



Effective velocity-correction projection methods for unsteady incompressible natural convection equations

Jiangong Pan, Haiyan Su^{*}, Xinlong Feng

College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, PR China



ARTICLE INFO

Keywords:

Natural convection equations
Projection methods
Velocity-correction
Rotational form
Stability

ABSTRACT

We present and analyze two categories of efficient decoupled methods for the unsteady incompressible natural convection equations in this article. The proposed methods include first order, second order standard form and rotational form, and are extended to any order schemes. Some consistency items are added to help complete the stability analysis, and they disappear in non-discrete space. Finally, the correctness of the theoretical analysis and the effectiveness of the given methods are verified by several numerical experiments.

1. Introduction

In atmospheric dynamics, natural convection problem is extremely important. The control equations of natural convection problem are consisted of continuity, energy, and momentum equation. As everyone knows, fluid is viscous, so they will produce heat when they are moving. On the one hand, we need to ensure that there's heat transferred, which can ensure nondivergence in a closed domain. On the other hand, we hope results are not be affected by boundary conditions, so we choose a sufficiently large domain. In the field of heat transmission science, natural convection problem has always been a hot issue. The widespread use of it in production and life, such as solar energy device, nuclear reaction systems, energy storage and more.

To sum up, it is worthy of study and discussion. Then, we consider the unsteady incompressible natural convection equations:

$$\begin{cases} \partial_t u - Pr\Delta u + \nabla p + (u \cdot \nabla u) = PrRaT, & \text{in } \Omega \times (0, T^*], \\ \nabla \cdot u = 0, & \text{in } \Omega \times (0, T^*], \\ \partial_t T - \kappa \Delta T + u \cdot \nabla T = \gamma, & \text{in } \Omega \times (0, T^*], \\ u(x, 0) = u^0(x), \quad T(x, 0) = T^0(x), & \text{on } \partial\Omega \times \{0\}, \\ u = 0, \quad T = T_0(x, t), & \text{on } \partial\Omega \times (0, T^*] \end{cases}$$

where Ω is a bounded domain in \mathbb{R}^2 which is assumed to have a Lipschitz continuous boundary $\partial\Omega$, $u = (u_1(x), u_2(x))$ represents the velocity vector, $p = p(x)$ is the pressure, $T = T(x)$ is the temperature, $u^0(x)$ and $T^0(x)$ represent the initial velocity and the initial temperature, $T_0(x, t)$ is the boundary condition for the temperature on $(0, T^*]$. γ is external drive temperature, Pr is the Prandtl number and Ra is the Rayleigh number, κ

is the thermal conductivity parameter and $j = (0, 1)^T$ is the two-dimensional vector, $T^* > 0$ represents a finite time.

In the late 1960s, Chorin [1] and Temam [2] presented the original version of projection methods which were widely used to approximate the incompressible time-dependent problem. Its main idea is to overcome the difficulties caused by incompressibility constraint. We refer to some reviews about this topic and they divide projection methods into three categories: pressure-correction ([3–9]), velocity-correction ([10–12]), and consistent splitting scheme ([13,14]) (which is equivalent to the gauge method only in the space continuous case [15,16]).

Natural convection problem is a coupled nonlinear system, which is very difficult to solve. Therefore, the focus of this paper is velocity-correction scheme and at the same time is computationally efficient by decoupling the linear solve of the velocity and the pressure. In [11,12], velocity-correction scheme is presented at first time, and its rigorously analyze by Guermond and Shen in [10]. With this type of scheme, the viscous term is first treated explicitly, and a correction to the velocity is made subsequently. Standard and rotational forms of the velocity-correction schemes have been identified with differing accuracy in the velocity and pressure approximation. However, in the other methods, the pressure term is made explicit at first and corrected at last. The scheme of Karniadakis et al. [17] is equivalent to the velocity-correction scheme in rotational form [10].

The paper is organized as follows. In Section 2, we introduce notation for the space approximation, and we give the pressure-correction methods which are proposed in [18]. In Section 3, we propose the first-order version and the second-order version of velocity-correction

* Corresponding author.

E-mail addresses: mathpjg@sina.com (J. Pan), shymath@163.com (H. Su), fxlmath@xju.edu.cn (X. Feng).

algorithm in standard form for natural convection equations. In [Section 4](#), we also give two schemes of the velocity-correction algorithm in rotation form and give its stability proof. In [Section 5](#), numerical tests confirm the good property of the method and compare the velocity-correction method and pressure-correction method. Some concluding remarks follow.

2. Preliminaries

In this section we denote the usual Sobolev space by $\mathcal{W}^{m,p}(\Omega)$ and $\mathcal{W}_0^{m,p}(\Omega)$ with the norm $\|\cdot\|_{m,p}$. Particularly, $H^m(\Omega)$ ($m = 0, \pm 1, \dots$) with norm $\|\cdot\|_s$ and semi norm $|\cdot|_s$ is denoted by Hilbert spaces $\mathcal{W}^{m,2}(\Omega)$. $\|\cdot\|_0$ and (\cdot, \cdot) denote the norm and inner product of $L^2(\Omega) = H^0(\Omega)$ respectively. Then, we introduce the following necessary function spaces:

$$\begin{aligned} L^2_{\int=0}(\Omega) &= \left\{ q \in L^2(\Omega), \int_{\Omega} q = 0 \right\}, \\ H^1_{\int=0}(\Omega) &= \left\{ q \in H^1(\Omega), \int_{\Omega} q = 0 \right\}, \\ H &= \{v \in L^2(\Omega)^2, \nabla \cdot v = 0, v \cdot n|_{\partial\Omega} = 0\}, \\ W &= \{s \in H^1(\Omega), s|_{\partial\Omega} = 0\}. \end{aligned}$$

In the analysis of projection methods, Helmholtz decomposition of $L^2(\Omega)^2$ is important [10]:

$$L^2(\Omega)^2 = H \oplus \nabla H^1_{\int=0}(\Omega) \quad (2.1)$$

Then, we define $\tau > 0$ be a real number as the time step. We set $t^k = k\tau$ for $0 \leq k \leq K = [T^*/\tau]$. We set $\{X_h\}_{h>0}$, $\{W_h\}_{h>0}$ and $\{M_h\}_{h>0}$ be conforming approximations of $H_0^1(\Omega)^2$, $H^1(\Omega)^2$ and $L_0^2(\Omega)$, respectively. Note that the finite element space $X_h \times M_h$ satisfies the discrete inf-sup condition [19]:

$$\exists c > 0, \inf_{q_h \in M_h} \sup_{\substack{v_h \in X_h \\ v_h \neq 0}} \frac{(\nabla \cdot v_h, q_h)}{\|v_h\|_0} \geq c \|q_h\|_0, \quad (2.2)$$

where c is a generic constant which is independent of the mesh-size h and the time step τ .

We propose some discrete differential operators to make the semi-discrete natural convection problem and continuous differential counterpart as in [17].

We define the discrete Laplace operator: $A_h : X_h \rightarrow X_h'$, by

$$(A_h u_h, v_h) = (\nabla u_h, \nabla v_h), \quad \forall (u_h, v_h) \in X_h \times X_h,$$

where X_h' is the dual space of X_h . The discrete divergence operator, $B_h : X_h \rightarrow M_h$, and the discrete gradient operator, $B_h^T : M_h \rightarrow X_h'$, by

$$(B_h v_h, p_h) = -(\nabla \cdot v_h, p_h) = (v_h, B_h^T p_h), \quad \forall (v_h, p_h) \in X_h \times M_h.$$

We define an extension of the L^2 -projection onto X_h , $\pi_h : H^{-1}(\Omega)^d \rightarrow X_h'$, such that

$$(\pi_h f, v_h) = (f, v_h), \quad \forall v_h \in X_h.$$

We build a discrete Helmholtz decomposition (2.1) through the technology in [17]. In particular, discrete vector field is decomposed into the sum of a discrete-divergence-free vector field plus the discrete-gradient of a scalar field. This can be realized by many methods. For example, we can set $\tilde{u} = u_h + B_h^T \phi_h$, with $u_h \in X_h$ and $B_h u_h = 0$. Another way, we also can set $\tilde{u} = u_h + \phi_h$, where M_h is constructed so that $M_h \subset H_{\int=0}^1(\Omega)$ and u_h is enforced to be orthogonal to ∇M_h . Then, we choose u_h in $X_h + \nabla M_h$ for easy implementation and optimum.

To analyze the discrete Helmholtz decomposition, we set a finite dimensional subspace $Y_h \subset L^2(\Omega)^2$ which satisfies $X_h \subset Y_h$. i_h is denoted by the continuous injection of X_h into Y_h and i_h^T is denoted by the L^2 -projection of Y_h into X_h . Moreover, we give an extension of B_h , and get an

operator $C_h : Y_h \rightarrow M_h$

$$C_h i_h = B_h, \quad i_h^T C_h^T = B_h^T.$$

Combining (2.2), we know B_h is surjective. However, C_h being an extension of B_h , we can obtain C_h is also surjective and C_h^T is injective. So $\|C_h^T q\|_0$ is a norm and, upon setting $H_h = \ker(C_h)$, the orthogonal decomposition of Y_h as follow:

$$Y_h = H_h \oplus C_h^T(M_h)$$

Then, we assume that A_h and C_h satisfy hypotheses as follow [14]:

$$\begin{aligned} \forall v_h \in X_h, \quad \forall v \in [H_0^1(\Omega) \cap H^2(\Omega)]^d, \quad (\|v_h - v\|_1 < ch\|v\|_2) \Rightarrow \|A_h v_h\|_0 < c\|v\|_2, \\ \forall q_h \in M_h, \quad \forall q \in H_{\int=0}^1(\Omega), \quad (\|q_h - q\|_0 < ch\|q\|_1) \Rightarrow \|C_h^T q_h\|_0 < c\|q\|_1. \end{aligned} \quad (2.3)$$

These hypotheses are usually satisfied with finite element algorithm of shape-regular meshes.

3. Velocity-correction projection methods: Standard form

In this section, we introduce a new class of methods that treat the viscous (velocity) term explicitly in the first substep and correct it afterwards. We provide BDF1/BDF2 standard velocity-correction projection methods (SVPM) as follows.

3.1. BDF1 standard velocity-correction projection methods

Algorithm 3.1.1. BDF1-SVPM.

Step I. For $k \geq 2$, we find $u^{k+1} \in H(\Omega)^2$, $p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{\tau} (u^{k+1} - \tilde{u}^k) - Pr\Delta \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = PrRajT^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and $\tilde{u}^1, \tilde{u}^2, T^2$ can be computed by pressure-correction projection methods.

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

$$\begin{cases} \frac{1}{\tau} (\tilde{u}^{k+1} - u^{k+1}) - Pr\Delta (\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1}|_{\partial\Omega} = 0. \end{cases}$$

Step III. We find $T^{k+1} \in W$ such that

$$\begin{cases} \frac{1}{2\tau} (3T^{k+1} - 4T^k + T^{k-1}) - \kappa\Delta T^k + (\tilde{u}^{k+1} \cdot \nabla) T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

In [Algorithm 3.1.1](#), by combining the discrete Helmholtz decomposition in [Section 2](#), in the first step we solve u^{k+1} and correct it to get \tilde{u}^{k+1} in the second step. Then, we use the discrete operator mentioned in the [Section 2](#) and obtain the following discretization algorithm.

Algorithm 3.1.2. Discretization BDF1-SVPM.

