced-order FD extrapolating model with fully second-order accuracy is computationally very effective for finding the numerical solutions of NSIBEs.

Figure 8 shows the errors between the classical FD solution and the reduced-order solutions obtained by the POD-based reduced-order FD extrapolating model with different number of POD basis functions when $t = 3$, $Re = 1{,}000$, and $Pr = 0.1$. The numerical errors do not exceed 4×10^{-4} if M_u, M_v, M_T, and $M_p > 6$. Thus, the results for the numerical experiment are consistent with those obtained for the theoretical case.

In addition, if one uses the POD-based reduced-order FD scheme with first-order time accuracy in [17], it is necessary to take the time step as $k = 10^{-4}$ and implement 30,000 steps in order to obtain the same as accuracy here. Thus, its computing load is by a factor of 100 times larger than that in this article and its truncation error accumulation in the computational process is increased greatly; as well we have repeating computations of the classical FD scheme with first-order time accuracy on $[0, t_N]$. Therefore, the POD-based reduced-order FD extrapolating model with fully second-order accuracy here is an improvement and innovation for the existing POD-based reduced-order methods (for example, see [17–26]).

6 Conclusions and discussions

In this article, we have established the POD-based reduced-order FD extrapolating model with second-order time accuracy for NSIBEs and provided the error estimates between the classical FD solutions and the reduced-order solutions obtained from the POD-based reduced-order FD extrapolating model for guidance to choose the number of POD basis functions and renew the POD basis. We have also given the algorithm implementation for the POD-based reduced-order FD extrapolating model and a numerical experiment to show that the numerical errors are consistent with those obtained for the theoretical case.

In particular, in this article, we have completely improved the existing POD-based reduced-order methods (for example, see [17–26]), namely we do only use the first fewer given classical FD numerical solutions on the very short time span $[0, t_0]$ ($t_0 \ll t_N$) as snapshots to construct the POD basis and establish the POD-based reduced-order FD extrapolating model with fully second-order accuracy for finding the numerical solutions on total time span $[0, t_N]$. Thus, we adopt the POD method, being sufficiently advantageous, namely we utilize the given data (on the very short time span $[0, t_0]$) to predict future physic phenomena (on time span $[t_0, t_N]$). Its idea has very important guiding role if it is extended to weather forecast. The POD-based reduced-order FD extrapolating model with fully second-order accuracy and error estimates here are different from those of the existing POD-based reduced-order methods (for example, see [17–26]). Especially, if one uses the classical FD schemes (2)-(5) including a large number of degrees of freedom to solve NSIBEs, there may appear no convergence after some computing steps due to the truncation error accumulation in the computational process; while if one uses the POD-based reduced-order FD extrapolating model (11)-(13), due to its fewer degrees of freedom, it can reduce the truncation error accumulation in the computational process and continuously simulate the development of the fluid flow.

Future work in this area will aim to extend the POD reduced-order FD extrapolating model with fully second-order accuracy, implementing it for a realistic atmosphere quality forecast system and more complicated PDEs.

Appendix

Proof of Theorem 1 First, by using a Taylor expansion to expand two terms of (1) at the point (x_j, y_k, t_{n+1}), we have

$$\begin{aligned} u^{n+1}_{j+\frac{1}{2},k} - u^{n+1}_{j-\frac{1}{2},k} &= u^{n+1}_{j+\frac{1}{2},k} - u^{n+1}_{j,k} + u^{n+1}_{j,k} - u^{n+1}_{j-\frac{1}{2},k} \\ &= \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)^{n+1}_{j,k} + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 u}{\partial x^2}\right)^{n+1}_{j,k} + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 u}{\partial x^3}\right)^{n+1}_{j,k} + \cdots \end{aligned}$$

$$\begin{aligned}
&\quad + \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_{j,k}^{n+1} - \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,k}^{n+1} + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k}^{n+1} - \cdots \\
&= \Delta x\left(\frac{\partial u}{\partial x}\right)_{j,k}^{n+1} + \frac{(\Delta x)^3}{24}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k}^{n+1} + \cdots,
\end{aligned} \tag{26}$$

