Journal of Computational Mathematics
Vol.xx, No.x, 2026, 1–19.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2512-m2024-0122

A HYBRID TWO-LEVEL FINITE ELEMENT METHOD FOR THE STATIONARY NATURAL CONVECTION PROBLEM AND PARALLEL IMPLEMENT*

Guoliang Zhang
School of Mathematics and Information Science & Center for Applied Mathematics of Guangxi, Guangxi University, Nanning 530004, China
Email: zglmath@gxu.edu.cn
Hongtao Chen
School of Mathematical Sciences, Xiamen University, Xiamen 361005, China
Email: chenht@xmu.edu.cn
Qiaoling He
College of Science, Shihezi University, Shihezi 832003, China
Email: hql2009_h@163.com
Jingwei Li[1)]
School of Mathematics and Statistic, Lanzhou University, Lanzhou 730000, China
Email: lijingwei@lzu.edu.cn

Abstract

This study presents a novel hybrid approach for addressing incompressible stationary natural convection problem, incorporating a parallel technique to enhance computational efficiency. Inspired by the traditional two-level method [He and Wang, Comput. Methods Appl. Mech. Engrg., 197 (2008)] and the two-step approach [Wu et al., Int. J. Heat Mass Transfer, 101 (2016)], both characterized by their iterative and corrective processes, we endeavor to alleviate the computational burden associated with the iterative process. Building upon these methods, our novel hybrid method involves two primary steps: initially solving the original problem using the finite element pair $P_1b - P_1 - P_1$ on a coarse mesh, followed by resolving the linearized equations using the higher-order pair $P_2 - P_1 - P_2$ on a fine mesh. While the first step employs iterative techniques, the second step entails directly solving a linearized problem. This novel approach can save lots of computational time in the iterative step compared to the traditional methods. Moreover, leveraging domain decomposition techniques, we implement a parallel strategy to further accelerate computations. Finally, we conduct several numerical examples to validate the efficiency of the proposed algorithms. The numerical results demonstrate optimal convergence rates comparable to those obtained using only the $P_2 - P_1 - P_2$ finite element pair under similar relative error conditions. Furthermore, the numerical simulations on the two obstacles flow and Bénard convection problem show the robustness and efficiency of the proposed algorithms.

Mathematics subject classification: 35B50, 35K55, 65M12, 65R20.
Key words: Natural convection problem, Finite element method, Hybrid two-level method, Parallel algorithm, Fully overlapping domain decomposition.

* Received June 20, 2024 / Revised version received November 11, 2024 / Accepted December 4, 2025 / Published online April 9, 2026 /
1) Corresponding author

1. Introduction

The natural convection problem is ubiquitous in the chemical industry, physical science, engineering field, and other areas such as double glass window design, room ventilation, and fire control [18]. In this paper, the natural convection problem takes the following form:

$$\begin{cases} -Pr\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = PrRa\,\mathbf{j}\,T & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega, \\ -\kappa\Delta T + \mathbf{u}\cdot\nabla T = \gamma & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{on } \partial\Omega, \\ T = T_0 & \text{on } \Gamma_1, \\ \dfrac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma_2 \end{cases} \tag{1.1}$$

with $\mathbf{u}$ being the velocity, p the pressure, T representing the temperature in an open, bounded and convex polygonal domain $\Omega \in \mathbb{R}^{\mathrm{d}}$ ($\mathrm{d} = 2, 3$) with a Lipschitz continuous boundary $\partial\Omega$, $\Gamma_2 = \partial\Omega/\Gamma_1$ is a regular open subset of $\partial\Omega$. γ is the prescribed forcing function, Pr, Ra and κ denotes the Prandtl number, the Rayleigh number and the thermal conductivity parameter, respectively. $\mathbf{j}$ is the unit vector in the direction of the gravity acceleration.

The difficulties in numerically solving the natural convection problem mainly lie in the incompressibility, the strong nonlinearity, and the coupling of the variables, which lead to a large numerical computational cost. There exist many works devoted to designing the fast and efficient numerical methods for the natural convection problem. A partial list includes the finite difference method [14], finite element method (FEM) [2, 23, 25, 28, 29], finite volume method [22]. Among them, the two-level method is a popular choice. Xu [26] first introduced a two-grid method for solving semi-linear elliptic equations, where the original problem is solved on a coarse mesh and a linear system is solved on a fine mesh using the same finite element pairs. Later, Layton [12] designed a two-level method to solve the Navier-Stokes equations. Furthermore, a simplified two-level method was proposed in [6] due to the ease to implement, which is then applied to solve the natural convection problems [9]. Additionally, some other variants of the two-level method have been developed to solve stationary conduction-convection equations [13, 21, 29]. Building on the two-level finite element method, Wu *et al.* [25] and Huang *et al.* [10] designed a two-step method, which can be regarded as the extension of two-level method, for solving the natural convection problem using different equal-order finite element pairs on a same mesh. It is well known that the equal-order finite element pairs fail to satisfy the so-called inf-sup condition [28], thus two stabilized terms based on local Gauss integration are adopted.

It is evident that the existing commendable works have been predominantly developed on the foundation of either the traditional two-level method or the two-step method. However, enhancing the efficiency of these two traditional algorithms poses certain challenges. Hence, inspired both the two-level method and the two-step method, we propose a hybrid two-level method which involves two primary steps: initially solving the original problem using the finite element pair $P_1b - P_1 - P_1$ on a coarse mesh, followed by resolving the linearized equations using the higher-order pair $P_2 - P_1 - P_2$ on a fine mesh. The novel algorithm exhibits several distinctions compared to traditional methods [6, 10, 25]:

- We leverage the strengths of both the two-level method [6] and the two-step method [25]. While the former employs same finite element pairs across different meshes, the latter

utilizes different finite element pairs within the same mesh. In contrast, our approach involves the use of different finite element pairs across different meshes, thereby reducing computational costs through the utilization of lower-order finite element pairs on coarse meshes in the first step.

- We carefully select stable finite element pairs that satisfy the inf-sup condition. Furthermore, our numerical results demonstrate that finite element pairs meeting the inf-sup condition yield an optimal convergence order.

The parallel strategy is also an alternative choice to improve computational efficiency. The main idea of parallel strategy is to divide the whole domain into multiple subdomains, which are then assigned to each processor. Shang and his collaborators [4,16,17,30,31] have done lots of works in parallel computation based on fully overlapping domain decomposition technique. Zhang *et al.* [27] combined the two-step method with the domain decomposition technique to devise a parallel two-step method to solve the Navier-Stokes equation. We then propose a parallel algorithm based on the fully overlapping domain decomposition technique to improve the efficiency of the hybrid two-level method for the natural convection problem.

The rest of this paper is structured as follows. Section 2 gives the review on some notations and well known results used throughout this paper. The local and parallel hybrid two-level algorithm and corresponding error estimates are given in Section 3. In Section 4, extensive numerical experiments are carried out to verify the efficiency and reliability of the proposed algorithms. Finally, we end this paper with a short conclusion in the last section.

2. Preliminaries

Let $\Omega \in \mathbb{R}^2$ be an open, bounded and convex domain with a Lipschitz continuous boundary $\partial\Omega$. For any nonnegative integer m, we denote by $H^m(\Omega)$ the Hilbert space of function with square integrable distribution derivatives up to order m in Ω, equipped with the standard norm $\|\cdot\|_m$ and semi-norm $|\cdot|_m$. Particularly, $L^2(\Omega) = H^0(\Omega)$ denotes the usual L^2 space with the corresponding norm $\|\cdot\|_0$. Throughout this paper, the letter c (with or without subscripts) denotes a generic positive constant that is independent of the mesh parameter and may take different values in different occurrences.

2.1. The natural convection problem

To set up the mathematical problem (1.1), we will begin by considering the following Hilbert spaces:

$$\mathbf{X} = H_0^1(\Omega)^2, \quad M = L_0^2(\Omega) = \left\{ q \in L^2(\Omega) : \int_\Omega q \mathrm{d}\mathbf{x} = 0 \right\},$$
$$W = \big\{ s \in H^1(\Omega) : s = 0 \text{ on } \Gamma_1 \big\}.$$

The weak formulation of original problem (1.1) is then given by: Find $(\mathbf{u}, p, T) \in \mathbf{X} \times M \times W$ such that for all $(\mathbf{v}, q, s) \in \mathbf{X} \times M \times W$

$$\begin{cases} \mathbb{B}\big((\mathbf{u}, p), (\mathbf{v}, q)\big) + b(\mathbf{u}, \mathbf{u}, \mathbf{v}) = PrRa(\mathbf{j}T, \mathbf{v}), \\ \kappa \widehat{a}(T, s) + \widehat{b}(\mathbf{u}, T, s) = (\gamma, s), \end{cases} \tag{2.1}$$

where the generalized bilinear form $\mathbb{B}((\cdot\,,\cdot), (\cdot\,,\cdot))$ is defined by

$$\mathbb{B}\big((\mathbf{u}, p), (\mathbf{v}, q)\big) = Pra(\mathbf{u}, \mathbf{v}) - d(\mathbf{v}, p) + d(\mathbf{u}, q), \quad \forall\, (\mathbf{u}, p), (\mathbf{v}, q) \in \mathbf{X} \times M.$$