Step I. For $k \geq 2$, we find $u_h^{k+1} \in Y_h$, $p_h^{k+1} \in M_h$ such that

$$\begin{cases} \frac{1}{\tau} (u_h^{k+1} - i_h \tilde{u}_h^k) + Pri A_h \tilde{u}_h^k + C_h^T p_h^{k+1} + \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = PrRajT_h^k, \\ C_h u_h^{k+1} = 0, \\ u_h^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases} \quad (3.1)$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by pressure-

correction projection methods.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{\tau} \left(\tilde{u}_h^{k+1} - i_h^T \tilde{u}_h^{k+1} \right) + PrA_h(\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1} \Big|_{\partial\Omega} = 0. \end{cases} \quad (3.2)$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{2\tau} (3T_h^{k+1} - 4T_h^k + T_h^{k-1}) - \kappa \Delta T_h^k + (\tilde{u}_h^{k+1} \cdot \nabla) T_h^{k+1} = \gamma_h^{k+1}, \\ T_h^{k+1} \Big|_{\partial\Omega} = T_0. \end{cases} \quad (3.3)$$

When $k \rightarrow \infty$, we suppose that this algorithm goes to a steady state. We can know $\tilde{u}_h = i_h^T u_h$ by (3.2). Thereby, we can obtain $B_h \tilde{u}_h \neq 0$ unless $B_h i_h^T u_h = C_h u_h$. However, only if i_h^T is identity operator and $B_h = C_h$, we can get $B_h i_h^T u_h = C_h u_h$. [Algorithm 3.1.2](#) is called consistent only if $X_h = Y_h$. For the sake of using more handy implementation methods for which $X_h \neq Y_h$, our main goals is finding a consistent way to discretize [Algorithm 3.1.1](#).

In order to solve the above difficulties, we consider a variant of [Algorithm 3.1.3](#):

Algorithm 3.1.3. The variation of discretization BDF1-SVPM.

Step I. For $k \geq 2$, we find $u_h^{k+1} \in Y_h$, $p_h^{k+1} \in M_h$ such that

$$\begin{cases} \frac{1}{\tau} \left(u_h^{k+1} - i_h \tilde{u}_h^k \right) + Pri_h A_h \tilde{u}_h^k + C_h^T p_h^{k+1} + i_h B_h^T p_h^k - C_h^T p_h^k \\ \quad + \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = PrRaj T_h^k, \\ C_h u_h^{k+1} = 0, \\ u_h^{k+1} \cdot n \Big|_{\partial\Omega} = 0, \end{cases} \quad (3.4)$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by pressure-correction projection methods.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{\tau} \left(\tilde{u}_h^{k+1} - i_h^T \tilde{u}_h^{k+1} \right) + PrA_h(\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1} \Big|_{\partial\Omega} = 0. \end{cases} \quad (3.5)$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{2\tau} (3T_h^{k+1} - 4T_h^k + T_h^{k-1}) - \kappa \Delta T_h^k + (\tilde{u}_h^{k+1} \cdot \nabla) T_h^{k+1} = \gamma_h^{k+1}, \\ T_h^{k+1} \Big|_{\partial\Omega} = T_0. \end{cases} \quad (3.6)$$

Obviously, in (3.4) the term $i_h B_h^T p_h^k - C_h^T p_h^k$ vanish when $X_h = Y_h$. This means [Algorithm 3.1.3](#) is the same as [Algorithm 3.1.2](#) as $X_h = Y_h$. We can rewrite (3.5) in an equivalent form: Applying i_h^T to (3.4) and adding the result to (3.5). Combining $i_h^T C_h^T = i_h^T i_h B_h^T$ and the fact $i_h^T i_h|_{X_h}$ is the identity

on X_h , we can get

$$\frac{1}{\tau} \left(\tilde{u}_h^{k+1} - \tilde{u}_h^k \right) + PrA_h \tilde{u}_h^{k+1} + B_h^T p_h^{k+1} + \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} = PrRaj T_h^k. \quad (3.7)$$

In order to know more details between algorithm (3.4–3.7–3.6)) and [Algorithm 3.1.3](#), let us apply i_h to (3.7) at time step t^k and subtract the result from (3.4), giving

$$\frac{1}{2\tau} \left(u_h^{k+1} - 2i_h \tilde{u}_h^k + i_h \tilde{u}_h^{k-1} \right) + C_h^T (p_h^{k+1} - p_h^k) = PrRaji_h (T_h^k - T_h^{k-1}) \quad (3.8)$$

Analogously, supposing that there is a steady state as $k \rightarrow \infty$, (3.8) can obtain $0 = C_h u_h = C_h i_h \tilde{u}_h = B_h \tilde{u}_h$. Hence, (3.4–3.7–3.6) is a consistent way of implementing [Algorithm 3.1.1](#).

Because of the constraint $(u^{k+1}, C_h^T r_h) = 0$ for all $r_h \in M_h$, (3.8) can be rewritten in an equivalent way as

$$(C_h^T (p_h^{k+1} - p_h^k), C_h^T r_h) = PrRaj (T_h^k - T_h^{k-1}, B_h^T r_h) + \frac{1}{2\tau} \left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}, B_h^T r_h \right). \quad (3.9)$$

In conclusion, implementing [Algorithm 3.1.1](#) consists of solving (3.9–3.7–3.6).

3.2. BDF2 standard velocity-correction projection methods

Algorithm 3.2.1. BDF2-SVPM.

Step I. For $k \geq 2$, we find $u^{k+1} \in H(\Omega)^2$, $p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{2\tau} (3u^{k+1} - 4\tilde{u}^k + \tilde{u}^{k-1}) - Pr\Delta \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = PrRaj T^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1} \cdot n \Big|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and $\tilde{u}^1, \tilde{u}^2, T^2$ can be computed by [Algorithm 3.1.1](#).

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

$$\begin{cases} \frac{1}{2\tau} (3\tilde{u}^{k+1} - 3u^{k+1}) - Pr\Delta(\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1}|_{\partial\Omega} = 0. \end{cases}$$

Step III. We find $T^{k+1} \in W$ such that

$$\begin{cases} \frac{1}{2\tau} (3T^{k+1} - 4T^k + T^{k-1}) - \kappa\Delta T^k + (\tilde{u}^{k+1} \cdot \nabla) T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

Next, we do the same as in [Section 3.1](#). When we add discrete operators, it is easy to see that the same inconsistencies occur. Therefore, we omit Algorithm 3.2.2 (Discretization BDF2-SVPM), and we directly give Algorithm 3.2.3 after adding consistency items.

Algorithm 3.2.3. The variation of discretization BDF2-SVPM.

Step I. For $k \geq 2$, we find $u_h^{k+1} \in Y_h$, $p_h^{k+1} \in M_h$ such that

$$\begin{cases} \frac{1}{2\tau} \left(3u_h^{k+1} - 4i_h \tilde{u}_h^k + i_h \tilde{u}_h^{k-1} \right) + Pri_h A_h \tilde{u}_h^k + C_h^T p_h^{k+1} + i_h B_h^T p_h^k - C_h^T p_h^k \\ \quad + \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = PrRajT_h^k, \\ C_h u_h^{k+1} = 0, \\ u_h^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases} \quad (3.10)$$

$$\begin{aligned} & 3 \left\| \delta \tilde{u}_h^{n+1} \right\|_0^2 + 3 \left\| 2\delta \tilde{u}_h^{n+1} - \delta \tilde{u}_h^n \right\|_0^2 + 8 \left\| \nabla \delta \tilde{u}_h^{n+1} \right\|_0^2 + 4 \left\| \nabla \delta^2 \tilde{u}_h^{n+1} \right\|_0^2 + 4\tau^2 \| (C_h^T - i_h B_h^T) \phi_h^{n+1} \|_0^2 \\ & \quad + 4\tau^2 \left\| D'_h \tilde{u}_h^{n+1} \right\|_0^2 + \| T_h^{n+1} \|_0^2 + \| 2T_h^{n+1} - T_h^n \|_0^2 \\ & \leq 3 \left\| \delta \tilde{u}_h^1 \right\|_0^2 + 3 \left\| 2\delta \tilde{u}_h^1 - \delta \tilde{u}_h^0 \right\|_0^2 + 8 \left\| \nabla \delta \tilde{u}_h^1 \right\|_0^2 + 4 \left\| \nabla \delta^2 \tilde{u}_h^1 \right\|_0^2 + 4\tau^2 \| (C_h^T - i_h B_h^T) \delta \phi_h^1 \|_0^2 \\ & \quad + 4\tau^2 \left\| D'_h \tilde{u}_h^1 \right\|_0^2 + \| T_h^1 \|_0^2 + \| 2T_h^1 - T_h^0 \|_0^2 + c \|\gamma_h\|_{C(0,T;H^{-1})}^2, \end{aligned}$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by Algorithm 3.1.3.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 3i_h^T u_h^{k+1} \right) + PrA_h (\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} \\ \quad - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1}|_{\partial\Omega} = 0. \end{cases} \quad (3.11)$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{2\tau} (3T_h^{k+1} - 4T_h^k + T_h^{k-1}) - \kappa\Delta T_h^k + (\tilde{u}_h^{k+1} \cdot \nabla) T_h^{k+1} = \gamma_h^{k+1}, \\ T_h^{k+1}|_{\partial\Omega} = T_0. \end{cases} \quad (3.12)$$

Then, we do same treatment and get the equivalence of (3.11):

$$\begin{aligned} & \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 4\tilde{u}_h^k + \tilde{u}_h^{k-1} \right) + PrA_h \tilde{u}_h^{k+1} + B_h^T p_h^{k+1} + \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} \\ & = PrRajT_h^k. \end{aligned} \quad (3.13)$$

Next, combining $(u^{k+1}, C_h^T r_h) = 0$ for all $r_h \in M_h$, let us apply i_h to (3.13) at time step t^k and subtract the result from (3.10), we have:

$$(C_h^T (p_h^{k+1} - p_h^k), C_h^T r_h) = PrRaj(T_h^k - T_h^{k-1}, B_h^T r_h) + \frac{1}{2\tau} \left(7\tilde{u}_h^k - 5\tilde{u}_h^{k-1} + \tilde{u}_h^{k-2}, B_h^T r_h \right). \quad (3.14)$$

Hence implementing Algorithm 3.2.1 contains three sub-steps (3.14–3.13–3.12).

Then, we focus the stability of Algorithm 3.2.3. We need to define some notation as follow:

$$\phi_h^{k+1} := \delta p_h^{k+1}, \quad D'_h \tilde{u}_h^{k+1} := PrA_h \delta \tilde{u}_h^{k+1}.$$

Theorem 3.1. The solution of the Algorithm 3.2.3 is bounded in the following sense:

where c is a constant.

Proof. We omitted the proofs because they are similar to the proof of the rotational version described in a later section.

Remark 3.1. The above algorithms can generalize to time-marching algorithms. For example, assuming \mathbf{v} to be a smooth function, q th-order backward difference formula (BDF q) is denoted by $\frac{1}{\tau} (\alpha_q \mathbf{v}^{k+1} - \sum_{j=0}^{q-1} \alpha_j \mathbf{v}^{k-j})$ that approximates $\partial_t \mathbf{v}(t^{k+1})$. For convenience, for any sequence $\phi_{\Delta t} := (\phi^0, \phi^1, \dots)$, we set

$$D\phi^{k+1} = \alpha_q \phi^{k+1} - \sum_{j=0}^{q-1} \alpha_j \phi^{k-j}.$$

Now, we give a general standard form of velocity-correction schemes:

Algorithm 3.2.4. BDF q -SVPM.

Step I. For $k \geq q$, we find $u^{k+1} \in H(\Omega)^2$, $p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{\tau} \left(\alpha_q u^{k+1} - \sum_{j=0}^{q-1} \alpha_j \tilde{u}^{k-j} \right) - Pr \Delta \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = Pr Raj T^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and $\tilde{u}^i (i = 1, \dots, k)$, T^k can be computed by lower order method.

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

4.1. BDF1 rotational velocity-correction projection methods

Algorithm 4.1.1. BDF1-RVPM.

Step I. For $k \geq 2$, we find $u^{k+1} \in H(\Omega)^2$, $p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{\tau} (u^{k+1} - \tilde{u}^k) + Pr \nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = Pr Raj T^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases}$$

$$\begin{cases} \frac{\alpha_q}{\tau} (\tilde{u}^{k+1} - u^{k+1}) - Pr \Delta (\tilde{u}^{k+1} - \tilde{u}^k) + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1}|_{\partial\Omega} = 0. \end{cases}$$

Step III. We find $T^{k+1} \in W$ such that

$$\begin{cases} \frac{1}{2\tau} (3T^{k+1} - 4T^k + T^{k-1}) - \kappa \Delta T^k + (\tilde{u}^{k+1} \cdot \nabla) T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and $\tilde{u}^1, \tilde{u}^2, T^2$ can be computed by pressure-correction projection methods.