$$\begin{aligned}
v_{j,k+\frac{1}{2}}^{n+1} - v_{j,k-\frac{1}{2}}^{n+1} &= v_{j,k+\frac{1}{2}}^{n+1} - v_{j,k}^{n+1} + v_{j,k}^{n+1} - v_{j,k-\frac{1}{2}}^{n+1} \\
&= \frac{\Delta y}{2}\left(\frac{\partial v}{\partial y}\right)_{j,k}^{n+1} + \frac{1}{2!}\left(\frac{\Delta y}{2}\right)^2\left(\frac{\partial^2 v}{\partial y^2}\right)_{j,k}^{n+1} + \frac{1}{3!}\left(\frac{\Delta y}{2}\right)^3\left(\frac{\partial^3 v}{\partial y^3}\right)_{j,k}^{n+1} + \cdots \\
&\quad + \frac{\Delta y}{2}\left(\frac{\partial v}{\partial y}\right)_{j,k}^{n+1} - \frac{1}{2!}\left(\frac{\Delta y}{2}\right)^2\left(\frac{\partial^2 v}{\partial y^2}\right)_{j,k}^{n+1} + \frac{1}{3!}\left(\frac{\Delta y}{2}\right)^3\left(\frac{\partial^3 v}{\partial y^3}\right)_{j,k}^{n+1} - \cdots \\
&= \Delta y\left(\frac{\partial v}{\partial y}\right)_{j,k}^{n+1} + \frac{(\Delta y)^3}{24}\left(\frac{\partial^3 v}{\partial y^3}\right)_{j,k}^{n+1} + \cdots.
\end{aligned} \tag{27}$$

Inserting (26) and (27) into (1) yields

$$\left[\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y}\right]_{j,k}^{n+1} = -\frac{(\Delta x)^2}{24}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k}^{n+1} - \frac{(\Delta y)^3}{24}\left(\frac{\partial^2 v}{\partial y^3}\right)_{j,k}^{n+1} + \cdots. \tag{28}$$

Thus, the truncation error TE_1 for (1) approximating $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$ is

$$TE_1 = O\bigl((\Delta x)^2, (\Delta y)^2\bigr). \tag{29}$$

Second, by using a Taylor expansion to expand all terms of (2) at the point $(x_{j+\frac{1}{2}}, y_k, t_n)$, we have

$$\begin{aligned}
u_{j+\frac{1}{2},k}^{n+1} - u_{j+\frac{1}{2},k}^{n-1} &= u_{j+\frac{1}{2},k}^{n+1} - u_{j+\frac{1}{2},k}^{n} + u_{j+\frac{1}{2},k}^{n} - u_{j+\frac{1}{2},k}^{n-1} \\
&= \Delta t\left(\frac{\partial u}{\partial t}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta t)^2}{2!}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta t)^3}{3!}\left(\frac{\partial^3 u}{\partial t^3}\right)_{j+\frac{1}{2},k}^{n} + \cdots \\
&\quad + \Delta t\left(\frac{\partial u}{\partial t}\right)_{j+\frac{1}{2},k}^{n} - \frac{(\Delta t)^2}{2!}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta t)^3}{3!}\left(\frac{\partial^3 u}{\partial t^3}\right)_{j+\frac{1}{2},k}^{n} - \cdots \\
&= 2\Delta t\left(\frac{\partial u}{\partial t}\right)_{j+\frac{1}{2},k}^{n} + \frac{2(\Delta t)^3}{3!}\left(\frac{\partial^3 u}{\partial t^3}\right)_{j+\frac{1}{2},k}^{n} + \cdots,
\end{aligned} \tag{30}$$

$$\begin{aligned}
&u_{j-\frac{1}{2},k}^{n} - 2u_{j+\frac{1}{2},k}^{n} + u_{j+\frac{3}{2},k}^{n} \\
&= \bigl[u_{j+\frac{3}{2},k}^{n} - u_{j+\frac{1}{2},k}^{n}\bigr] + \bigl[u_{j-\frac{1}{2},k}^{n} - u_{j+\frac{1}{2},k}^{n}\bigr] \\
&= \Delta x\left(\frac{\partial u}{\partial x}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j+\frac{1}{2},k}^{n} \\
&\quad + \frac{(\Delta x)^4}{4!}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j+\frac{1}{2},k}^{n} + \cdots - \Delta x\left(\frac{\partial u}{\partial x}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j+\frac{1}{2},k}^{n} \\
&\quad - \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta x)^4}{4!}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j+\frac{1}{2},k}^{n} - \cdots \\
&= (\Delta x)^2\left(\frac{\partial^2 u}{\partial x^2}\right)_{j+\frac{1}{2},k}^{n} + \frac{(\Delta x)^4}{12}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j+\frac{1}{2},k}^{n} + \cdots,
\end{aligned} \tag{31}$$