The bilinear forms read as

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &= (\nabla\mathbf{u},\nabla\mathbf{v}), \quad \forall\,\mathbf{u},\mathbf{v}\in\mathbf{X},\\ \widehat{a}(T,s) &= (\nabla T,\nabla s), \quad \forall\,T,s\in W,\\ d(\mathbf{v},p) &= (p,\nabla\cdot\mathbf{v}), \quad \forall\,(\mathbf{v},p)\in\mathbf{X}\times M, \end{aligned}$$

and the trilinear forms

$$\begin{aligned} b(\mathbf{u},\mathbf{v},\mathbf{w}) &= \big((\mathbf{u}\cdot\nabla)\mathbf{v},\mathbf{w}\big)+\frac{1}{2}\big((\nabla\cdot\mathbf{u})\mathbf{v},\mathbf{w}\big)\\ &= \frac{1}{2}\big((\mathbf{u}\cdot\nabla)\mathbf{v},\mathbf{w}\big)-\frac{1}{2}\big((\mathbf{u}\cdot\nabla)\mathbf{w},\mathbf{v}\big), \quad \forall\,\mathbf{u},\mathbf{v},\mathbf{w}\in\mathbf{X},\\ \widehat{b}(\mathbf{u},T,s) &= \big((\mathbf{u}\cdot\nabla)T,s\big)+\frac{1}{2}\big((\nabla\cdot\mathbf{u})T,s\big)\\ &= \frac{1}{2}\big((\mathbf{u}\cdot\nabla)T,s\big)-\frac{1}{2}\big((\mathbf{u}\cdot\nabla)s,T\big), \qquad \forall\,\mathbf{u}\in\mathbf{X}, \quad \forall\,T,s\in W. \end{aligned}$$

It is well known that the trilinear forms satisfy the following properties:

$$\begin{aligned} &b(\mathbf{u},\mathbf{v},\mathbf{w}) = -b(\mathbf{u},\mathbf{w},\mathbf{v}), \quad b(\mathbf{u},\mathbf{v},\mathbf{v}) = 0,\\ &b(\mathbf{u},\mathbf{v},\mathbf{w}) \le N_1\|\nabla\mathbf{u}\|_0\|\nabla\mathbf{v}\|_0\|\nabla\mathbf{w}\|_0, \quad \forall\,\mathbf{u},\mathbf{v},\mathbf{w}\in\mathbf{X}, \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} &\widehat{b}(\mathbf{u},T,s) = -\widehat{b}(\mathbf{u},s,T), \quad \widehat{b}(\mathbf{u},T,T) = 0,\\ &\widehat{b}(\mathbf{u},T,s) \le N_2\|\nabla\mathbf{u}\|_0\|\nabla T\|_0\|\nabla s\|_0, \quad \forall\,(\mathbf{u},T,s)\in\mathbf{X}\times W\times W, \end{aligned} \tag{2.3}$$

where

$$\begin{aligned} N_1 &= \sup_{\substack{\mathbf{u},\mathbf{v},\mathbf{w}\in\mathbf{X},\\ \mathbf{u},\mathbf{v},\mathbf{w}\neq 0}} \frac{|b(\mathbf{u},\mathbf{v},\mathbf{w})|}{\|\nabla\mathbf{u}\|_0\|\nabla\mathbf{v}\|_0\|\nabla\mathbf{w}\|_0},\\ N_2 &= \sup_{\substack{\mathbf{u}\in\mathbf{X},T,s\in W\\ \mathbf{u},T,s\neq 0}} \frac{|\widehat{b}(\mathbf{u},T,s)|}{\|\nabla\mathbf{u}\|_0\|\nabla T\|_0\|\nabla s\|_0}. \end{aligned}$$

As we know, the above bilinear and trilinear forms satisfy the following properties.

(1) The inf-sup condition [1]

$$\beta\|q\|_0 \le \sup_{\substack{\mathbf{v}\in\mathbf{X},\\ \mathbf{v}\neq 0}} \frac{|(\operatorname{div}\mathbf{v},q)|}{\|\nabla\mathbf{v}\|_0}, \quad \forall\,q\in M, \tag{2.4}$$

where β is a positive constant depending on Ω.

(2) The Poincare's inequality

$$\|\mathbf{v}\|_0 \le c_0\|\nabla\mathbf{v}\|_0, \quad \forall\,\mathbf{v}\in\mathbf{X}. \tag{2.5}$$

(3) It holds for the trilinear forms that [1]

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) \le c_1\|\mathbf{u}\|_0\|\mathbf{v}\|_2\|\mathbf{w}\|_1, \quad \forall\,\mathbf{u}\in L^2(\Omega)^2, \quad \mathbf{v}\in H^2(\Omega)^2, \quad \mathbf{w}\in H_0^1(\Omega)^2, \tag{2.6}$$

$$\widehat{b}(\mathbf{u},T,s) \le c_2\|\mathbf{u}\|_0\|T\|_2\|s\|_1, \quad \forall\,\mathbf{u}\in L^2(\Omega)^2, \quad T\in H^2(\Omega), \quad s\in H_0^1(\Omega). \tag{2.7}$$

For problem (2.1), we have the following regulation result [1, 11].

Lemma 2.1. *Under conditions* (2.2)-(2.4)*, there exists at least a solution pair* $(\mathbf{u}, p, T) \in \mathbf{X} \times M \times W$ *to problem* (2.1) *satisfying*

$$\|\nabla \mathbf{u}\|_0 \leq Ra\,\kappa^{-1}\|\gamma\|_{-1}, \quad \|\nabla T\|_0 \leq \kappa^{-1}\|\gamma\|_{-1}$$

with

$$\|\gamma\|_{-1} = \sup_{s\in W} \frac{|(\gamma, s)|}{|s|_1}.$$

Furthermore, if Pr, Ra, κ *and* γ *satisfy*

$$0 < \sigma := Pr^{-1}Ra\,\kappa^{-1}\|\gamma\|_{-1}\big(N_1 + PrN_2\kappa^{-1}\big) < 1,$$

then the problem (2.1) *admits a unique solution pair* $(\mathbf{u}, p, T) \in \mathbf{X} \times M \times W$*.*

2.2. Mixed finite element approximation

In this section, two mixed finite element pairs to approximate the problem (2.1) and the corresponding error estimates will be given. For the finite element discretization, let Γ_μ be the triangulation partition of Ω with a mesh-size function $\mu(x)$, consisting of triangular elements K. Define the mesh parameter $\mu_\Omega = \max_{K\in\tau_\mu}\{\mathrm{diam}(K)\}$ where μ is a real positive parameter tending to 0. The finite element subspaces $\mathbf{X}_\mu \times M_\mu \times W_\mu$ of $\mathbf{X} \times M \times W$ is characterized by Γ_μ, which satisfy the following assumptions.

Assumption 2.1 (Triangulation). There exists a constant $\zeta \geq 1$ such that

$$\mu_\Omega^\zeta \leq c\mu(x), \quad \forall\, x \in \Omega.$$

Assumption 2.2 (Approximation). For each $(\mathbf{u}, p) \in H^{k+1}(\Omega)^d \times H^k(\Omega)(k \geq 1)$, there exists an approximation $(\pi_\mu\mathbf{u}, \rho_\mu p) \in X_\mu \times M_\mu$ such that

$$\|\mathbf{u} - \pi_\mu\mathbf{u}\|_1 \leq c\mu^s\|\mathbf{u}\|_{1+s}, \quad \|p - \rho_\mu p\|_0 \leq c\mu^s\|p\|_s, \quad 0 \leq s \leq k.$$

In this paper, we choose two examples of finite element pairs and denote by $P_i(K)$ the set of polynomials of degree no more than $i \in \mathbb{N}$ on K. We suppose $\mu = h$ and H, then the following two finite element pairs $(\mathbf{X}_H, M_H, W_H)$ and $(\mathbf{X}_h, M_h, W_h)$ will be focused. Furthermore, we shall assume these space are nested i.e. $(\mathbf{X}_H, M_H, W_H) \in (\mathbf{X}_h, M_h, W_h)$, which will significantly simplify our analysis, see e.g. [5].

Case 1. P_1b-P_1-P_1 element.

$$\begin{aligned}
\mathbf{X}_H &= \big\{\mathbf{v}_H \in \mathbf{X} :\ \mathbf{v}_H|_K \in (P_1 \oplus \mathrm{span}\{\widehat{b}\})^d,\ \forall\, K \in \Gamma_H\big\},\\
M_H &= \big\{q_H \in M \cap C^0(\Omega) :\ q_H|_K \in P_1,\ \forall\, K \in \Gamma_H\big\},\\
W_H &= \big\{s_H \in W \cap C^0(\Omega) :\ s_H|_K \in P_1,\ \forall\, K \in \Gamma_H\big\}.
\end{aligned}$$

Case 2. P_2-P_1-P_2 element.

$$\begin{aligned}
\mathbf{X}_h &= \big\{\mathbf{v}_h \in \mathbf{X} :\ \mathbf{v}_h|_K \in (P_2)^d,\ \forall\, K \in \Gamma_h\big\},\\
M_h &= \big\{q_h \in M \cap C^0(\Omega) :\ q_h|_K \in P_1,\ \forall\, K \in \Gamma_h\big\},\\
W_h &= \{s_h \in W :\ s_h|_K \in P_2,\ \forall\, K \in \Gamma_h\},
\end{aligned}$$

where $\widehat{b}$ is a bubble function. It is worth noting that both solution spaces $\mathbf{X}_H \times M_H$ and $\mathbf{X}_h \times M_h$ satisfy the following discrete inf-sup condition:

$$\beta_0 \|q_\mu\|_0 \leq \sup_{\substack{\mathbf{v}_\mu \in \mathbf{X}_\mu, \\ \mathbf{v}_\mu \neq 0}} \frac{(\operatorname{div} \mathbf{v}_\mu, q_\mu)}{\|\nabla \mathbf{v}_\mu\|_0}, \quad \forall\, q_\mu \in M_\mu, \tag{2.8}$$

where β_0 is a positive constant independent of h.