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

$$\begin{cases} \frac{1}{\tau} (\tilde{u}^{k+1} - u^{k+1}) + Pr \nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1}|_{\partial\Omega} = 0. \end{cases}$$

Remark 3.2. Combining Algorithm 3.1.1 and Algorithm 3.2.1, we get $\frac{\partial p^{k+1}}{\partial n}|_{\partial\Omega} = (f(t^{k+1}) + \Delta \tilde{u}^0) \cdot n|_{\partial\Omega}$. It's not hard to see that this is an artificial Neumann boundary condition of pressure appeared in the new scheme. And they prevent from being fully second-order on the velocity in the H^1 -norm and on the pressure in the L^2 -norm. This difficult can be overcome by rotational velocity-correction projection methods in the next section.

4. Velocity-correction projection methods: rotation form

In this section we focus our attention on BDF1/BDF2 rotational velocity-correction projection methods (RVPM). We are led to replace $-\Delta \tilde{u}^k$ with $\nabla \times \nabla \times \tilde{u}^k$ because of $\Delta \tilde{u}^k = \nabla \nabla \cdot \tilde{u}^k - \nabla \times \nabla \times \tilde{u}^k$. The splitting error of this methods does not depend on the time stepping.

Step III. We find $T^{k+1} \in W$ such that

$$\begin{cases} \frac{1}{2\tau} (3T^{k+1} - 4T^k + T^{k-1}) - \kappa \Delta T^k + (\tilde{u}^{k+1} \cdot \nabla) T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

Since $\nabla \times \nabla \times \tilde{u}^k = \nabla \nabla \cdot \tilde{u}^k - \Delta \tilde{u}^k$, we can obtain a natural approximation of $\nabla \times \nabla \times \tilde{u}^k$ is $i_h A_h \tilde{u}_h^k - C_h^T B_h \tilde{u}_h^k$. And by using the previous discrete operator, we can get the following fully discretized scheme:

Algorithm 4.1.2. Discretization BDF1-RVPM.

Step I. For $k \geq 2$, find $u_h^{k+1} \in Y_h$, $p_h^{k+1} \in M_h$ such that

$$\begin{cases} \frac{1}{\tau} \left(u_h^{k+1} - i_h \tilde{u}_h^k \right) + Pr i_h A_h \tilde{u}_h^k - C_h^T B_h \tilde{u}_h^k + C_h^T p_h^{k+1} \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = Pr Raj T_h^k, \\ C_h u_h^{k+1} = 0, \\ u_h^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by pressure-correction projection methods.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{\tau} (\tilde{u}_h^{k+1} - i_h^T \tilde{u}_h^{k+1}) + PrA_h(\tilde{u}^{k+1} - \tilde{u}^k) + B_h^T B_h \tilde{u}_h^k + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1} \Big|_{\partial\Omega} = T_0. \end{cases} \quad (4.3)$$

$$\begin{cases} \frac{1}{\tau} (\tilde{u}_h^{k+1} - i_h^T \tilde{u}_h^{k+1}) + PrA_h(\tilde{u}^{k+1} - \tilde{u}^k) + B_h^T B_h \tilde{u}_h^k + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1} \Big|_{\partial\Omega} = 0. \end{cases}$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{\tau} (3T_h^{k+1} - 4T_h^k + T_h^{k-1}) - \kappa \Delta T_h^k + (\tilde{u}_h^{k+1} \cdot \nabla) T_h^{k+1} = \gamma_h^{k+1}, \\ T_h^{k+1} \Big|_{\partial\Omega} = T_0. \end{cases}$$

By proceeding as in [Section 3](#), Algorithm 4.1.2 is not consistent at steady state. Then, we now consider the following modified algorithm:

Algorithm 4.1.3. The variation of discretization BDF1-RVPM.

Step I. For $k \geq 2$, find $u_h^{k+1} \in Y_h, p_h^{k+1} \in M_h$ such that

$$\begin{cases} \frac{1}{\tau} (u_h^{k+1} - i_h \tilde{u}_h^k) + Pri_h A_h \tilde{u}_h^k - C_h^T B_h \tilde{u}_h^k + C_h^T p_h^{k+1} + i_h B_h^T p_h^k - C_h^T p_h^k \\ \quad + \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = PrRaj T_h^k, \\ C_h u_h^{k+1} = 0, \\ u_h^{k+1} \cdot n \Big|_{\partial\Omega} = 0, \end{cases} \quad (4.1)$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by pressure-correction projection methods.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{\tau} (\tilde{u}_h^{k+1} - i_h^T \tilde{u}_h^{k+1}) + PrA_h(\tilde{u}^{k+1} - \tilde{u}^k) + B_h^T B_h \tilde{u}_h^k + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} \\ \quad - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1} \Big|_{\partial\Omega} = 0. \end{cases} \quad (4.2)$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{\tau} (3\tilde{u}^{k+1} - 3u^{k+1}) + Pr\nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1} \Big|_{\partial\Omega} = 0. \end{cases}$$

Doing the same things as [Section 3](#), we can get the equivalent form which avoids computing the velocity $u_h^{k+1} \in Y_h$. By using i_h^T in (4.1) and add it to (4.2), then we can get

$$\frac{1}{\tau} (\tilde{u}_h^{k+1} - \tilde{u}_h^k) + PrA_h \tilde{u}_h^{k+1} + B_h^T p_h^{k+1} + \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} = PrRaj T_h^k. \quad (4.4)$$

Next, at time step t^k , we multiply $-i_h$ to (4.4) and add the result to (4.1). We have

$$\frac{1}{\tau} \left(u_h^{k+1} - 2i_h \tilde{u}_h^k + i_h \tilde{u}_h^{k-1} \right) + C_h^T \left(p_h^{k+1} - p_h^k - B_h \tilde{u}_h^k \right) = PrRaji_h (T_h^k - T_h^{k-1})$$

Finally, because of the constraint $(u^{k+1}, C_h^T r_h) = 0$ for all $r_h \in M_h$, this problem can be changed as follow:

$$\left(C_h^T \left(p_h^{k+1} - p_h^k - B_h \tilde{u}_h^k \right), C_h^T r_h \right) = PrRaj (T_h^k - T_h^{k-1}, B_h^T r_h) + \frac{1}{2\tau} \left(2\tilde{u}_h^k - 1\tilde{u}_h^{k-1}, B_h^T r_h \right). \quad (4.5)$$

Hence implementing Algorithm 4.1.1 contains three sub-steps (4.5–4.4–4.3).

4.2. BDF2 rotational velocity-correction projection methods

Algorithm 4.2.1. BDF2-RVPM.

Step I. For $k \geq 2$, we find $u^{k+1} \in H(\Omega)^2, p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{2\tau} (3u^{k+1} - 4\tilde{u}^k + \tilde{u}^{k-1}) + Pr\nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = PrRaj T^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1} \cdot n \Big|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and $\tilde{u}^1, \tilde{u}^2, T^2$ can be computed by Algorithm 4.1.1.

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

Step III. We find $T^{k+1} \in W$ such that

$$\begin{cases} \frac{1}{2\tau} (3T^{k+1} - 4T^k + T^{k-1}) - \kappa\Delta T^k + (\tilde{u}^{k+1} \cdot \nabla) T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

Next, we do the same as in [Section 4.1](#). When we add discrete operators, it is easy to see that the same inconsistencies occur. Therefore, we omit Algorithm 4.2.2 (Discretization BDF2-RVPM), and we directly give Algorithm 4.2.3 after adding consistency items.

Algorithm 4.2.3. The variation of discretization BDF2-RVPM.

Step I. For $k \geq 2$, find $(\tilde{u}_h^{k+1}, p_h^{k+1}) \in Y_h \times M_h$ such that

$$\begin{cases} \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 4i_h \tilde{u}_h^k + i_h \tilde{u}_h^{k-1} \right) + PrA_h \tilde{u}_h^k - C_h^T B_h \tilde{u}_h^k + C_h^T p_h^{k+1} + i_h B_h^T p_h^k - C_h^T p_h^k \\ \quad + \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k = PrRajT_h^k, \\ C_h \tilde{u}_h^{k+1} = 0, \\ \tilde{u}_h^{k+1} \cdot n|_{\partial\Omega} = 0, \end{cases} \quad (4.6)$$

where $\tilde{u}_h^0 = \pi_h u^0(x)$ and $\tilde{u}_h^1, \tilde{u}_h^2, T_h^2$ can be computed by Algorithm 4.1.3.

Step II. We find $\tilde{u}_h^{k+1} \in X_h$ such that

$$\begin{cases} \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 3i_h^T \tilde{u}_h^{k+1} \right) + PrA_h (\tilde{u}^{k+1} - \tilde{u}^k) + B_h^T B_h \tilde{u}_h^k + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} \\ \quad - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}_h^{k+1}|_{\partial\Omega} = 0. \end{cases} \quad (4.7)$$

Step III. We find $T_h^{k+1} \in W_h$ such that

$$\begin{cases} \frac{1}{2\tau} (3T_h^{k+1} - 4T_h^k + T_h^{k-1}) - \kappa\Delta T_h^k + (\tilde{u}_h^{k+1} \cdot \nabla) T_h^{k+1} = \gamma_h^{k+1}. \\ T_h^{k+1}|_{\partial\Omega} = T_0. \end{cases} \quad (4.8)$$

Then, we can get the equivalent form which avoids computing the velocity $u_h^{k+1} \in Y_h$. By using i_h^T in (4.6) and add it to (4.7), then we can get

$$\begin{aligned} & \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 4\tilde{u}_h^k + \tilde{u}_h^{k-1} \right) + PrA_h \tilde{u}_h^{k+1} + B_h^T p_h^{k+1} + \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} \\ &= PrRajT_h^k. \end{aligned} \quad (4.9)$$

Next, at time step t^k , we multiply $-i_h$ to (4.9) and add the result to (4.6) and use $(u^{k+1}, C_h^T r_h) = 0$ for all $r_h \in M_h$, this problem can be changed into the following form:

In conclusion, implementing Algorithm 4.2.1 consists of solving

$$(C_h^T \left(p_h^{k+1} - p_h^k - B_h \tilde{u}_h^k \right), C_h^T r_h) = PrRaj(T_h^k - T_h^{k-1}, B_h^T r_h) + \frac{1}{2\tau} \left(7\tilde{u}_h^k - 5\tilde{u}_h^{k-1} + \tilde{u}_h^{k-2}, B_h^T r_h \right). \quad (4.10)$$

(4.10–4.9–4.8).