$$\begin{aligned}
&u^n_{j+\frac{1}{2},k-1} - 2u^n_{j+\frac{1}{2},k} + u^n_{j+\frac{1}{2},k+1} \\
&\quad = \left[u^n_{j+\frac{1}{2},k+1} - u^n_{j+\frac{1}{2},k}\right] + \left[u^n_{j+\frac{1}{2},k} - u^n_{j+\frac{1}{2},k-1}\right] \\
&\quad = \Delta y\left(\frac{\partial u}{\partial y}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta y)^2}{2!}\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta y)^3}{3!}\left(\frac{\partial^3 u}{\partial y^3}\right)^n_{j+\frac{1}{2},k} \\
&\qquad + \frac{(\Delta y)^4}{4!}\left(\frac{\partial^4 u}{\partial y^4}\right)^n_{j+\frac{1}{2},k} + \cdots \\
&\qquad - \Delta y\left(\frac{\partial u}{\partial y}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta y)^2}{2!}\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{j+\frac{1}{2},k} - \frac{(\Delta y)^3}{3!}\left(\frac{\partial^3 u}{\partial y^3}\right)^n_{j+\frac{1}{2},k} \\
&\qquad + \frac{(\Delta y)^4}{4!}\left(\frac{\partial^4 u}{\partial y^4}\right)^n_{j+\frac{1}{2},k} - \cdots \\
&\quad = (\Delta y)^2\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta y)^4}{12}\left(\frac{\partial^4 u}{\partial y^4}\right)^n_{j+\frac{1}{2},k} + \cdots, \qquad (32)
\end{aligned}$$

$$\begin{aligned}
u^n_{j+1,k} - u^n_{j,k} &= \left[u^n_{j+1,k} - u^n_{j+\frac{1}{2},k}\right] + \left[u^n_{j+\frac{1}{2},k} - u^n_{j,k}\right] \\
&= \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)^n_{j+\frac{1}{2},k} + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 u}{\partial x^2}\right)^n_{j+\frac{1}{2},k} + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 u}{\partial x^3}\right)^n_{j+\frac{1}{2},k} + \cdots \\
&\quad + \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)^n_{j+\frac{1}{2},k} - \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 u}{\partial x^2}\right)^n_{j+\frac{1}{2},k} \\
&\quad + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 u}{\partial x^3}\right)^n_{j+\frac{1}{2},k} - \cdots \\
&= \Delta x\left(\frac{\partial u}{\partial x}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta x)^3}{24}\left(\frac{\partial^3 u}{\partial x^3}\right)^n_{j+\frac{1}{2},k} + \cdots, \qquad (33)
\end{aligned}$$

$$\begin{aligned}
&u^n_{j+\frac{1}{2},k+\frac{1}{2}} - u^n_{j+\frac{1}{2},k-\frac{1}{2}} \\
&\quad = \left[u^n_{j+\frac{1}{2},k+\frac{1}{2}} - u^n_{j+\frac{1}{2},k}\right] + \left[u^n_{j+\frac{1}{2},k} - u^n_{j+\frac{1}{2},k-\frac{1}{2}}\right] \\
&\quad = \frac{\Delta y}{2}\left(\frac{\partial u}{\partial y}\right)^n_{j+\frac{1}{2},k} + \frac{1}{2!}\left(\frac{\Delta y}{2}\right)^2\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{j+\frac{1}{2},k} + \frac{1}{3!}\left(\frac{\Delta y}{2}\right)^3\left(\frac{\partial^3 u}{\partial y^3}\right)^n_{j+\frac{1}{2},k} + \cdots \\
&\qquad + \frac{\Delta y}{2}\left(\frac{\partial u}{\partial y}\right)^n_{j+\frac{1}{2},k} - \frac{1}{2!}\left(\frac{\Delta y}{2}\right)^2\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{j+\frac{1}{2},k} + \frac{1}{3!}\left(\frac{\Delta y}{2}\right)^3\left(\frac{\partial^3 u}{\partial y^3}\right)^n_{j+\frac{1}{2},k} - \cdots \\
&\quad = \Delta y\left(\frac{\partial u}{\partial y}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta y)^3}{24}\left(\frac{\partial^3 u}{\partial y^3}\right)^n_{j+\frac{1}{2},k} + \cdots, \qquad (34)
\end{aligned}$$

$$\begin{aligned}
p^n_{j+1,k} - p^n_{j,k} &= \left[p^n_{j+1,k} - p^n_{j+\frac{1}{2},k}\right] + \left[p^n_{j+\frac{1}{2},k} - p^n_{j,k}\right] \\
&= \frac{\Delta x}{2}\left(\frac{\partial p}{\partial x}\right)^n_{j+\frac{1}{2},k} + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 p}{\partial x^2}\right)^n_{j+\frac{1}{2},k} + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 p}{\partial x^3}\right)^n_{j+\frac{1}{2},k} + \cdots \\
&\quad + \frac{\Delta x}{2}\left(\frac{\partial p}{\partial x}\right)^n_{j+\frac{1}{2},k} - \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 p}{\partial x^2}\right)^n_{j+\frac{1}{2},k} + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 p}{\partial x^3}\right)^n_{j+\frac{1}{2},k} - \cdots \\
&= \Delta x\left(\frac{\partial p}{\partial x}\right)^n_{j+\frac{1}{2},k} + \frac{(\Delta x)^3}{24}\left(\frac{\partial^3 p}{\partial x^3}\right)^n_{j+\frac{1}{2},k} + \cdots. \qquad (35)
\end{aligned}$$