Then we obtain the discrete variational formulations: Find $(\mathbf{u}_H, p_H, T_H) \in \mathbf{X}_H \times M_H \times W_H$ such that for all $(\mathbf{v}_H, q_H, s_H) \in \mathbf{X}_H \times M_H \times W_H$,

$$\begin{cases} \mathbb{B}\big((\mathbf{u}_H, p_H), (\mathbf{v}_H, q_H)\big) + b(\mathbf{u}_H, \mathbf{u}_H, v_H) = PrRa(\mathbf{j}T_H, \mathbf{v}_H), \\ \kappa \widehat{a}(T_H, s_H) + \widehat{b}(\mathbf{u}_H, T_H, s_H) = (\gamma, s_H), \end{cases} \tag{2.9}$$

and find $(\mathbf{u}_h, p_h, T_h) \in \mathbf{X}_h \times M_h \times W_h$ such that for all $(\mathbf{v}_h, q_h, s_h) \in \mathbf{X}_h \times M_h \times W_h$,

$$\begin{cases} \mathbb{B}\big((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\big) + b(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) = PrRa(\mathbf{j}T_h, \mathbf{v}_h), \\ \kappa \widehat{a}(T_h, s_h) + \widehat{b}(\mathbf{u}_h, T_h, s_h) = (\gamma, s_h). \end{cases} \tag{2.10}$$

We have the following existence and uniqueness results [1].

Theorem 2.1. *Assume that the uniqueness condition* $0 < \sigma < 1$ *and discrete inf-sup condition* (2.8) *hold.*

(1) Assume the exact solution pair $(\mathbf{u}, p, T)$ *to the problem* (2.1) *belongs to* $H^2(\Omega)^2 \times H^1(\Omega) \times H^2(\Omega)$. *Then the approximated solution pair* $(\mathbf{u}_H, p_H, T_H) \in \mathbf{X}_H \times M_H \times W_H$ *of the problem* (2.9) *satisfies the following error estimates:*

$$\begin{aligned} &\|\nabla(\mathbf{u} - \mathbf{u}_H)\|_0 + \|p - p_H\|_0 + \|\nabla(T - T_H)\|_0 \leq cH(Pr\|\mathbf{u}\|_2 + \|p\|_1 + \kappa\|T\|_2), \\ &\|\mathbf{u} - \mathbf{u}_H\|_0 + \|T - T_H\|_0 \leq cH^2(Pr\|\mathbf{u}\|_2 + \|p\|_1 + \kappa\|T\|_2). \end{aligned}$$

(2) Assume the exact solution pair $(\mathbf{u}, p, T)$ *to the problem* (2.1) *belongs to* $H^3(\Omega)^2 \times H^2(\Omega) \times H^3(\Omega)$. *Then the approximated solution pair* $(\mathbf{u}_h, p_h, T_h) \in \mathbf{X}_h \times M_h \times W_h$ *to the problem* (2.10) *satisfies the following error estimates:*

$$\begin{aligned} &\|\nabla(\mathbf{u} - \mathbf{u}_h)\|_0 + \|p - p_h\|_0 + \|\nabla(T - T_h)\|_0 \leq c\,h^2(Pr\|\mathbf{u}\|_3 + \|p\|_2 + \kappa\|T\|_3), \\ &\|\mathbf{u} - \mathbf{u}_h\|_0 + \|T - T_h\|_0 \leq c\,h^3(Pr\|\mathbf{u}\|_3 + \|p\|_2 + \kappa\|T\|_3). \end{aligned}$$

3. The Hybrid Two-level and Parallel Two-level Method

3.1. Hybrid two-level method

In this subsection, we develop a hybrid two-level method to solve the stationary natural convection problem (Algorithm 3.1).

Importantly, the solution of a nonlinear system is restricted to the first step of the algorithm, which is performed efficiently by leveraging a coarse mesh and a low-order finite element pair. In the second step, a linearization method is adopted to avoid iteration.

Algorithm 3.1: Hybrid Two-level Algorithm.

Step I. Solve a nonlinear problem using a low order finite element pair on a coarse mesh: Find $(\mathbf{u}_H, p_H, T_H) \in \mathbf{X}_H \times M_H \times W_H$, such that for all $(\mathbf{v}_H, q_H, s_H) \in \mathbf{X}_H \times M_H \times W_H$,

$$\begin{cases} \mathbb{B}\big((\mathbf{u}_H, p_H), (\mathbf{v}_H, q_H)\big) + b(\mathbf{u}_H, \mathbf{u}_H, \mathbf{v}_H) = PrRa(\mathbf{j}T_H, \mathbf{v}_H), \\ \kappa\widehat{a}(T_H, s_H) + \widehat{b}(\mathbf{u}_H, T_H, s_H) = (\gamma, s_H). \end{cases}$$

Step II. Solve a linear problem using a high order finite element pair on a fine mesh: Find $(\mathbf{u}^h, p^h, T^h) \in \mathbf{X}_h \times M_h \times W_h$ such that for all $(\mathbf{v}_h, q_h, s_h) \in \mathbf{X}_h \times M_h \times W_h$,

$$\begin{cases} \mathbb{B}\big((\mathbf{u}^h, p^h), (\mathbf{v}_h, q_h)\big) + b\big(\mathbf{u}_H, \mathbf{u}^h, \mathbf{v}_h\big) = PrRa\big(\mathbf{j}T^h, \mathbf{v}_h\big), \\ \kappa\widehat{a}\big(T^h, s_h\big) + \widehat{b}\big(\mathbf{u}_H, T^h, s_h\big) + \widehat{b}\big(\mathbf{u}^h, T_H, s_h\big) - \widehat{b}(\mathbf{u}_H, T_H, s_h) = (\gamma, s_h). \end{cases} \tag{3.1}$$

Next, we will give the stability and error estimate of the proposed algorithm. To this end, let us recall the Galerkin projection operator in [19] $(R_h(\mathbf{u}, p), Q_h(\mathbf{u}, p)) : \mathbf{X} \times M \to \mathbf{X}_h \times M_h$ satisfying

$$\mathbb{B}\big(\big(R_h(\mathbf{u}, p), Q_h(\mathbf{u}, p)\big), (\mathbf{v}_h, q_h)\big) = \mathbb{B}\big((\mathbf{u}, p), (\mathbf{v}_h, q_h)\big), \quad \forall\, (\mathbf{v}_h, q_h) \in \mathbf{X}_h \times M_h.$$

And we have [3]

$$\begin{aligned} &Pr\|R_h(u,p) - u\|_0 + h\big(Pr\|\nabla(R_h(u,p) - u)\|_0 + \|Q_h(u,p) - p\|_0\big) \\ &\le c\, h^3(Pr\|u\|_3 + \|p\|_2), \end{aligned} \tag{3.2}$$

Furthermore, we introduce the elliptic projection $I_h T : W \to W_h$ as

$$\widehat{a}(I_h T, s_h) = \widehat{a}(T, s_h), \quad \forall\, s_h \in W_h,$$

which satisfies

$$\|I_h T - T\|_0 + h\|\nabla(I_h T - T)\|_0 \le c\, h^3 \|T\|_3. \tag{3.3}$$

The main theorem in this paper is stated as follows.

Theorem 3.1. *Under the conditions of Lemma* 2.1 *and Theorem* 2.1*, the solution* $(\mathbf{u}^h, p^h, T^h) \in \mathbf{X}_h \times M_h \times W_h$ *computed by Algorithm* 3.1 *satisfies*

$$\|\nabla \mathbf{u}^h\|_0 \le c_0^2 Ra\, \kappa^{-1} \|\gamma\|_0, \quad \|\nabla T^h\|_0 \le \kappa^{-1} \|\gamma\|_0. \tag{3.4}$$

Moreover, the following error estimates hold that

$$Pr\|\nabla(\mathbf{u} - \mathbf{u}^h)\|_0 + \|p - p^h\|_0 + \kappa\|\nabla(T - T^h)\|_0 \le c(h^2 + H^2).$$

Proof. Taking $(\mathbf{v}_h, q_h) = (\mathbf{u}^h, p^h)$ in (3.1) and using (2.2), we have

$$Pra(\mathbf{u}^h, \mathbf{u}^h) + b\big(\mathbf{u}_H, \mathbf{u}^h, \mathbf{u}^h\big) = PrRa(\mathbf{j}T^h, \mathbf{u}^h),$$

which, together with the antisymmetric of b and the coercivity of a, implies that

$$Pr\|\nabla \mathbf{u}^h\|_0^2 \le PrRa\|T^h\|_0 \|\mathbf{u}^h\|_0 \le c_0^2 PrRa\|\nabla T^h\|_0 \|\nabla \mathbf{u}^h\|_0, \tag{3.5}$$

where we have used the Poincare's inequality (2.5) in the last inequality.

Next taking $s_h = T^h$ in (3.1) and using the inequality (2.5), we have

$$\kappa\|\nabla T^h\|_0^2 \le \|\gamma\|_{-1}|T^h|_1 = \|\gamma\|_{-1}\|\nabla T^h\|_0,$$

which leads to the desired result (3.4) by combining with (3.5).