At last, let us shift focus on stability of Algorithm 4.2.3. In first, we define

$$\phi_h^{k+1} := \delta p_h^{k+1} - B_h \tilde{u}_h^k, \quad D_h' \tilde{u}_h^{k+1} := PrA_h \delta \tilde{u}_h^{k+1} + B_h^T B_h \tilde{u}_h^k. \quad (4.11)$$

Theorem 4.1. The solution of Algorithm 4.2.3 is bounded in the following sense:

$$\begin{aligned} & 3 \left\| \delta \tilde{u}_h^{n+1} \right\|_0^2 + 6 \left\| B_h \tilde{u}_h^{n+1} \right\|_0^2 + 3 \left\| 2\delta \tilde{u}_h^{n+1} - \delta \tilde{u}_h^n \right\|_0^2 + 8 \left\| \nabla \delta \tilde{u}_h^{n+1} \right\|_0^2 + 4 \left\| \nabla \delta^2 \tilde{u}_h^{n+1} \right\|_0^2 \\ & + 4\tau^2 \left\| (C_h^T - i_h B_h^T) \phi_h^{n+1} \right\|_0^2 + 4\tau^2 \left\| D_h' \tilde{u}_h^{n+1} \right\|_0^2 + \left\| T_h^{n+1} \right\|_0^2 + \left\| 2T_h^{n+1} - T_h^n \right\|_0^2 \\ & \leq 3 \left\| \delta \tilde{u}_h^1 \right\|_0^2 + 6 \left\| B_h \tilde{u}_h^1 \right\|_0^2 + 3 \left\| 2\delta \tilde{u}_h^1 - \delta \tilde{u}_h^0 \right\|_0^2 + 8 \left\| \nabla \delta \tilde{u}_h^1 \right\|_0^2 + 4 \left\| \nabla \delta^2 \tilde{u}_h^1 \right\|_0^2 \\ & + 4\tau^2 \left\| (C_h^T - i_h B_h^T) \delta \phi_h^1 \right\|_0^2 + 4\tau^2 \left\| D_h' \tilde{u}_h^1 \right\|_0^2 + \left\| T_h^1 \right\|_0^2 + \left\| 2T_h^1 - T_h^0 \right\|_0^2 + c \left\| \gamma_h \right\|_{C(0,T;H^{-1})}^2. \end{aligned}$$

where c is a constant. *Proof.* Applying i_h^T to (4.6) and subtracting the result from (4.9), we can get

$$\begin{aligned} & \frac{1}{2\tau} \left(3\tilde{u}_h^{k+1} - 3i_h^T \tilde{u}_h^{k+1} \right) + PrA_h (\tilde{u}^{k+1} - \tilde{u}^k) + B_h^T B_h \tilde{u}_h^k + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} \\ & \quad - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0. \end{aligned} \quad (4.12)$$

Then using the time increment operator δ to (4.6, 4.9, 4.12), we obtain

$$\begin{cases} 3\delta \tilde{u}_h^{k+1} + 2\tau C_h^T \phi_h^{k+1} = i_h \left(4\delta \tilde{u}_h^k - \delta \tilde{u}_h^{k-1} \right) - 2\tau i_h D_h' \tilde{u}_h^k + 2\tau (C_h^T - i_h B_h^T) \phi_h^k \\ \quad - 2\tau \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) i_h \tilde{u}_h^k + 2\tau \left((2\tilde{u}_h^{k-2} - \tilde{u}_h^{k-3}) \cdot \nabla \right) i_h \tilde{u}_h^{k-1} \\ \quad + 2\tau PrRaj i_h \delta T_h^k, \\ C_h \tilde{u}_h^{k+1} = 0. \end{cases} \quad (4.13)$$

$$\begin{aligned} & 3\delta \tilde{u}_h^{k+1} - 4\delta \tilde{u}_h^k + \delta \tilde{u}_h^{k-1} + 2\tau D_h' \tilde{u}_h^{k+1} + 2\tau \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} \\ & - 2\tau \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) \tilde{u}_h^k = -2\tau B_h^T \phi_h^{k+1} + 2\tau PrRaj \delta T_h^k. \end{aligned} \quad (4.14)$$

$$\begin{aligned} & 3\delta \tilde{u}_h^{k+1} + 2\tau D_h' \delta \tilde{u}_h^{k+1} + 2\tau \left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1} - 2\tau \left((2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}) \cdot \nabla \right) \tilde{u}_h^{k-1} \\ & = 3i_h^T \delta \tilde{u}_h^{k+1}. \end{aligned} \quad (4.15)$$

Let us multiply (4.8) by $4\tau T_h^{k+1}$ and integrate over Ω . Combining

$$2(3u_h^{n+1} - 4u_h^n + u_h^{n-1}, u_h^{n+1}) = \|u_h^{n+1}\|_0^2 + \|2u_h^{n+1} - u_h^n\|_0^2 + \|\delta \delta u_h^{n+1}\|_0^2 - \|u_h^n\|_0^2 - \|2u_h^n - u_h^{n-1}\|_0^2.$$

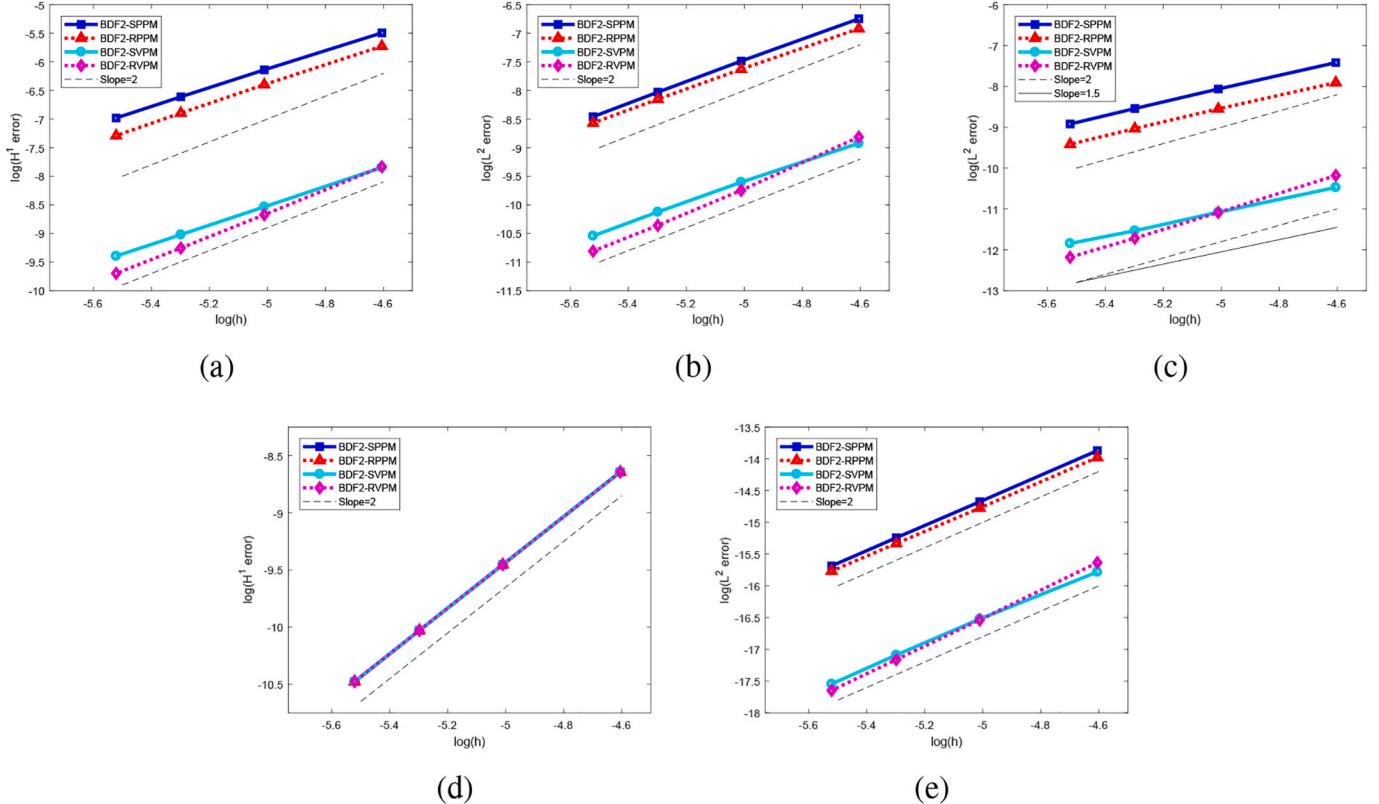


Fig. 1. Convergence analysis. (a) H^1 error for the velocity, (b) L^2 error for the velocity, (c) L^2 error for the pressure, (d) H^1 error for the Temperature, and (e) L^2 error for the Temperature.

and $(u \cdot \nabla s, s) = 0$, $\forall u \in H, s \in W_h$, we obtain

$$\begin{aligned} \|T_h^{k+1}\|_0^2 + \|2T_h^{k+1} - T_h^k\|_0^2 + \|\delta\delta T_h^{k+1}\|_0^2 - \|T_h^k\|_0^2 - \|2T_h^k - T_h^{k-1}\|_0^2 \\ + 2\tau\kappa\|\nabla T_h^{k+1}\|_0^2 \leq c_0\|\gamma_h\|_{-1}^2. \end{aligned} \quad (4.16)$$

Next, we square them and sum the results. During this process, we use the following inequalities:

$$\|B_h v_h\|_0 \leq \|\nabla v_h\|_0, \quad \forall v_h \in X_h; \quad \|i_h^T y_h\|_0 \leq \|y_h\|_0, \quad \forall y_h \in Y_h.$$

First, we square (4.13) and get

$$\begin{aligned} 9\|\delta u_h^{k+1}\|_0^2 + 4\tau^2\|C_h^T \phi_h^{k+1}\|_0^2 = \left\|4\delta\tilde{u}_h^k - \delta\tilde{u}_h^{k-1}\right\|_0^2 + 4\tau^2\left\|D_h' \tilde{u}_h^k\right\|_0^2 + 4\tau^2\left\|(C_h^T - i_h B_h^T)\delta\phi_h^k\right\|_0^2 \\ - 8\tau^2\left(\left(2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}\right) \cdot \nabla\right) i_h \tilde{u}_h^k \\ - \left(\left(2\tilde{u}_h^{k-2} - \tilde{u}_h^{k-3}\right) \cdot \nabla\right) i_h \tilde{u}_h^{k-1}, D_h' \tilde{u}_h^k \\ - 4\tau\left(4\delta\tilde{u}_h^k - \delta\tilde{u}_h^{k-1}, D_h' \tilde{u}_h^k\right) + c_1\|\delta T_h^k\|_0^2. \end{aligned} \quad (4.17)$$

where c_1 is a constant.

Note that we use a equality $(i_h v_h, (C_h^T - i_h B_h^T)q_h) = 0$ for all $v_h \in X_h$ and all $q_h \in M_h$, because $i_h^T i_h$ is the identity and $i_h^T C_h^T = B_h^T$.

Second, we square (4.14) and have

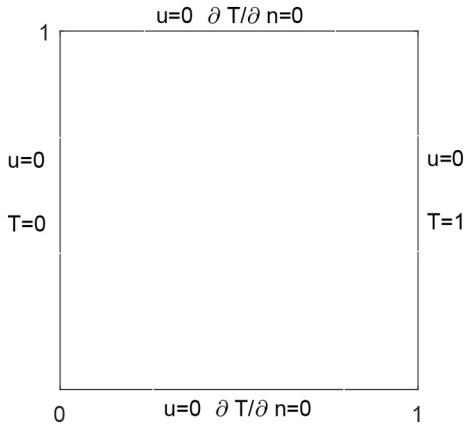


Fig. 2. Thermal driven cavity flow setting.

where c_2 is a constant.

Third, we square (4.15) and obtain

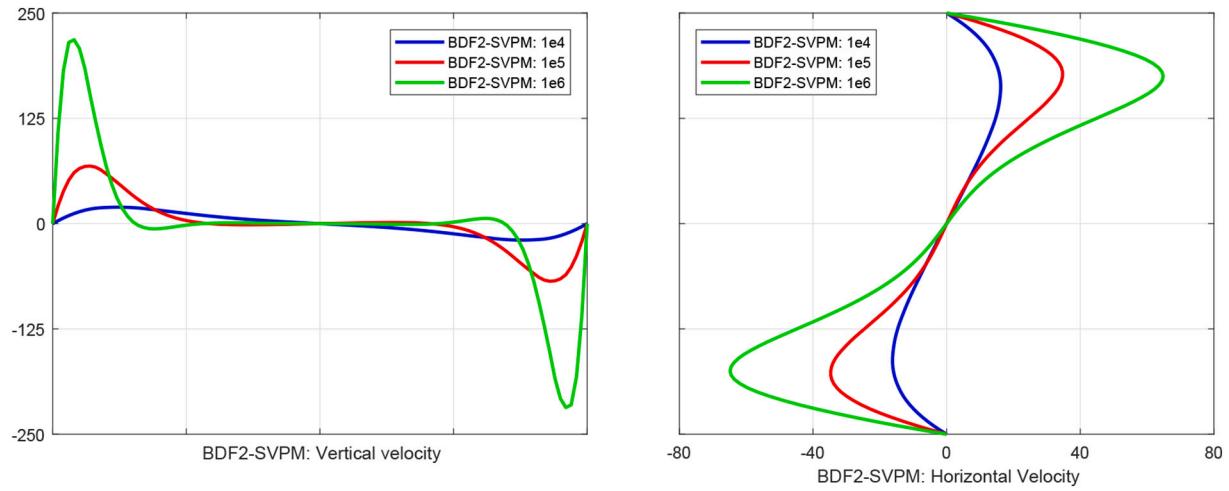
$$\begin{aligned} 4\tau^2\|B_h^T \phi_h^{k+1}\|_0^2 + c_2\|\delta T_h^k\|_0^2 = \left\|3\delta\tilde{u}_h^{k+1} - 4\delta\tilde{u}_h^k + \delta\tilde{u}_h^{k-1}\right\|_0^2 + 4\tau^2\left\|D_h' \tilde{u}_h^{k+1}\right\|_0^2 \\ + 4\tau\left(3\delta\tilde{u}_h^{k+1} - 4\delta\tilde{u}_h^k + \delta\tilde{u}_h^{k-1}, D_h' \tilde{u}_h^{k+1}\right) \\ + 8\tau^2\left(\left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}\right) \cdot \nabla\right) \tilde{u}_h^{k+1} - \left(\left(2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}\right) \cdot \nabla\right) \tilde{u}_h^k, D_h' \tilde{u}_h^{k+1} \end{aligned} \quad (4.18)$$

Table 1Comparison of maximum vertical velocity at $y = 0.5$ with mesh size used in computation.