Inserting (30)-(35) into (2) yields

$$\begin{aligned}
&\left[\frac{\partial u}{\partial t}+\frac{u\partial u}{\partial x}+\frac{v\partial u}{\partial y}+\frac{\partial p}{\partial x}-\gamma\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)\right]_{j+\frac{1}{2},k}^{n}\\
&\quad=-\frac{2(\Delta t)^2}{3!}\left(\frac{\partial^3 u}{\partial t^3}\right)_{j+\frac{1}{2},k}^{n}+\frac{\gamma(\Delta x)^2}{12}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j+\frac{1}{2},k}^{n}+\frac{\gamma(\Delta y)^2}{12}\left(\frac{\partial^4 u}{\partial y^4}\right)_{j+\frac{1}{2},k}^{n}\\
&\qquad-\frac{(\Delta x)^2}{24}\left(\frac{u\partial^3 u}{\partial x^3}\right)_{j+\frac{1}{2},k}^{n}-\frac{(\Delta y)^2}{24}\left(\frac{v\partial^3 u}{\partial y^3}\right)_{j+\frac{1}{2},k}^{n}-\frac{(\Delta x)^2}{24}\left(\frac{\partial^3 p}{\partial x^3}\right)_{j+\frac{1}{2},k}^{n}+\cdots. \qquad (36)
\end{aligned}$$

Therefore, the truncation error TE_2 for (2) approximating $\frac{\partial u}{\partial t}+\frac{u\partial u}{\partial x}+\frac{v\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\gamma(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})$ is

$$TE_2=O\big((\Delta t)^2,(\Delta x)^2,(\Delta y)^2\big). \qquad (37)$$

Next, using the same approach as in (37), the truncation errors TE_3 and TE_4 for (3) and (4), respectively, approximating the third and fourth equations for Problem I are given by

$$TE_3=O\big((\Delta t)^2,(\Delta x)^2,(\Delta y)^2\big), \qquad (38)$$

$$TE_4=O\big((\Delta t)^2,(\Delta x)^2,(\Delta y)^2\big). \qquad (39)$$

Since (5) is developed from (1), (2), (3), and (4), the numerical solutions for Problem I obtained from (2), (3), (4), and (5) have the following errors:

$$\begin{aligned}
&\big\|\big(u(x_{j+\frac{1}{2}},y_k,t_n),v(x_j,y_{k+\frac{1}{2}},t_n),T(x_j,y_{k+\frac{1}{2}},t_n),p(x_j,y_k,t_n)\big)-\big(u_{j+\frac{1}{2},k}^{n},v_{j,k+\frac{1}{2}}^{n},T_{j,k+\frac{1}{2}}^{n},p_{j,k}^{n}\big)\big\|\\
&\quad=O\big(\Delta t^2,\Delta x^2,\Delta y^2\big),\quad n=1,2,\ldots,N, \qquad (40)
\end{aligned}$$

which completes the proof of Theorem 1. □

Competing interests

The author declares that they have no competing interests.

Acknowledgements
This research was supported by National Science Foundation of China grant 11271127.

Received: 13 September 2014 Accepted: 16 October 2014 Published: 24 Oct 2014

References

1. Wu, JH: The 2D Incompressible Boussinesq Equations. Peking University Summer School Lecture Notes, Beijing (2012)
2. Xin, XK, Liu, RX, Jiang, BC: Computational Fluid Dynamics. National Defense Science Technology Press, Changsha (1989) (in Chinese)
3. Loéve, MM: Probability Theory. Van Nostrand, Princeton (1955)
4. Lumley, JL: The structure of inhomogeneous turbulent flows. In: Yaglom, AM (ed.) Atmos. Turbul. Radio Wave Propag., pp. 166-178 (1967)
5. Sirovich, L: Turbulence and the dynamics of coherent structures: parts I-III. Q. Appl. Math. **45**(3), 561-590 (1987)
6. Holmes, P, Lumley, JL, Berkooz, G: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, Cambridge (1996)
7. Fukunaga, K: Introduction to Statistical Recognition. Academic Press, New York (1990)
8. Jolliffe, IT: Principal Component Analysis. Springer, Berlin (2002)
9. Aubry, N, Holmes, P, Lumley, JL, Stone, E: The dynamics of coherent structures in the wall region of a turbulent boundary layer. J. Fluid Mech. **192**, 115-173 (1988)