To obtain the error estimates, let $(\mathbf{v}, q, s) = (\mathbf{v}_h, q_h, s_h)$ in (2.1) and

$$(\mathbf{e}^h, \eta^h, \xi^h) = \big(R_h(\mathbf{u}, p) - \mathbf{u}^h, Q_h(\mathbf{u}, p) - p^h, I_h T - T^h\big).$$

Then subtracting (2.1) from (3.1) yields

$$\begin{cases} \quad \mathbb{B}\big((\mathbf{e}^h, \eta^h), (\mathbf{v}_h, q_h)\big) + b(\mathbf{u} - \mathbf{u}_H, \mathbf{u}, \mathbf{v}_h) + b\big(\mathbf{u}_H, \mathbf{u} - \mathbf{u}^h, \mathbf{v}_h\big) \\ = PrRa\big(\mathbf{j}\xi^h, \mathbf{v}_h\big) + PrRa\big(\mathbf{j}(T - I_h T), \mathbf{v}_h\big), \\ \quad \kappa\widehat{a}\big(\xi^h, s_h\big) + \widehat{b}(\mathbf{u} - \mathbf{u}_H, T, s_h) + \widehat{b}\big(\mathbf{u}_H, T - T^h, s_h\big) + \widehat{b}\big(\mathbf{u} - \mathbf{u}^h, T_H, s_h\big) \\ = \widehat{b}(\mathbf{u} - \mathbf{u}_H, T_H, s_h). \end{cases} \tag{3.6}$$

Next, taking $(\mathbf{v}_h, q_h, s_h) = (\mathbf{e}^h, \eta^h, \xi^h)$ in (3.6), we have

$$\begin{cases} \quad Pra(\mathbf{e}^h, \mathbf{e}^h) + b\big(\mathbf{u} - \mathbf{u}_H, \mathbf{u}, \mathbf{e}^h\big) + b\big(\mathbf{u}_H, \mathbf{u} - \mathbf{u}^h, \mathbf{e}^h\big) \\ = PrRa(\mathbf{j}\xi^h, \mathbf{e}^h) + PrRa\big(\mathbf{j}(T - I_h T), \mathbf{e}^h\big), \\ \quad \kappa\widehat{a}(\xi^h, \xi^h) + \widehat{b}\big(\mathbf{u} - \mathbf{u}_H, T, \xi^h\big) + \widehat{b}\big(\mathbf{u}_H, T - T^h, \xi^h\big) + \widehat{b}\big(\mathbf{u} - \mathbf{u}^h, T_H, \xi^h\big) \\ = \widehat{b}\big(\mathbf{u} - \mathbf{u}_H, T_H, \xi^h\big). \end{cases}$$

Using (2.3), (2.7), (3.2) and Theorem 2.1, we can get

$$\begin{aligned} \kappa\|\nabla\xi^h\|_0 &\le c_2\|\mathbf{u} - \mathbf{u}_H\|_0\|T\|_2 + N_2\|\nabla\mathbf{u}_H\|_0\|\nabla(T - I_h T)\|_0 \\ &\quad + N_2\|\nabla(\mathbf{u} - \mathbf{u}^h)\|_0\|\nabla T_H\|_0 + c_2\|\mathbf{u} - \mathbf{u}_H\|_0\|T_H\|_2 \\ &\le c\big(h^2 + H^2 + \|(\mathbf{u} - \mathbf{u}^h)\|_0\big) \le c\big(h^2 + H^2 + \|\nabla\mathbf{e}^h\|_0\big), \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} Pr\|\nabla\mathbf{e}^h\|_0 &\le c_1\|\mathbf{u} - \mathbf{u}_H\|_0\|\mathbf{u}\|_2 + N_1\big\|\nabla\big(\mathbf{u} - R_h(\mathbf{u}, p)\big)\big\|_0\|\nabla\mathbf{u}_H\|_0 \\ &\quad + c_0 PrRa\|\nabla\xi^h\|_0 + c_0 PrRa\|T - I_h T\|_0 \\ &\le c_1\|\mathbf{u} - \mathbf{u}_H\|_0\|\mathbf{u}\|_2 + N_1\big\|\nabla\big(\mathbf{u} - R_h(\mathbf{u}, p)\big)\big\|_0\|\nabla\mathbf{u}_H\|_0 \\ &\quad + c_0 PrRa\big(c\big(h^2 + H^2 + \|\mathbf{u} - \mathbf{u}^h\|_0\big)\big) + c_0 PrRa\|T - I_h T\|_0 \\ &\le c_1\|\mathbf{u} - \mathbf{u}_H\|_0\|\mathbf{u}\|_2 + N_1\big\|\nabla\big(\mathbf{u} - R_h(\mathbf{u}, p)\big)\big\|_0\|\nabla\mathbf{u}_H\|_0 \\ &\quad + c\left(h^2 + H^2 + \big\|\nabla\big(\mathbf{u} - R_h(\mathbf{u}, p)\big)\big\|_0 + \|\nabla\mathbf{e}^h\|_0\right) + c_0 PrRa\|T - I_h T\|_0. \end{aligned}$$

Thus, we have

$$\|\nabla\mathbf{e}^h\|_0 \le c(h^2 + H^2).$$

From the result in (3.7), it holds that

$$\|\nabla\xi^h\|_0 \le c(h^2 + H^2). \tag{3.8}$$

It follows from (3.6) and inf-sup condition (2.8) that

$$d\big(\mathbf{v}_h, \eta^h\big) = Pra\big(\mathbf{e}^h, \mathbf{v}_h\big) + b(\mathbf{u} - \mathbf{u}_H, \mathbf{u}, \mathbf{v}_h) + b\big(\mathbf{u}_H, \mathbf{u} - \mathbf{u}^h, \mathbf{v}_h\big) + PrRa\big(\mathbf{j}\xi^h, \mathbf{v}_h\big),$$

and

$$\begin{aligned}\beta_0\|\eta^h\|_0 &\le Pr\|\nabla \mathbf{e}^h\|_0 + c_1\|\mathbf{u}-\mathbf{u}_H\|_0\|u\|_2 \\ &\quad + N_1\|\nabla \mathbf{u}_H\|_0\|\nabla(\mathbf{u}-\mathbf{u}^h)\|_0 + c_0 PrRa\|\nabla \xi^h\|_0 \\ &\le c(h^2+H^2), \end{aligned} \tag{3.9}$$

where we have used the inequalities (2.2), (2.5), (2.6), (3.7) and (3.8). The proof is completed by combining (3.7)-(3.9) with the triangle inequality. □

Remark 3.1. This proposed algorithm can be considered as a combination of two-level method [6] and two-step method [25]. Though the error estimate depends on the coarse mesh size, the algorithm achieves second-order accuracy with only one correction step using high order finite element pairs. Moreover, the numerical results in Section 4 show that the relative error of our method is comparable to that of the one-level method which only uses $P_2 - P_1 - P_2$ finite element pair with the same fine mesh. Nevertheless, our method only requires solving a linearized problem on a fine mesh which can save a lot of computational cost.

Remark 3.2. In our numerical tests, we consider two distinct mesh relations: $H = \alpha h$ and $h = H^2$. The proposed algorithm performs effectively under both configurations and exhibits second-order convergence. However, it is noted that the error analysis presented in this paper is only applicable to the case of $H = \alpha h$. Establishing optimal error estimates for the relation $h = H^2$ remains an open problem and will be subject to our future investigation.

3.2. Parallel strategy based on fully overlapping domain decomposition

In this subsection, we combine the novel two-level method and fully overlapping domain decomposition technique to design a more efficient algorithm for solving the stationary incompressible natural convection problem.

In the parallel algorithm, each processor is assigned a coarse mesh with H and a composite mesh generated by a fine mesh with h. The fine mesh is then divided into k separate, non-overlapping subdomains $D_i, i = 1, 2, \ldots, k$. Next, each subdomain is extended to a larger domain Ω_i such that $D_i \subset\subset \Omega_i \subset \Omega$. The subdomain D_i is associated with a fine mesh size h within it, but has a larger mesh size $\overline{h}$ elsewhere. Then we can define the composite meshes as $T_i^{h,\overline{h}}(\Omega), i = 1, 2, \ldots, k$. Denote $X_i^{h,\overline{h}} \subset X_h, M_i^{h,\overline{h}} \subset M_h, W_i^{\nu,\overline{\nu}} \subset W_h, i = 1, \ldots, k$ on these composite meshes. These meshes are generated similarly to the technique described in [27], as shown in Fig. 3.1 for a case of $k = 2$.

The parallel two-level algorithm is described as Algorithm 3.2.

Defining piecewise norms

$$\begin{aligned}|||\nabla(\mathbf{u}-\mathbf{u}^h)|||_{0,\Omega} &= \left(\sum_{i=1}^{k}\|\nabla(\mathbf{u}-\mathbf{u}^{h,i})\|_{0,D_i}^2\right)^{\frac{1}{2}}, \\ |||p-p^h|||_{0,\Omega} &= \left(\sum_{i=1}^{k}\|p-p^{h,i}\|_{0,D_i}^2\right)^{\frac{1}{2}}, \\ |||\nabla(T-T^h)|||_{0,\Omega} &= \left(\sum_{i=1}^{k}\|\nabla(T-T^{h,i})\|_{0,D_i}^2\right)^{\frac{1}{2}}, \end{aligned}$$

we have the following convergence results.

Algorithm 3.2: The Parallel Two-level Algorithm.

(1) Assign a coarse mesh $T^H(\Omega)$ and a global composite mesh $T_i^{h,\overline{h}}(\Omega)$ in each processor $D_i, i = 1, \ldots, k$.