Ra	GFEM	BDF2-SVPM	BDF2-RVPM	Ref. [20]	Ref. [21]	Ref. [22]
10^4	16.41(11×11)	19.52(11×11)	15.98(11×11)	19.91(11×11)	19.51(41×41)	19.63(71×71)
10^5	51.22(21×21)	68.42(21×21)	58.83(21×21)	70.60(21×21)	68.22(81×81)	68.85(71×71)
10^6	201.20(32×32)	218.28(32×32)	189.34(32×32)	228.12(32×32)	216.75(81×81)	221.60(71×71)

Table 2Comparison of maximum horizontal velocity at $x = 0.5$ with mesh size used in computation.

Ra	GFEM	BDF2-SVPM	BDF2-RVPM	Ref. [20]	Ref. [21]	Ref. [22]
10^4	15.70(11×11)	16.19(11×11)	13.70(11×11)	15.90(11×11)	16.18(41×41)	16.10(71×71)
10^5	41.00(21×21)	34.70(21×21)	30.76(21×21)	33.51(21×21)	34.81(81×81)	34.00(71×71)
10^6	80.25(32×32)	64.81(32×32)	57.97(32×32)	65.52(32×32)	65.33(81×81)	65.40(71×71)

**Fig. 3.** BDF2-SVPM: variation of vertical velocity at mid-height (left) and horizontal velocity at mid-width for varying Rayleigh numbers (right).

$$\begin{aligned} 9\|i_h^T \delta u_h^{k+1}\|_0^2 &= 9\left\|\delta \tilde{u}_h^{k+1}\right\|_0^2 + 4\tau^2\left\|D_h' \delta \tilde{u}_h^{k+1}\right\|_0^2 + 4\tau\left(3\delta \tilde{u}_h^{k+1}, D_h' \delta \tilde{u}_h^{k+1}\right) \\ &\quad + 8\tau^2\left(\left(\left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}\right) \cdot \nabla\right)\tilde{u}_h^{k+1} - \left(\left(2\tilde{u}_h^{k-2} - \tilde{u}_h^{k-3}\right) \cdot \nabla\right)\tilde{u}_h^{k-1}, D_h' \delta \tilde{u}_h^{k+1}\right) \end{aligned} \quad (4.19)$$

For the sake of simplicity, we sum up (4.17–4.19) and set some notation

$$\begin{aligned} \alpha_3 &:= 8\tau^2\left(\left(\left(2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}\right) \cdot \nabla\right)i_h \tilde{u}_h^k - \left(\left(2\tilde{u}_h^{k-2} - \tilde{u}_h^{k-3}\right) \cdot \nabla\right)i_h \tilde{u}_h^{k-1}, D_h' \tilde{u}_h^k\right) \\ &\quad + \left(\left(\left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}\right) \cdot \nabla\right)\tilde{u}_h^{k+1} - \left(\left(2\tilde{u}_h^{k-1} - \tilde{u}_h^{k-2}\right) \cdot \nabla\right)\tilde{u}_h^k, D_h' \tilde{u}_h^{k+1}\right) \\ &\quad + \left(\left(\left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}\right) \cdot \nabla\right)\tilde{u}_h^{k+1} - \left(\left(2\tilde{u}_h^{k-2} - \tilde{u}_h^{k-3}\right) \cdot \nabla\right)\tilde{u}_h^{k-1}, D_h' \delta \tilde{u}_h^{k+1}\right) \\ &= 8\tau^2\left(\left(\left(2\tilde{u}_h^k - \tilde{u}_h^{k-1}\right) \cdot \nabla\right)\tilde{u}_h^{k+1}, D_h' \delta \tilde{u}_h^{k+1}\right) \end{aligned}$$

$$\begin{aligned} \alpha_1 &:= \left\|3\delta \tilde{u}_h^{k+1} - 4\delta \tilde{u}_h^k + \delta \tilde{u}_h^{k-1}\right\|_0^2 - \left\|4\delta \tilde{u}_h^k - \delta \tilde{u}_h^{k-1}\right\|_0^2, \\ \alpha_2 &:= 4\tau\left[\left(3\delta \tilde{u}_h^{k+1}, D_h' \delta \tilde{u}_h^{k+1}\right) + \left(3\delta \tilde{u}_h^{k+1} - 4\delta \tilde{u}_h^k + \delta \tilde{u}_h^{k-1}, D_h' \tilde{u}_h^{k+1}\right) + \left(4\delta \tilde{u}_h^k - \delta \tilde{u}_h^{k-1}, D_h' \tilde{u}_h^k\right)\right] \\ &= 4\tau\left[\left(3\delta \tilde{u}_h^{k+1}, D_h' \tilde{u}_h^{k+1}\right) + \left(3\delta \tilde{u}_h^{k+1} - 4\delta \tilde{u}_h^k + \delta \tilde{u}_h^{k-1}, D_h' \delta \tilde{u}_h^{k+1}\right)\right], \end{aligned}$$

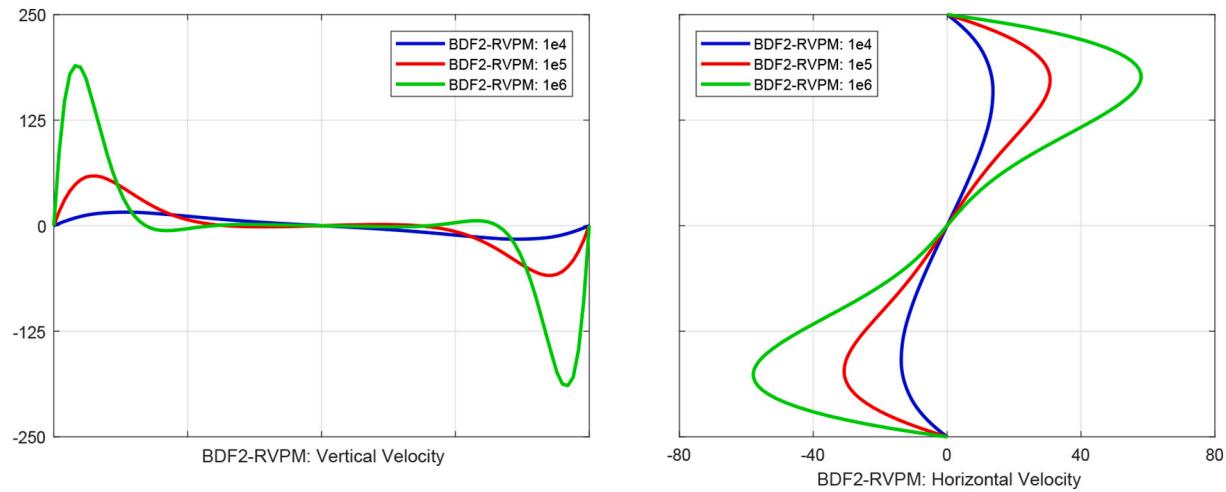


Fig. 4. BDF2-RVPM: variation of vertical velocity at mid-height (left) and horizontal velocity at mid-width for varying Rayleigh numbers (right).

Table 3

Comparison of average Nusselt number on the vertical boundary of the cavity at $x = 0$ with mesh size used in computation.

Ra	GFEM	BDF2-SVPM	BDF2-RVPM	Ref. [20]	Ref. [21]	Ref. [22]
10^4	2.40(11×11)	2.27(11×11)	2.03(11×11)	2.15(11×11)	2.24(41×41)	2.08(71×71)
10^5	5.11(21×21)	4.59(21×21)	4.16(21×21)	4.35(21×21)	4.52(81×81)	4.30(71×71)
10^6	6.00(32×32)	9.09(32×32)	8.32 (32×32)	8.83(32×32)	8.92(81×81)	8.74(71×71)

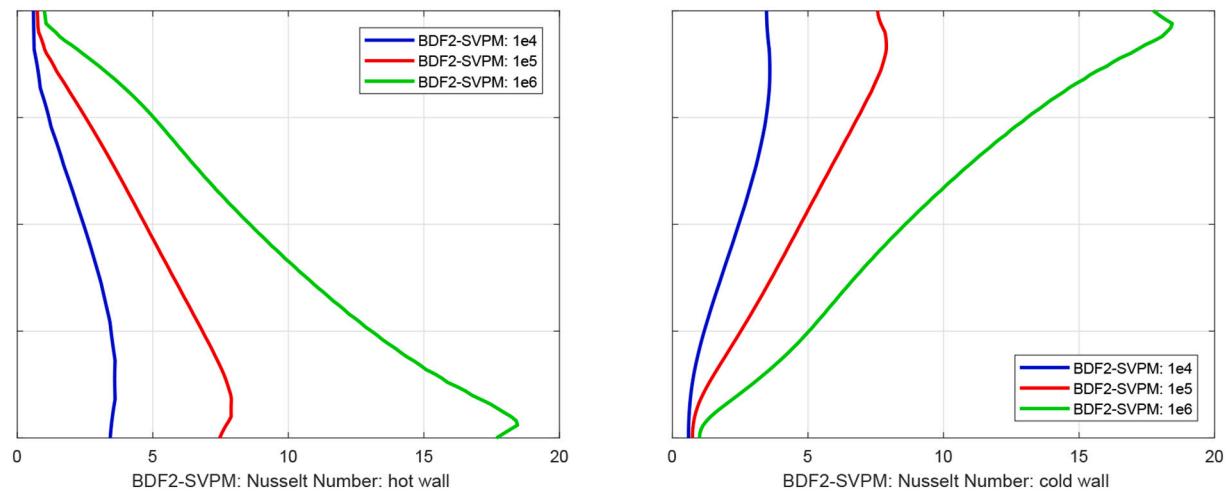


Fig. 5. BDF2-SVPM: variation of local Nusselt number at cavity hot wall (left) and cavity cold wall (right).

then, we can have

$$\begin{aligned}
& 9\|\delta u_h^{k+1}\|_0^2 + 4\tau^2 \left(\|C_h^T \phi_h^{k+1}\|_0^2 - \|B_h^T \phi_h^{k+1}\|_0^2 \right) + 4\tau^2 \left\| D_h^t \tilde{u}_h^{k+1} \right\|_0^2 + 9\left\| \delta \tilde{u}_h^{k+1} \right\|_0^2 \\
& + \alpha_1 + \alpha_2 + \alpha_3 + 4\tau^2 \left\| D_h^t \delta \tilde{u}_h^{k+1} \right\|_0^2 \\
& = 4\tau^2 \left\| D_h^t \tilde{u}_h^k \right\|_0^2 + 4\tau^2 \left\| (C_h^T - i_h B_h^T) \delta \phi_h^k \right\|_0^2 + 9\left\| i_h^T \delta \tilde{u}_h^{k+1} \right\|_0^2 + c\left\| \delta T_h^k \right\|_0^2.
\end{aligned}$$