10. Berkooz, G, Holmes, P, Lumley, JL: The proper orthogonal decomposition in analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**, 539-575 (1993)
11. Cazemier, W, Verstappen, RWCP, Veldman, AEP: Proper orthogonal decomposition and low-dimensional models for driven cavity flows. Phys. Fluids **10**, 1685-1699 (1998)
12. Ly, HV, Tran, HT: Proper orthogonal decomposition for flow calculations and optimal control in a horizontal CVD reactor. Q. Appl. Math. **60**, 631-656 (2002)
13. Moin, P, Moser, RD: Characteristic-eddy decomposition of turbulence in channel. J. Fluid Mech. **200**, 417-509 (1989)
14. Rajaee, M, Karlsson, SKF, Sirovich, L: Low dimensional description of free shear flow coherent structures and their dynamical behavior. J. Fluid Mech. **258**, 1401-1402 (1994)
15. Roslin, RD, Gunzburger, MD, Nicolaides, RA, Erlebacher, G, Hussaini, MY: A self-contained automated methodology for optimal flow control validated for transition delay. AIAA J. **35**, 816-824 (1997)
16. Selten, F: Baroclinic empirical orthogonal functions as basis functions in an atmospheric model. J. Atmos. Sci. **54**, 2100-2114 (1997)
17. Sun, P, Luo, ZD, Zhuo, YJ: A reduced finite difference scheme based on proper orthogonal decomposition for the non-stationary conduction-convection problems. Math. Numer. Sin. **31**(3), 323-334 (2009)
18. Kunisch, K, Volkwein, S: Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J. Numer. Anal. **40**, 492-515 (2002)
19. Luo, ZD, Ou, QL, Xie, ZH: A reduced finite difference scheme and error estimates based on POD method for the non-stationary Stokes equation. Appl. Math. Mech. **32**(7), 847-858 (2011)
20. Fang, F, Pain, CC, Navon, IM, Piggott, MD, Gorman, GJ, Farrell, PE, Allison, PA, Goddard, AJH: A POD reduced-order 4D-Var adaptive mesh ocean modelling approach. Int. J. Numer. Methods Fluids **60**(7), 709-732 (2009)
21. Luo, ZD, Wang, RW, Zhu, J: Finite difference scheme based on proper orthogonal decomposition for the non-stationary Navier-Stokes equations. Sci. China Ser. A **50**(8), 1186-1196 (2007)
22. Luo, ZD, Yang, XZ, Zhou, YJ: A reduced finite difference scheme based on singular value decomposition and proper orthogonal decomposition for Burgers equation. J. Comput. Appl. Math. **229**(1), 97-107 (2009)
23. Sun, P, Luo, ZD, Zhuo, YJ: Some reduced finite difference schemes based on a proper orthogonal decomposition technique for parabolic equations. Appl. Numer. Math. **60**, 154-164 (2010)
24. Luo, ZD, Du, J, Xie, ZH, Guo, Y: A reduced stabilized mixed finite element formulation based on proper orthogonal decomposition for the no-stationary Navier-Stokes equations. Int. J. Numer. Methods Eng. **88**, 31-46 (2011)
25. Fang, F, Pain, CC, Navon, IM, Piggott, MD, Gorman, GJ, Allison, P, Goddard, AJH: A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. Ocean Model. **28**, 127-136 (2009)
26. Luo, ZD, Chen, J, Sun, P, Yang, XZ: Finite element formulation based on proper orthogonal decomposition for parabolic equations. Sci. China Ser. A **52**(3), 587-596 (2009)
27. Sun, P, Li, H, Teng, F, Luo, ZD: A reduced-order extrapolating algorithm of fully second-order finite difference scheme for non-stationary Burgers equation. Sci. Sin., Math. **42**(12), 1171-1183 (2012)
28. Luo, ZD, Li, H, Sun, P, Gao, JQ: A reduced-order finite difference extrapolation algorithm based on POD technique for the non-stationary Navier-Stokes equations. Appl. Math. Model. **37**(7), 5464-5473 (2013)

10.1186/1687-1847-2014-272
Cite this article as: Luo: A POD-based reduced-order finite difference extrapolating model for the non-stationary incompressible Boussinesq equations. *Advances in Difference Equations* 2014, 2014:272