(2) Perform the following two-level procedure in parallel:

Step I. Find $(\mathbf{u}_H, p_H, T_H) \in \mathbf{X}_H \times M_H \times W_H$ such that for all $(\mathbf{v}_H, q_H, s_H) \in \mathbf{X}_H \times M_H \times W_H$,

$$\begin{cases} \mathbb{B}\big((\mathbf{u}_H, p_H), (\mathbf{v}_H, q_H)\big) + b(\mathbf{u}_H, \mathbf{u}_H, \mathbf{v}_h) = PrRa(\mathbf{j}T_H, \mathbf{v}_H), \\ \kappa\widehat{a}(T_H, s_H) + \widehat{b}(\mathbf{u}_H, T_H, s_H) = (\gamma, s_H). \end{cases}$$

Step II. Find $(\mathbf{u}^{h,i}, p^{h,i}, T^{h,i}) \in X_i^{h,\overline{h}} \times M_i^{h,\overline{h}} \times W_i^{h,\overline{h}}$ on $T_i^{h,\overline{h}}(\Omega)$ such that

$$\begin{cases} \mathbb{B}\big((\mathbf{u}^{h,i}, p^{h,i}), (\mathbf{v}, q)\big) + b\big(\mathbf{u}_{H,i}, \mathbf{u}^{h,i}, \mathbf{v}\big) = PrRa(\mathbf{j}T^{h,i}, \mathbf{v}), \\ \kappa\widehat{a}(T^{h,i}, s) + \widehat{b}\big(\mathbf{u}_{H,i}, T^{h,i}, s\big) + \widehat{b}\big(\mathbf{u}^{h,i}, T_{H,i}, s\big) - \widehat{b}(\mathbf{u}_{H,i}, T_{H,i}, s) \\ = (\gamma, s), \quad \forall\, (\mathbf{v}, q, s) \in X_i^{h,\overline{h}} \times M_i^{h,\overline{h}} \times W_i^{h,\overline{h}}. \end{cases}$$

(3) Set the global solution as $(\mathbf{u}^h, p^h, T^h) = (\mathbf{u}^{h,i}, p^{h,i}, T^{h,i})$ in $D_i, i = 1, 2, \ldots, k$.

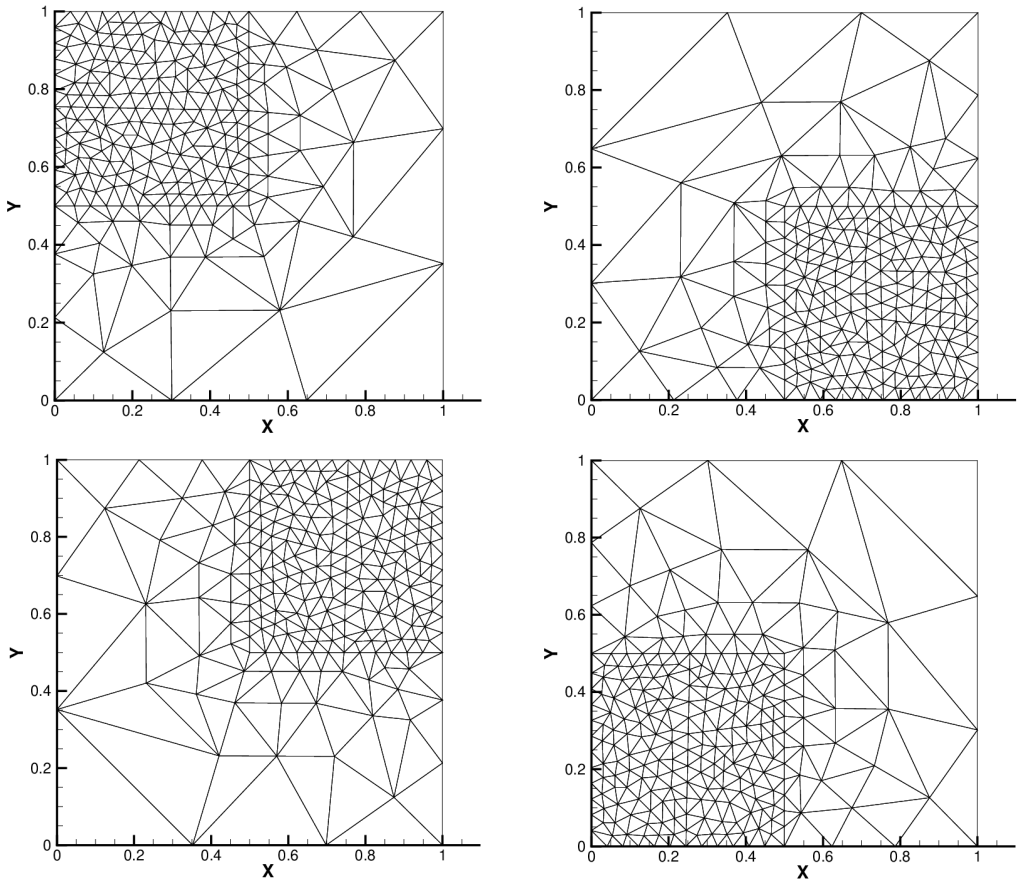

Fig. 3.1. A fully overlapping domain decomposition for a square domain with four subdomains.

Theorem 3.2. *Assume that $D_i \subset\subset \Omega_i \subset \Omega$ $(i = 1, 2, \ldots, k)$ and the solution pair $(\mathbf{u}^h, p^h, T^h)$ is computed by Algorithm 3.2, then under the conditions of Theorem 3.1, the following error estimate holds that:*

$$Pr|||\nabla(\mathbf{u} - \mathbf{u}^h)|||_{0,\Omega} + |||p - p^h|||_{0,\Omega} + \kappa|||\nabla(T - T^h)|||_{0,\Omega} \leq ck^{\frac{1}{2}}(h^2 + H^2).$$

Proof. Inspired by the ideas in [7,17], it follows from Theorem 3.1 that for each subdomain $D_i, i = 1, 2, \ldots, k$,

$$Pr\|\nabla(\mathbf{u} - \mathbf{u}^h)\|_{0,D_i} + \|p - p^h\|_{0,D_i} + \kappa\|\nabla(T - T^h)\|_{0,D_i} \leq c(h^2 + H^2).$$

Thus, we have

$$\begin{aligned} &Pr\left(\sum_{i=1}^{k}\|\nabla(\mathbf{u} - \mathbf{u}^{h,i})\|_{0,D_i}^2\right)^{\frac{1}{2}} + \left(\sum_{i=1}^{k}\|p - p^{h,i}\|_{0,D_i}^2\right)^{\frac{1}{2}} + \kappa\left(\sum_{i=1}^{k}\|\nabla(T - T^{h,i})\|_{0,D_i}^2\right)^{\frac{1}{2}} \\ &\leq ck^{\frac{1}{2}}(h^2 + H^2), \end{aligned}$$

which follows the desired results. □

Remark 3.3. In this paper, we only focus on the implement detail on the two dimensional problems. Extension to the three dimensional problems is straightforward, which remains as our future work.

4. Numerical Tests

In this section, we will perform some numerical tests to demonstrate the accuracy and the efficiency of our proposed methods. In Step 1, the Newton iteration is adopted and the stopping criterion is set to be 10^{-8}. As is known, Newton iteration exhibits a superior convergence rate compared to the Oseen iteration and Stokes iteration [11]. We define the relationship of H and h as $H = \alpha h$, where α is taken as a positive integer. All numerical tests are performed by using the public finite element software FreeFem++ [8].

4.1. Convergence test

We consider the natural convection problem (1.1) on a unit square $\Omega = [0, 1]^2$. The exact solution $(\mathbf{u}, p, T)$ is given by

$$\begin{aligned} &\mathbf{u} = \big(u_1(x, y), u_2(x, y)\big), \\ &u_1(x, y) = 10x^2(x - 1)^2y(y - 1)(2y - 1), \\ &u_2(x, y) = -10x(x - 1)(2x - 1)y^2(y - 1)^2, \\ &p(x, y) = 10(2x - 1)(2y - 1), \\ &T(x, y) = u_1(x, y) + u_2(x, y) \end{aligned}$$

with $Pr = 1, Ra = 10, \kappa = 1$ and the right hand sides is determined by the exact solution.

By setting the mesh size h=1/20,1/40,1/60 and 1/80 with scaling factors $\alpha = 2$ and $\alpha = 10$, we test the spatial convergence of Algorithm 3.1. The numerical errors and CPU times are reported in Tables 4.1 and 4.2, from which we observe that the convergence rate with respect

to h is of second order which is consistent with the expected theoretical results. Even with a coarser mesh ($H = 10h$) in Table 4.2, the numerical error is nearly identical to that of $H = 2h$ in Table 4.1, while the CPU time is noticeably reduced. These results indicate the robustness and efficiency of Algorithm 3.1 across different scaling factors α.

Furthermore, convergence tests are conducted using only the finite element pair P_2–P_1–P_2 with Newton iteration, serving as a benchmark for comparison (see Table 4.3). It is notable that the relative errors across Tables 4.1-4.3 are highly consistent and that optimal second-order convergence is attained in all instances.

To test the convergence of the parallel two-level method, we divide Ω into four disjoint rectangular subdomains

$$D_1 = \left(0, \frac{1}{2}\right) \times \left(0, \frac{1}{2}\right), \quad D_2 = \left(\frac{1}{2}, 1\right) \times \left(0, \frac{1}{2}\right),$$
$$D_3 = \left(\frac{1}{2}, 1\right) \times \left(0, \frac{1}{2}\right), \quad D_4 = \left(\frac{1}{2}, 1\right) \times \left(\frac{1}{2}, 1\right).$$

The finite element solutions on composite meshes are computed in parallel using mesh sizes $h = 1/(10 \times 2^i)$, $\overline{h} = 1/(5 \times 2^i), i = 1, 2, 3, 4$. In addition, we choose the two-level mesh with $H = 2h$. The resulting numerical errors and CPU times are presented in Table 4.4. From these tables, it can be observed that Algorithm 3.2 achieves a second-order convergence rate and requires less CPU time than that in Table 4.1 with the same mesh size.