Combining

$$\|C_h^T \phi_h^{k+1}\|_0^2 - \|B_h^T \phi_h^{k+1}\|_0^2 = \|(C_h^T - i_h B_h^T) \phi_h^{k+1}\|_0^2, \quad \|i_h^T \delta \tilde{u}_h^{k+1}\|_0 \leq \|\delta \tilde{u}_h^{k+1}\|_0,$$

we can deduce

$$\begin{aligned}
& 9\left\| \delta \tilde{u}_h^{k+1} \right\|_0^2 + 4\tau^2 \left\| (C_h^T - i_h B_h^T) \phi_h^{k+1} \right\|_0^2 + 4\tau^2 \left\| D_h^t \tilde{u}_h^{k+1} \right\|_0^2 + \alpha_1 + \alpha_2 + \alpha_3 + 4\tau^2 \left\| D_h^t \delta \tilde{u}_h^{k+1} \right\|_0^2 \\
& \leq 4\tau^2 \left\| D_h^t \tilde{u}_h^k \right\|_0^2 + 4\tau^2 \left\| (C_h^T - i_h B_h^T) \delta \phi_h^k \right\|_0^2 + c\left\| \delta T_h^k \right\|_0^2.
\end{aligned} \tag{4.20}$$

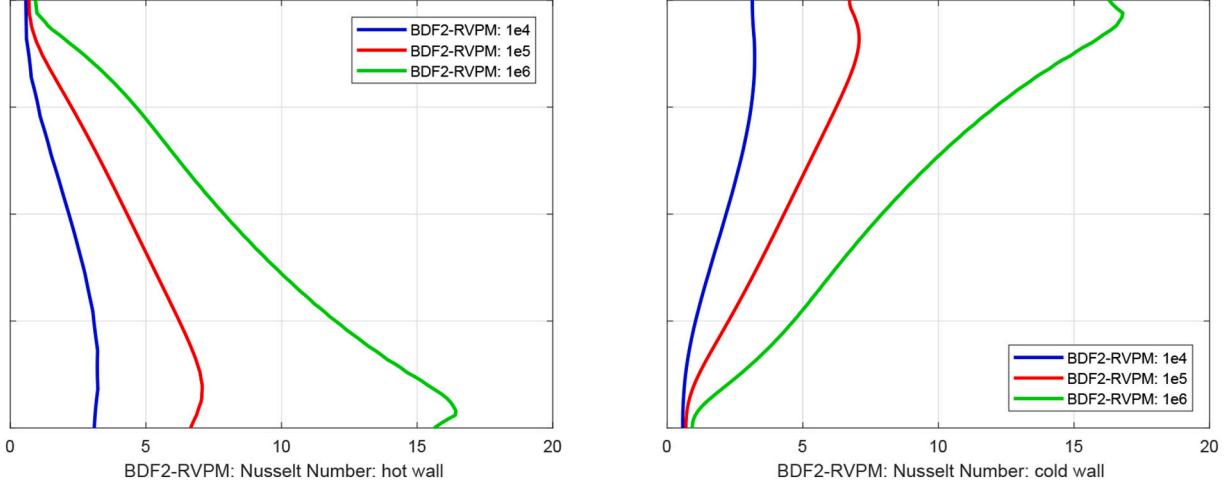


Fig. 6. BDF2-RVPM: variation of local Nusselt number at cavity hot wall (left) and cavity cold wall (right).

Then, we handle the $\alpha_1, \alpha_2, \alpha_3$ as follow:

$$\begin{aligned} \alpha_1 &:= \left\| 3\delta\tilde{u}_h^{k+1} - 4\delta\tilde{u}_h^k + \delta\tilde{u}_h^{k-1} \right\|_0^2 - \left\| 4\delta\tilde{u}_h^k - \delta\tilde{u}_h^{k-1} \right\|_0^2 \\ &= -6 \left\| \delta\tilde{u}_h^{k+1} \right\|_0^2 + 3 \left\| 2\delta\tilde{u}_h^{k+1} - \delta\tilde{u}_h^k \right\|_0^2 + 3 \left\| \delta^3\tilde{u}_h^{k+1} \right\|_0^2 - 3 \left\| \delta\tilde{u}_h^k \right\|_0^2 - 3 \left\| 2\delta\tilde{u}_h^k - \delta\tilde{u}_h^{k-1} \right\|_0^2. \end{aligned} \quad (4.21)$$

Then, in order to deal with α_2 , we need use a equality $2(3a - 4b + c, a - b) = 5(a - b)^2 + (a - 2b + c)^2 - (b - c)^2$. We obtain

$$\begin{aligned} &3 \left\| \delta\tilde{u}_h^{k+1} \right\|_0^2 + 6 \left\| B_h\tilde{u}_h^{k+1} \right\|_0^2 + 3 \left\| 2\delta\tilde{u}_h^{k+1} - \delta\tilde{u}_h^k \right\|_0^2 + 8 \left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 + 4 \left\| \nabla\delta^2\tilde{u}_h^{k+1} \right\|_0^2 \\ &+ 4\tau^2 \left\| (C_h^T - i_h B_h^T) \phi_h^{k+1} \right\|_0^2 + 4\tau^2 \left\| D'_h\tilde{u}_h^{k+1} \right\|_0^2 + \left\| T_h^{k+1} \right\|_0^2 + \left\| 2T_h^{k+1} - T_h^k \right\|_0^2 \\ &+ 3 \left\| \delta^2\tilde{u}_h^{k+1} \right\|_0^2 + 4 \left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 + 4\tau^2 \left\| D'_h\delta\tilde{u}_h^{k+1} \right\|_0^2 + \left\| \delta\delta T_h^{k+1} \right\|_0^2 \\ &+ 2\tau \left\| \nabla T_h^{k+1} \right\|_0^2 + 8\tau^2 \left\| \nabla \left(2\tilde{u}_h^k - \tilde{u}_h^{k-1} \right) \right\|_0^2 \left\| \nabla\tilde{u}_h^{k+1} \right\|_0^2 \left\| \nabla D'_h\tilde{u}_h^{k+1} \right\|_0^2 \\ &\leq 3 \left\| \delta\tilde{u}_h^k \right\|_0^2 + 6 \left\| B_h\tilde{u}_h^k \right\|_0^2 + 3 \left\| 2\delta\tilde{u}_h^k - \delta\tilde{u}_h^{k-1} \right\|_0^2 + 8 \left\| \nabla\delta\tilde{u}_h^k \right\|_0^2 + 4 \left\| \nabla\delta^2\tilde{u}_h^k \right\|_0^2 \\ &+ 4\tau^2 \left\| (C_h^T - i_h B_h^T) \delta\phi_h^k \right\|_0^2 + 4\tau^2 \left\| D'_h\tilde{u}_h^k \right\|_0^2 + \left\| T_h^k \right\|_0^2 + \left\| 2T_h^k - T_h^{k-1} \right\|_0^2 + c \|\gamma_h\|_{-1}^2, \end{aligned}$$

$$\begin{aligned} \alpha_2 &= 12\tau \left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 + 12\tau \left(\delta\tilde{u}_h^{k+1}, B_h^T B_h\tilde{u}_h^k \right) + 4\tau \left(3\delta\tilde{u}_h^{k+1} - 4\delta\tilde{u}_h^k + \delta\tilde{u}_h^{k-1}, A_h \delta^2\tilde{u}_h^{k+1} + B_h^T B_h\delta\tilde{u}_h^k \right) \\ &= 12\tau \left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 + 10\tau \left\| \nabla\delta^2\tilde{u}_h^{k+1} \right\|_0^2 + 2\tau \left\| \nabla\delta^3\tilde{u}_h^{k+1} \right\|_0^2 - 2\tau \left\| \nabla\delta^2\tilde{u}_h^k \right\|_0^2 + 6\tau \left\| B_h\tilde{u}_h^{k+1} \right\|_0^2 \\ &- 6\tau \left\| B_h\tilde{u}_h^k \right\|_0^2 - 6\tau \left\| B_h\delta^2\tilde{u}_h^{k+1} \right\|_0^2 - 8\tau \left\| B_h\delta\tilde{u}_h^k \right\|_0^2 - 2\tau \left\| B_h\delta^2\tilde{u}_h^k \right\|_0^2 + 2\tau \left\| B_h\delta\tilde{u}_h^{k-1} \right\|_0^2. \end{aligned}$$

Next, combining the inequation $\|B_h v_h\|_0 \leq \|\nabla v_h\|_0$ for all $v_h \in X_h$, we can get

$$\begin{aligned} \alpha_2 &\geq 4\tau \left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 + 8\tau \left(\left\| \nabla\delta\tilde{u}_h^{k+1} \right\|_0^2 - \left\| \nabla\delta\tilde{u}_h^k \right\|_0^2 \right) + 4\tau \left(\left\| \nabla\delta^2\tilde{u}_h^{k+1} \right\|_0^2 - \left\| \nabla\delta^2\tilde{u}_h^k \right\|_0^2 \right) \\ &+ 6\tau \left(\left\| B_h\tilde{u}_h^{k+1} \right\|_0^2 - \left\| B_h\tilde{u}_h^k \right\|_0^2 \right) \end{aligned} \quad (4.22)$$

Finally, using the property of $((u \cdot \nabla)v, w)$ for all $u, v, w \in X_h$. We have

$$\begin{aligned} \alpha_3 &= 8\tau \left(\left((2\tilde{u}_h^k - \tilde{u}_h^{k-1}) \cdot \nabla \right) \tilde{u}_h^{k+1}, D'_h\delta\tilde{u}_h^{k+1} \right) \\ &\leq 8\tau^2 \left\| \nabla \left(2\tilde{u}_h^k - \tilde{u}_h^{k-1} \right) \right\|_0^2 \left\| \nabla\tilde{u}_h^{k+1} \right\|_0^2 \left\| \nabla D'_h\delta\tilde{u}_h^{k+1} \right\|_0^2. \end{aligned} \quad (4.23)$$

From the above, we use (4.16, 4.20, 4.21, 4.22, 4.23), then we obtain

where c is a generic constant. Doing suitable scaling and summing it from 1 to n , we obtain the final conclusion

$$\begin{aligned} &3 \left\| \delta\tilde{u}_h^{n+1} \right\|_0^2 + 6 \left\| B_h\tilde{u}_h^{n+1} \right\|_0^2 + 3 \left\| 2\delta\tilde{u}_h^{n+1} - \delta\tilde{u}_h^n \right\|_0^2 + 8 \left\| \nabla\delta\tilde{u}_h^{n+1} \right\|_0^2 + 4 \left\| \nabla\delta^2\tilde{u}_h^{n+1} \right\|_0^2 \\ &+ 4\tau^2 \left\| (C_h^T - i_h B_h^T) \phi_h^{n+1} \right\|_0^2 + 4\tau^2 \left\| D'_h\tilde{u}_h^{n+1} \right\|_0^2 + \left\| T_h^{n+1} \right\|_0^2 + \left\| 2T_h^{n+1} - T_h^n \right\|_0^2 \\ &\leq 3 \left\| \delta\tilde{u}_h^1 \right\|_0^2 + 6 \left\| B_h\tilde{u}_h^1 \right\|_0^2 + 3 \left\| 2\delta\tilde{u}_h^1 - \delta\tilde{u}_h^0 \right\|_0^2 + 8 \left\| \nabla\delta\tilde{u}_h^1 \right\|_0^2 + 4 \left\| \nabla\delta^2\tilde{u}_h^1 \right\|_0^2 \\ &+ 4\tau^2 \left\| (C_h^T - i_h B_h^T) \delta\phi_h^1 \right\|_0^2 + 4\tau^2 \left\| D'_h\tilde{u}_h^1 \right\|_0^2 + \left\| T_h^1 \right\|_0^2 + \left\| 2T_h^1 - T_h^0 \right\|_0^2 + c \|\gamma_h\|_{C(0,T;H^{-1})}^2. \end{aligned}$$

where c is a generic constant.

Remark 4.1. The above algorithms can generalize to time-marching algorithms. For example, assuming v to be a smooth function, qth-

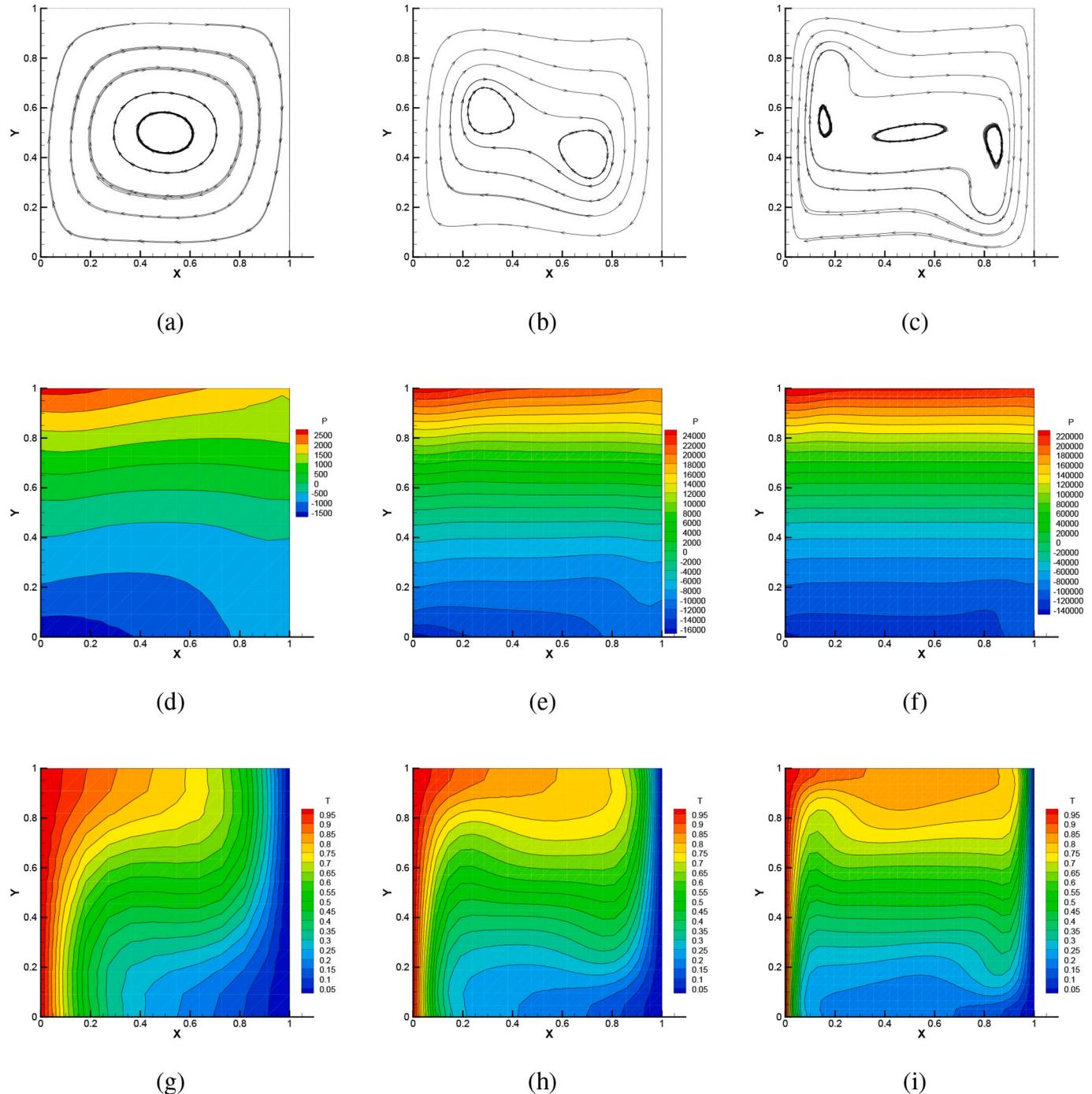


Fig. 7. The numerical streamlines, the isobars and the isotherms of BDF2-SVPM as $Pr = 0.71$, $\tau = 0.001$. The first column: $Ra = 10^4$, the second column: $Ra = 10^5$, the third column: $Ra = 10^6$, respectively.

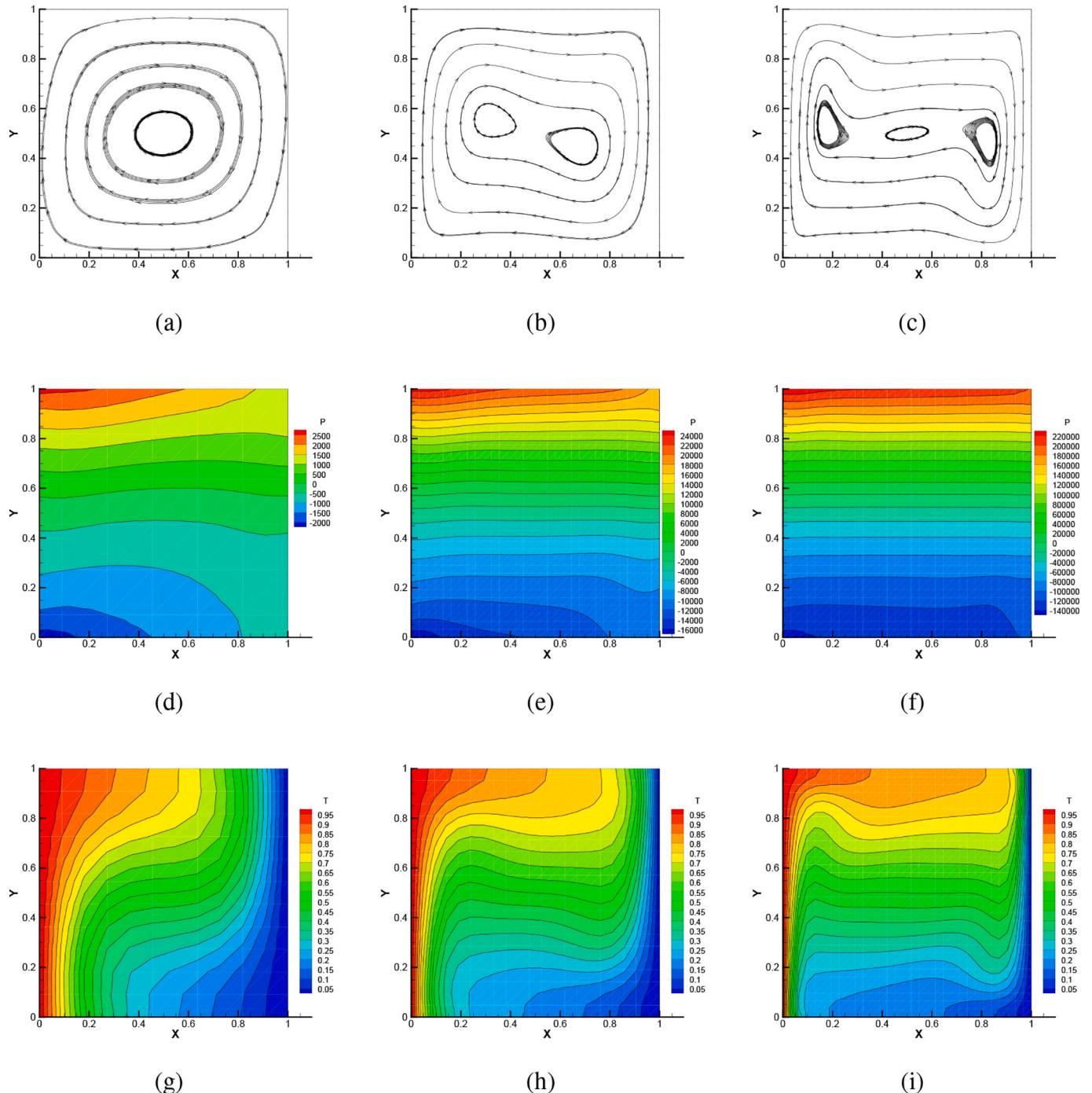


Fig. 8. The numerical streamlines, the isobars and the isotherms of BDF2-RVPM as $Pr = 0.71$, $\tau = 0.001$. The first column: $Ra = 10^4$, the second column: $Ra = 10^5$, the third column: $Ra = 10^6$, respectively.

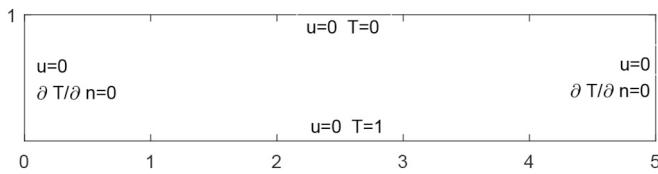


Fig. 9. Bénard convection setting.

order backward difference formula (BDF q) is denoted by $\frac{1}{\tau} \left(\alpha_q v^{k+1} - \sum_{j=0}^{q-1} \alpha_j v^{k-j} \right)$ that approximates $\partial_t v(t^{k+1})$. For convenience, for any sequence $\phi_{\Delta t} := (\phi^0, \phi^1, \dots)$, we set

$$D\phi^{k+1} = \alpha_q \phi^{k+1} - \sum_{j=0}^{q-1} \alpha_j \phi^{k-j}.$$

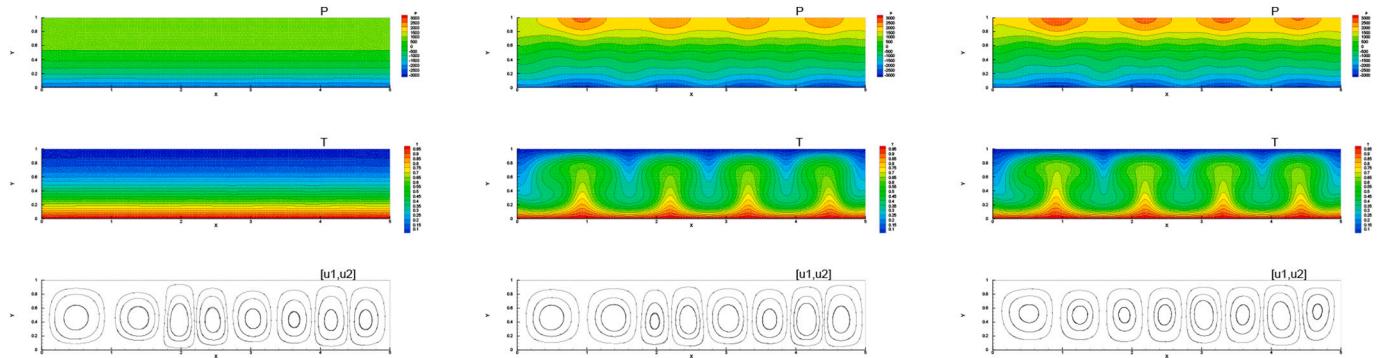


Fig. 10. Numerical streamlines, isotherms and isobars of the Bénard convection problem based on BDF2-SVPM. The first column: $t = 0.15$, the second column: $t = 0.35$, the third column: $t = 0.55$, respectively.

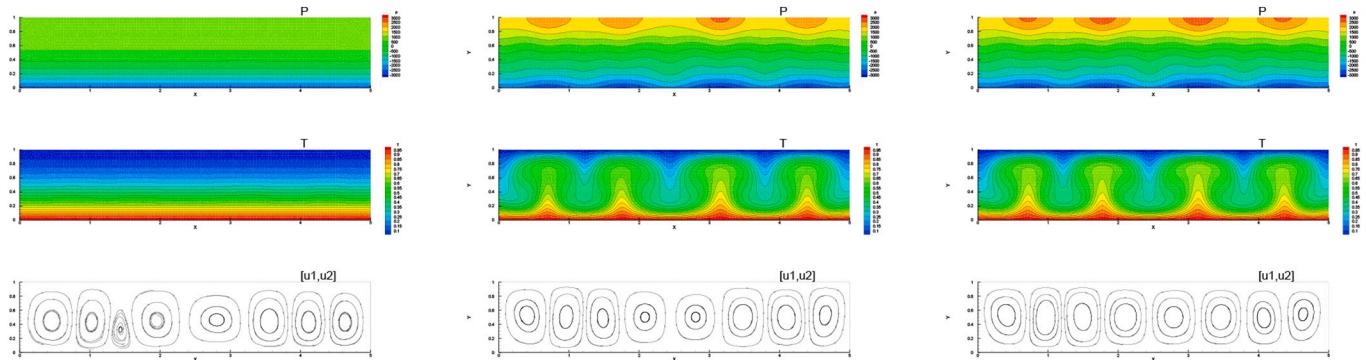


Fig. 11. Numerical streamlines, isotherms and isobars of the Bénard convection problem based on BDF2-RVPM. The first column: $t = 0.15$, the second column: $t = 0.35$, the third column: $t = 0.55$, respectively.