Table 4.1: Numerical error and CPU time of Algorithm 3.1 with $\alpha = 2$ for the natural convection problem.

h	H	$\|\nabla(u-u_h)\|_0/\|\nabla u\|_0$	Rate	$\|p-p_h\|_0/\|p\|_0$	Rate	$\|\nabla(T-T_h)\|_0/\|\nabla T\|_0$	Rate	CPU(s)
1/20	1/10	1.097e-2	-	1.687e-3	-	5.267e-3	-	1.27
1/40	1/20	2.647e-3	2.08	4.188e-4	2.02	1.262e-3	2.09	5.25
1/80	1/40	6.797e-4	1.95	1.068e-4	1.96	3.141e-4	2.01	20.95
1/160	1/80	1.657e-4	1.97	2.614e-5	2.04	7.843e-5	2.00	92.77

Table 4.2: Numerical error and CPU time of Algorithm 3.1 with $\alpha = 10$ for the natural convection problem.

h	H	$\|\nabla(u-u_h)\|_0/\|\nabla u\|_0$	Rate	$\|p-p_h\|_0/\|p\|_0$	Rate	$\|\nabla(T-T_h)\|_0/\|\nabla T\|_0$	Rate	CPU(s)
1/20	1/2	1.099e-2	-	1.717e-3	-	6.102e-3	-	0.78
1/40	1/4	2.652e-3	2.05	4.334e-4	1.99	1.653e-3	1.89	3.15
1/80	1/8	6.803e-4	1.96	1.094e-4	1.99	3.869e-4	2.10	13.66
1/160	1/16	1.659e-4	2.03	2.694e-5	2.02	1.000e-4	1.95	66.43

Table 4.3: Numerical error and CPU time of $P_2 - P_1 - P_2$ finite element pair for the natural convection problem.

h	$\|\nabla(u-u_h)\|_0/\|\nabla u\|_0$	Rate	$\|p-p_h\|_0/\|p\|_0$	Rate	$\|\nabla(T-T_h)\|_0/\|\nabla T\|_0$	Rate	CPU(s)
1/20	1.097e-2	-	1.687e-3	-	5.265e-3	-	2.90
1/40	2.647e-3	2.08	4.188e-4	2.02	1.262e-3	2.03	11.78
1/80	6.797e-4	1.95	1.068e-4	1.96	3.140e-4	2.01	48.67
1/160	1.657e-4	1.97	2.614e-5	2.04	7.840e-5	2.00	247.71

Table 4.4: Numerical error and CPU time of Algorithm 3.2 with $\alpha = 2$ for the natural convection problem.

h	$\overline{h}$	$\|\nabla(u-u_h)\|_0/\|\nabla u\|_0$	Rate	$\|p-p_h\|_0/\|p\|_0$	Rate	$\|\nabla(T-T_h)\|_0/\|\nabla T\|_0$	Rate	CPU(s)
1/20	1/10	6.378e-3	-	8.869e-4	-	3.080e-3	-	0.66
1/40	1/20	1.402e-3	2.19	2.135e-4	2.06	6.933e-4	2.15	2.76
1/80	1/40	3.630e-4	1.95	5.423e-5	1.98	1.780e-4	1.96	11.52
1/160	1/80	8.912e-5	1.97	1.354e-5	2.00	4.514e-5	1.98	54.26

Table 4.5: A comparison of two algorithms with $h = H^2$ for the natural convection problem.

Method	h	H	$\|\nabla(u-u_h)\|_0/\|\nabla u\|_0$	Rate	$\|p-p_h\|_0/\|p\|_0$	Rate	$\|\nabla(T-T_h)\|_0/\|\nabla T\|_0$	Rate
	1/4	1/2	3.05e-1	-	1.65e-1	-	5.02e-2	-
Hybrid	1/16	1/4	1.61e-2	2.12	8.28e-3	2.16	2.57e-3	2.14
two-level	1/36	1/6	3.36e-3	1.93	1.60e-3	2.03	5.22e-4	1.97
method	1/64	1/8	1.12e-3	1.91	5.61e-4	1.83	1.72e-4	1.93
	1/100	1/10	3.77e-4	2.40	2.48e-4	1.83	6.74e-5	2.10
	1/4	1/2	3.05e-1	-	1.65e-1	-	5.02e-2	-
Two-level	1/16	1/4	1.61e-2	2.12	8.21e-3	2.16	2.56e-3	2.14
method [6]	1/36	1/6	3.36e-3	1.93	1.55e-3	2.06	5.20e-4	1.96
	1/64	1/8	1.12e-3	1.91	5.09e-4	1.93	1.70e-4	1.94
	1/100	1/10	3.77e-4	2.40	1.91e-4	2.19	6.54e-5	2.14

To compare with the classical two-level method [6], we set $h = H^2$. A numerical comparison of these two methods is presented in Table 4.5. The numerical results produced by our method are comparable to those of the traditional two-level method, demonstrating that our method performs effectively with $h = H^2$.

4.2. The thermal-driven cavity flow problem

In this test, we consider the benchmark thermal-driven cavity flow problem [15,20,24] on the unit square domain $\Omega = [0,1]^2$ without the forcing term. The boundary condition is illustrated in Fig. 4.1. The parameter settings is Pr=0.71 and κ=1.0.

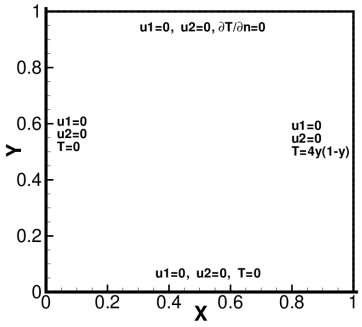

Fig. 4.1. Configuration of square driven flow for the natural convection problem.

We first get the approximate solutions using Algorithm 3.1 on non-structured mesh with $H = 1/30$ and $h = 1/60$. The numerical streamline, temperature fields and pressure with the different Rayleigh numbers Ra=10^3,10^4 and 10^5 are drawn in Fig. 4.2, respectively. We also report the profiles with the high Rayleigh numbers Ra=10^5 equipped with $H = 1/8$ and $h = 1/64$ in Fig. 4.3. The isotherms of pressure exhibit an approximately linear distribution with vertical contours, consistent with the results reported in [15, 20]. These results confirm the robustness of the proposed algorithm even for problems with low regularity under the mesh relation $h = H^2$.

Then, we apply Algorithm 3.2 to solve the thermal-driven cavity flow problem on non-structured mesh with $\overline{H} = 1/15$, $H = 1/30$, $\overline{h} = 1/30$ and $h = 1/60$. To verify the efficiency of the Algorithm 3.2, Rayleigh number is taken as $Ra = 10^3$. We divide the domain Ω into four disjoint rectangular subdomains

$$D_1 = \left(0, \frac{1}{2}\right) \times \left(0, \frac{1}{2}\right), \quad D_2 = \left(\frac{1}{2}, 1\right) \times \left(0, \frac{1}{2}\right),$$
$$D_3 = \left(\frac{1}{2}, 1\right) \times \left(0, \frac{1}{2}\right), \quad D_4 = \left(\frac{1}{2}, 1\right) \times \left(\frac{1}{2}, 1\right).$$

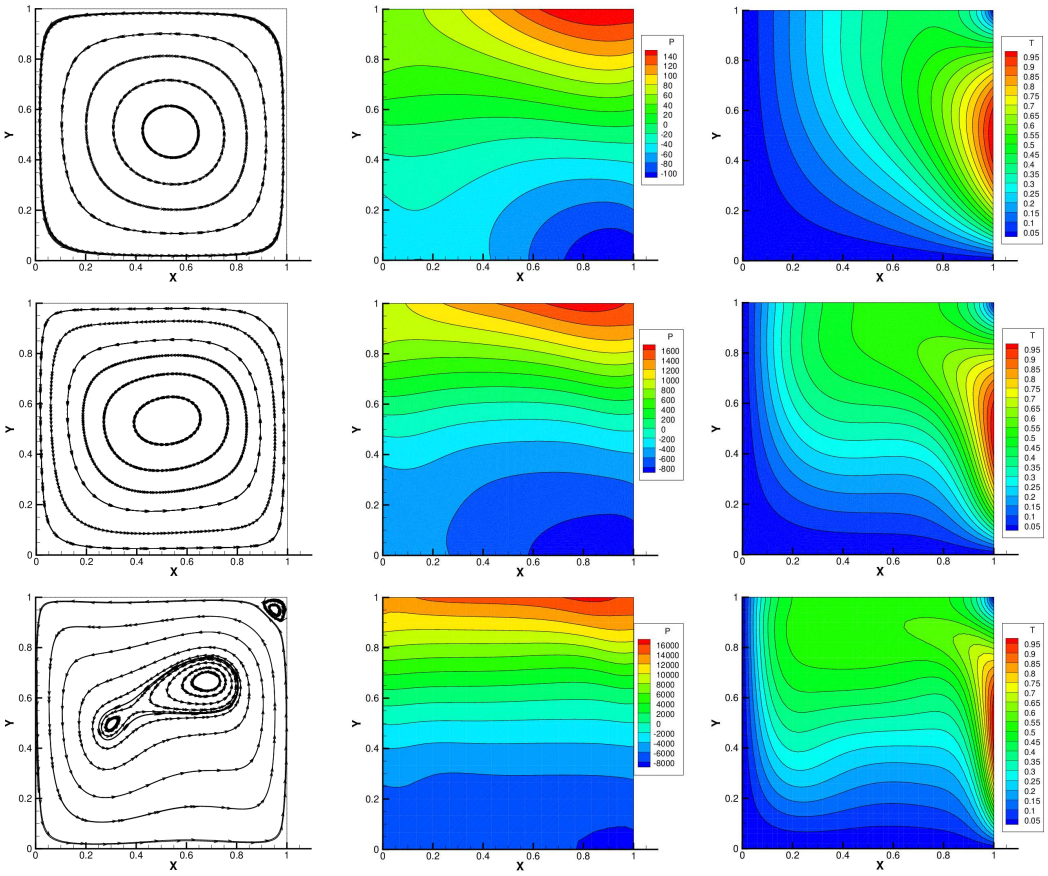

Fig. 4.2. The numerical streamlines (left), isobars (middle), isotherms (right) computed by Algorithm 3.1 with $H = 1/30$ and $h = 1/60$ for the thermal driven square flow with Rayleigh numbers Ra=10^3,10^4 and 10^5 (from top to bottom).