Now, we give a general rotational form of velocity-correction schemes:

Algorithm 4.2.4. BDFq-RVPM.

Step I. For $k \geq q$, we find $u^{k+1} \in H(\Omega)^2$, $p^{k+1} \in L_0^2$, such that

$$\begin{cases} \frac{1}{2\tau}(3T^{k+1} - 4T^k + T^{k-1}) - \kappa\Delta T^k + (\tilde{u}^{k+1} \cdot \nabla)T^{k+1} = \gamma^{k+1}, \\ T^{k+1}|_{\partial\Omega} = T_0. \end{cases}$$

$$\begin{cases} \frac{1}{\tau}\left(\alpha_q u^{k+1} - \sum_{j=0}^{q-1} \alpha_j \tilde{u}^{k-j}\right) + Pr\nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = PrRajT^k, \\ \nabla \cdot u^{k+1} = 0, \\ u^{k+1}|_{\partial\Omega} = 0, \end{cases}$$

where $\tilde{u}^0 = u^0(x)$ and \tilde{u}^i ($i = 1, \dots, k$), T^k can be computed by lower order method.

Step II. We find $\tilde{u}^{k+1} \in H_0^1(\Omega)^2$ such that

$$\begin{cases} \frac{\alpha_q}{\tau}(\tilde{u}^{k+1} - u^{k+1}) + Pr\nabla \times \nabla \times \tilde{u}^k + \nabla p^{k+1} + ((2\tilde{u}^k - \tilde{u}^{k-1}) \cdot \nabla) \tilde{u}^{k+1} - ((2\tilde{u}^{k-1} - \tilde{u}^{k-2}) \cdot \nabla) \tilde{u}^k = 0, \\ \tilde{u}^{k+1}|_{\partial\Omega} = 0. \end{cases}$$

Step III. We find $T^{k+1} \in W$ such that

5. Numerical experiments

In this section, we present three numerical experiments to verify the

stability of the velocity corrected projection method for solving

unsteady natural convective equations. First, we make convergence tests to check the theoretical analysis in the previous sections. Then we give a numerical experiment is the thermal driven cavity flow problem. Last, we give an example is Bénard convection problem. All examples in this chapter use (P_2, P_1, P_2) finite element pairs.

5.1. A numerical example with the exact solution

First, we consider the exact solution problem to investigate the convergence rate estimates. Let Ω be the unit square in \mathbb{R}^2 . We set the source term so that the exact solution is given as follows:

$$\begin{cases} u_1(x, y) &= 10x^2(x-1)^2y(y-1)(2y-1)\cos(t), \\ u_2(x, y) &= -10x(x-1)(2x-1)y^2(y-1)^2\cos(t), \\ p(x, y) &= 10(2x-1)(2y-1)\cos(t), \\ T(x, y) &= u_1(x, y) + u_2(x, y) \end{cases}$$

And we choose the parameters: $\tau = h$, $T^* = 0.1$, $\kappa = 1$, $Pr = 1$ and $Ra = 1$.

From Fig. 1, we compared the convergence efficiency of BDF2 standard pressure-correction projection methods (BDF2-SPPM), BDF2 rotational pressure-correction projection methods (BDF2-RPPM) in [18] and BDF2-SVPM, BDF2-RVPM with respect to velocity, pressure and temperature. It is not difficult for us to find that our proposed velocity correction algorithm shows great advantages in the accuracy of velocity and pressure. We can also see that BDF2-RVPM is better than BDF2-SVPM. Because in Fig. 1c, we can find that pressure in BDF2-RVPM is order 2 accurate in the L^2 -norm and in BDF2-SVPM is order 1.5.

5.2. Thermal driven cavity flow problem

In this section, we consider the thermal driven cavity flow problem which has no exact solution to test our algorithm. We consider the boundary conditions like Fig. 2: left bounds $T = 1$ and lower bounds $T = 0$, upper and lower boundaries $\frac{\partial T}{\partial n} = 0$ and on all boundaries $u = 0$. In the test, we take $\kappa = 1$, $\tau = 0.001$, $Pr = 0.71$ and the initial conditions: $u^0 = (0, 0)$, $T^0 = 0$. We have performed our computations for Rayleigh number as: $Ra = 10^4, 10^5, 10^6$. Then, we compared the numerical results with the famous benchmark solutions of de Vahl Davis [20] and some other authors such as Aytekin Çbk et al. [21], Massarotti et al. [22], Manzari [23]. In [20], it used second-order central approximations to solve natural convection problem in a square cavity. Ref. [22] used a projection-based stabilized finite element method. Ref. [22] used a semi-implicit form of the characteristic-based split scheme. Ref. [23] used an explicit finite element algorithm. We also compared the results with the classical Galerkin Finite Element Method (GFEM), and we use the same mesh sizes (11×11 , 21×21) for the proposed method and GFEM.

In Tables 1 and 2, we give extreme values of vertical velocity and horizontal velocity at $y = 0.5/x = 0.5$ for different Rayleigh numbers. We compare the result with [20–22]. We can find our results of BDF2-SVPM are agreement with the benchmark data. We also can see that as the Rayleigh number increases, BDF2-RVPM and GFEM results which are not close to the benchmark data.

Then, we provide the vertical velocity distribution at the mid-height and horizontal velocity distribution at the mid-width in Figs. 3–4. These results are compared with [21]. We can easily find, when Rayleigh numbers increases, the differences in the profiles presented in Figs. 3–4 are getting larger.

Next, we consider the rate of heat transfer along the vertical walls of the cavity which is called Nusselt number. And the local Nusselt number can be calculated as $Nu = \partial T/\partial x$. We compare the average Nusselt number with benchmark problems in Table 3. And we can understand from Table 3, our methods are agreement with the benchmark data. Comparing the variation of the Nusselt number along the hot wall and cold wall for different Rayleigh numbers in Figs. 5–6 with [20–22], these profiles are also look reasonable.

In the study of natural convection problems, we can often see diagrams showing the streamlines, pressure isolines and temperature isolines. We present these results in Figs. 7–8 and compare them with benchmark data in [20–22]. They have the same performance. However, the algorithm we proposed has some shortcomings and cannot calculate larger Ra . When $Ra \geq 1e7$, the velocity contour map becomes extremely distorted and irregular.

5.3. Bénard convection

In this subsection, we illustrate a classical fluid dynamics phenomenon: the Bénard convection problem. The domain and boundary conditions of the problem are displayed in Fig. 9. We give the initial conditions are given: $u^0 = (0, 0)$, $T^0 = 0$.

In Fig. 10, we present the numerical results by using BDF2-SVPM with $h = 1/60$, $Pr = 0.71$, $Ra = 10^4$, $\kappa = 1$, $\tau = 0.001$, at $t = 0.15$, $t = 0.35$ and $t = 0.55$. In Fig. 11, we present the numerical results by using BDF2-RVPM with $h = 1/60$, $Pr = 0.71$, $Ra = 10^4$, $\kappa = 1$, $\tau = 0.01$, at $t = 0.15$, $t = 0.35$ and $t = 0.55$. Compare our numerical results with [24, 25]. From these figures, we can see that our proposed methods can simulate the temperature field, pressure field and fluid field effectively and can deal with complex problems. As time goes on, And we can find the graph of BDF2-RVPM starts to distort before that of BDF2-SVPM.

6. Conclusions

Combining with extending for pressure-correction projection methods, we present two different versions of velocity-correction projection method for natural convection problem. And, we also use a decoupling technique in the process of derivation of our new method. Furthermore, the stability is derived for the proposed methods. Besides, the theoretical analysis is verified by the numerical results, which show that velocity-correction projection method is better than pressure-correction projection method and BDF2-RVPM is better than BDF2-SVPM. Besides, the new velocity-correction projection method can be extended to the time-dependent problems and other fluid dynamical models.

Declaration of Competing Interest

None.

Acknowledgements

The authors would like to thank the editors and referees for their valuable comments and suggestions which helped them to improve the results of this paper. Thanks to the following funds for their support of this work: the NSF of China (No. 11701493, No. 12061076, No. 12061075), the Natural Science Foundation of Xinjiang Province (No. 2016D01C073), the Tianshan Youth Project of Xinjiang Province (No. 2017Q079), the Key Research Project of Universities in the Xinjiang Uygur Autonomous Region (No. XJEDU2020I001), the Open Project of Key Laboratory of Xinjiang Uygur Autonomous Region (No. 2020D04002).

References

- [1] A.J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comput.* 22 (1968) 745–762.
- [2] R. Temam, Sur approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires II, *Arch. Ration. Mech. Anal.* 33 (1969) 377–385.
- [3] W. E, J.G. Liu, Projection method I: Convergence and numerical boundary layers, *SIAM J. Numer. Anal.* 32 (1995) 1017–1057.
- [4] J.L. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods, *Math. Comput.* 73 (2004) 1719–1737.
- [5] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.

- [6] A. Prohl, Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations, *Advances in Numerical Mathematics*, B.G. Teubner, Stuttgart, 1997.
- [7] J. Shen, On error estimates of the projection methods for the Navier-Stokes equations: first-order schemes, *SIAM J. Numer. Anal.* 29 (1992) 57–77.
- [8] L.J.P. Timmermans, P.D. Minev, F. N., D. Vosse, An approximate projection scheme for incompressible flow using spectral elements, *Int. J. Numer. Methods Fluids* 22 (1996) 673–688.
- [9] J.V. Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* 7 (1986) 870–891.
- [10] J.L. Guermond, J. Shen, Velocity-correction projection methods for incompressible flows, *SIAM J. Numer. Anal.* 41 (1) (2003) 112–134.
- [11] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 97 (1991) 414–443.
- [12] S.A. Orszag, M. Israeli, M.O. Deville, Boundary conditions for incompressible flows, *J. Sci. Comput.* 1 (1986) 75–111.
- [13] J.L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, *J. Comput. Phys.* 192 (2003) 262–276.
- [14] J. Shen, X. Yang, Error estimates for finite element approximations of consistent splitting schemes for incompressible flows, *DCDS-B.* 8 (2007) 663–676.
- [15] E. Weinan, Liu Jian-Guo, Gauge method for viscous incompressible flows, *Commun. Math. Sci.* 1 (2003) 317–332.
- [16] R.H. Nochetto, J.H. Pyo, Error estimates for semi-discrete gauge methods for the Navier-Stokes equations, *Math. Comput.* 74 (2005) 521–542.
- [17] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 97 (1991) 414–443.
- [18] J.L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 6011–6045.
- [19] J.L. Guermond, J. Shen, X. Yang, Error analysis of fully discrete velocity-correction methods for incompressible flows, *Math. Comput.* 77 (2008) 1387–1405.
- [20] G. De Vahl Davis, Natural convection of air in a square cavity: a benchmark solution, *Int. J. Numer. Meth. Fl.* 3 (1983) 249C264.
- [21] A. Cibik, S. Kaya, A projection-based stabilized finite element method for steady-state natural convection problem, *J. Math. Anal. Appl.* 381 (2011) 469–484.
- [22] N. Massarotti, P. Nithiarasu, O.C. Zienkiewicz, Characteristic-based-Split (CBS) algorithm for incompressible flow problems with heat transfer, *Int. J. Numer. Method. H.* 8 (1998) 969C990.
- [23] M.T. Manzari, An explicit finite element algorithm for convective heat transfer problems, *Int. J. Numer. Method. H* 9 (1999), 860C877.
- [24] Z. Si, X. Song, P. Huang, Modified Characteristics Gauge-Uzawa Finite Element Method for Time Dependent Conduction-Convection Problems, *J. Sci. Comput.* 58 (2014) 1–24.
- [25] J. Wu, P. Huang, X. Feng, A new variational multiscale FEM for the steady-state natural convection problem with bubble stabilization, *Numer. Heat. Trans. A-Appl.* 68 (2015) 777–796.