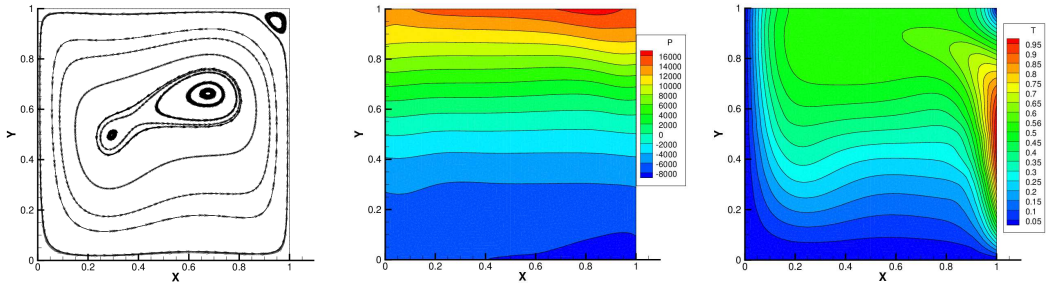

Fig. 4.3. The numerical streamlines (left), isobars (middle), isotherms (right) computed by Algorithm 3.1 with $H = 1/8$ and $h = 1/64$ for the thermal driven square flow with Rayleigh numbers $Ra = 10^5$.

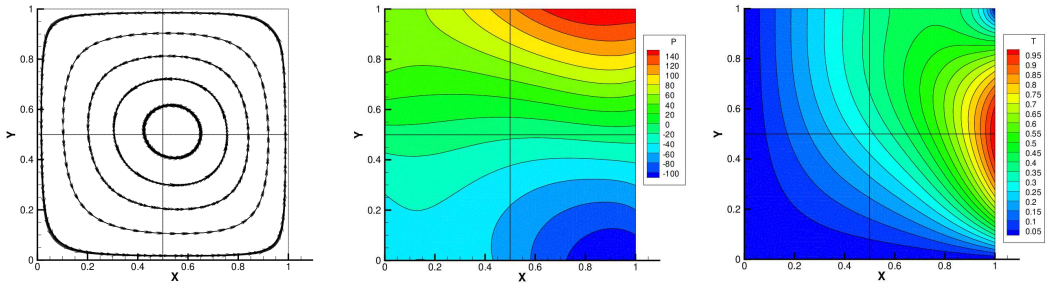

Fig. 4.4. The numerical streamlines (left), isobars (middle), isotherms (right) of Algorithm 3.2 with Rayleigh number $Ra = 10^3$ for the thermal driven square flow.

The corresponding numerical results of velocity streamline, pressure and temperature are plotted in Fig. 4.4. It can be seen that Algorithm 3.2 for thermal driven cavity flow problem is compatible to Algorithm 3.1, while requiring less CPU time due to the reduced number of freedom of degree.

4.3. The two obstacles flow

In this test, we consider a flow configuration with two obstacles located at the center of the square domain. The corresponding boundary conditions are shown in Fig. 4.5.

Fig. 4.6 presents the numerical results by Algorithm 3.1 on a mesh of 3168 triangle elements, with the CPU time of 18.084 seconds. For Algorithm 3.2, the domain Ω is divided into two

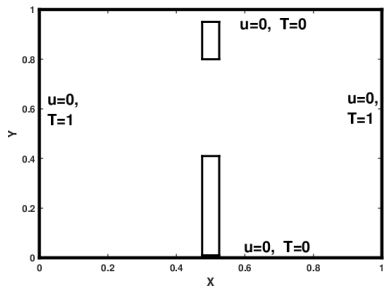

Fig. 4.5. Configuration of two obstacles problem.

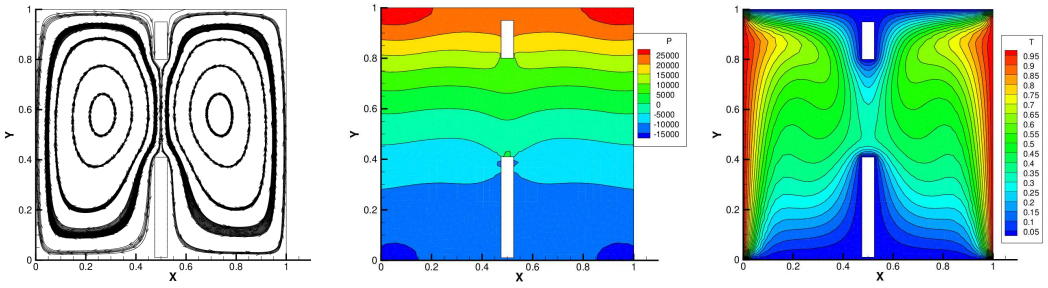

Fig. 4.6. The numerical streamlines (left), isobars (middle) and isotherms (right) of Algorithm 3.1 for the two obstacles problem.

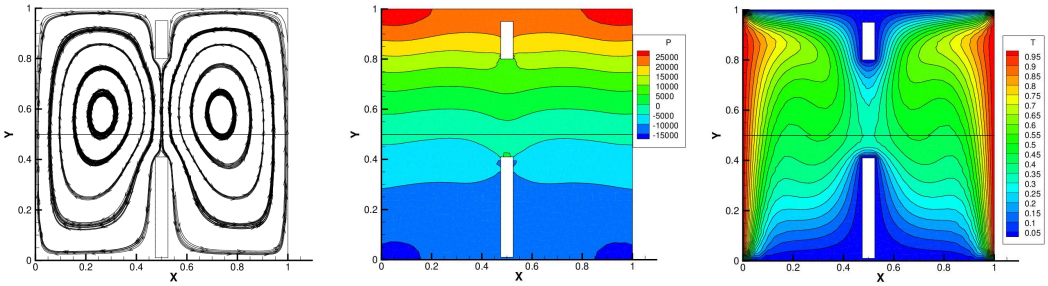

Fig. 4.7. The numerical streamlines (left), isobars (middle) and isotherms (right) of Algorithm 3.2 for the two-obstacle problem.

disjoint rectangular subdomains

$$D_1 = (0,1) \times \left(0, \frac{1}{2}\right), \quad D_2 = (0,1) \times \left(\frac{1}{2}, 1\right),$$

and solved using fully overlapping domain decomposition method on a mesh of 3122 triangle elements. The corresponding numerical results of streamlines, isobars, isotherms are plotted in Fig. 4.7, with the CPU time of 11.089 seconds. It can be seen that both algorithms produces the expected symmetric flow due to the symmetric boundary condition and Algorithm 3.2 takes less CPU time than Algorithm 3.1.

4.4. Bénard convection problem

We consider the Bénard convection problem on the domain $\Omega = [0,5] \times [0,1]$ without the external forcing. Fig. 4.8 displays the boundary conditions for velocity and temperature. The parameter settings are $Pr = 1, \kappa = 1$ and $Ra = 4000$.

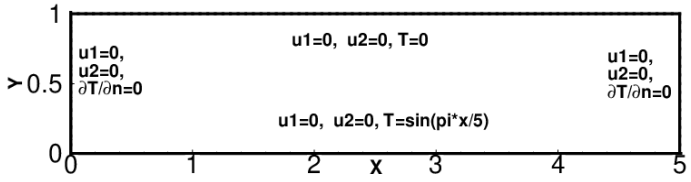

Fig. 4.8. Configuration of natural convection problem.

Fig. 4.9 shows the numerical streamline, temperature and pressure computed by Algorithm 3.1 with mesh size $H = 1/60$ and $h = 1/120$. The corresponding computational time is 45.69 seconds. Fig. 4.10 presents the numerical streamline, temperature and pressure computed by Algorithm 3.2 where the domain is divided into five disjoint subdomains

$$D_1 = (0,1) \times (0,1), \quad D_2 = (1,2) \times (0,1), \quad D_3 = (2,3) \times (0,1),$$
$$D_4 = (3,4) \times (0,1), \quad D_5 = (4,5) \times (0,1)$$

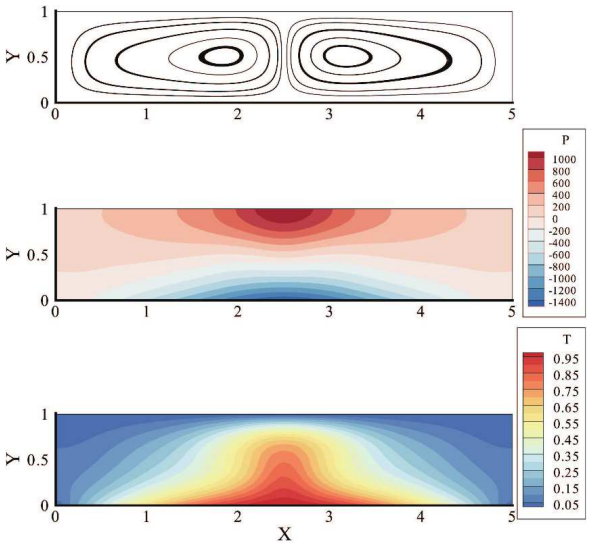

Fig. 4.9. The numerical streamlines (top), isobars (middle) and isotherms (bottom) of Algorithm 3.1 for the Bénard convection problem.

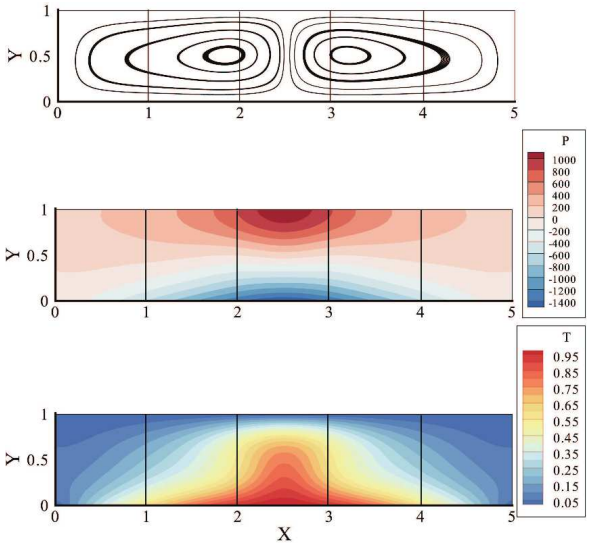

Fig. 4.10. The numerical streamlines (top), isobars (middle) and isotherms (bottom) of Algorithm 3.2 for the Bénard convection problem.

with mesh size $\overline{H} = 1/30, H = 1/60, \overline{h} = 1/60$ and $h = 1/120$. The corresponding computational time is 34.21 seconds. It can be observed that the Algorithm 3.2 is advisable and takes the less computational time.

5. Conclusions

In this paper, we propose and analyze the hybrid two-level and parallel two-level algorithms for the stationary natural convection problem. The proposed two-level algorithms are efficient and achieve the optimal second-order accuracy. The parallel strategy based on the fully overlapping domain decomposition can dramatically improve the computational efficiency due to the less degree of freedom. A variety of numerical results are performed to demonstrate that the relative error of the novel two-level algorithm is comparable to the algorithm of only using $P_2 - P_1 - P_2$, even though we do not obtain the optimal error estimate. The extension to more general complex fluid problem will be subject to our future investigation.

Acknowledgements. We sincerely thank the anonymous reviewers for their insightful comments, which have helped improve the quality of this paper.

G. Zhang's work was supported in part by the Guangxi Natural Science Foundation (Grant No. 2026GXNSFBA00640283) and Tianyuan Fund for Mathematics of the National Natural Science Foundation of China (Grant No. 12526545). H. Chen's work was partially supported by the Natural Science Foundation of Fujian Province of China (Grant No. 2025J01026), by the National Key R&D Program of China (Grant Nos. 2022YFA1004500 and 2024YFA1012503), by the National Natural Science Foundation of China (Grant No. 12371372). J. Li's work was partially supported by the Talent Scientific Fund of Lanzhou University and by the National Natural Science Foundation of China (Grant Nos. 12301510, 12171216 and 12526514).

References

[1] J. Boland and W. Layton, Error analysis for finite element methods for steady natural convection problems, *Numer. Funct. Anal. Optim.*, **11** (1990), 449–483.

[2] Y. Cai and J. Shen, Error estimates for a fully discretized scheme to a Cahn-Hilliard phase-field model for two-phase incompressible flows, *Math. Comp.*, **87** (2018), 2057–2090.

[3] V. Ervin, W. Layton, and J. Maubach, A posteriori error estimators for a two-level finite element method for the Navier-Stokes equations, *Numer. Methods Partial Differential Equations*, **12** (1996), 333–346.

[4] X. Feng and Y. He, The convergence of a new parallel algorithm for the Navier-Stokes equations, *Nonlinear Anal. Real World Appl.*, **10** (2009), 23–41.

[5] Y. He, A two-level finite element Galerkin method for the nonstationary Navier-Stokes equations, I: Spatial discretization, *J. Comput. Math.*, **22** (2004), 21–32.

[6] Y. He and A. Wang, A simplified two-level method for the steady Navier-Stokes equations, *Comput. Methods Appl. Mech. Engrg.*, **197** (2008), 1568–1576.

[7] Y. He, J. Xu, and A. Zhou, Local and parallel finite element algorithms for the Navier-Stokes problem, *J. Comput. Math.*, **24** (2006), 227–238.

[8] F. Hecht, New development in FreeFem++, *J. Numer. Math.*, **20** (2012), 251–265.

[9] P. Huang, An efficient two-level finite element algorithm for the natural convection equations, *Appl. Numer. Math.*, **118** (2017), 75–86.

[10] P. Huang, X. Feng, and Y. He, An efficient two-step algorithm for the incompressible flow problem, *Adv. Comput. Math.*, **41** (2015), 1059–1077.

[11] P. Huang, W. Li, and Z. Si, Several iterative schemes for the stationary natural convection equations at different Rayleigh numbers, *Numer. Methods Partial Differential Equations*, **31** (2015), 761–776.

[12] W. Layton, A two-level discretization method for the Navier-Stokes equations, *Comput. Math. Appl.*, **26** (1993), 33–38.

[13] H. Liang and T. Zhang, Parallel two-grid finite element method for the time-dependent natural convection problem with non-smooth initial data, *Comput. Math. Appl.*, **77** (2019), 2221–2241.

[14] Z. Luo, J. Zhu, Z. Xie, and G. Xu, Difference scheme and numerical simulation based on mixed finite element method for natural convection problem, *Appl. Math. Mech.*, **9** (2003), 1100–1110.

[15] R. Nochetto and J. Pyo, The finite element Gauge-Uzawa method. Part II: The Boussinesq equations, *Math. Models Methods Appl. Sci.*, **16** (2006), 1599–1626.

[16] Y. Shang, Y. He, and X. Feng, A new parallel finite element algorithm based on two-grid discritization for the generalized Stokes problem, *Int. J. Numer. Anal. Model.*, **13** (2016), 676–688.

[17] Y. Shang, Y. He, D. Kim, and X. Zhou, A new parallel finite element algorithm for the stationary Navier-Stokes equations, *Finite Elem. Anal. Des.*, **47** (2011), 1262–1279.

[18] M. Sheikholeslami, R. Ellahi, and M. Hassan, A study of natural convection heat transfer in a nanofluid filled enclosure with elliptic inner cylinder, *Int. J. Numer. Methods Heat Fluid Flow*, **24** (2014), 1906–1927.

[19] Z. Si, X. Song, and P. Huang, Modified characteristics Gauge-Uzawa finite element method for time dependent conduction-convection problems, *J. Sci. Comput.*, **58** (2014), 1–24.

[20] Z. Si, T. Zhang, K. Wang, A Newton iterative scheme mixed finite element method for stationary conduction-convection problems, *Int. J. Comput. Fluid Dyn.*, **24** (2010), 135–141.

[21] H. Su, J. Zhao, D. Gui, and X. Feng, Two-level defect-correction Oseen iterative stabilized finite element method for the stationary conduction-convection equations, *Int. Commun. Heat Mass Transfer*, **56** (2014), 133–145.

[22] S. Voelker, T. Burton, and S.P. Vanka, Finite-volume multigrid calculation of natural-convection flows on unstructured grids, *Numer. Heat Transfer, Part B*, **30** (1996), 1–22.

[23] W. Wang and G. Zhang, A hybrid two-grid algorithm for the steady magnetohydrodynamic system, *J. Sci. Comput.*, **101** (2024), 64.

[24] J. Wu, P. Huang, and X. Feng, A new variational multiscale FEM for the steady-state natural convection problem with bubble stabilization, *Numer. Heat Transfer, Part A*, **68** (2015), 777–796.

[25] J. Wu, P. Huang, X. Feng, and D. Liu, An efficient two-step algorithm for steady-state natural convection problem, *Int. J. Heat Mass Transfer*, **101** (2016), 387–398.

[26] J. Xu, Two-grid discretization techniques for linear and nonlinear PDEs, *SIAM J. Numer. Anal.*, **33** (1996), 1759–1777.

[27] G. Zhang, H. Su, and X. Feng, A novel parallel two-step algorithm based on finite element discretization for the incompressible flow problem, *Numer. Heat Transfer, Part B*, **73** (2018), 329–341.

[28] T. Zhang, Y. Hou, and H. Zheng, A finite element variational multiscale method for steady-state natural convection problem based on two local Gauss integrations, *Numer. Methods Partial Differential Equations*, **30** (2014), 361–375.

[29] T. Zhang, X. Zhao, and P. Huang, Decoupled two level finite element methods for the steady natural convection problem, *Numer. Algorithms*, **68** (2015), 837–866.

[30] B. Zheng and Y. Shang, Parallel iterative stabilized finite element algorithms based on the lowest equal-order elements for the stationary Navier-Stokes equations, *Appl. Math. Comput.*, **357** (2019), 35–56.

[31] B. Zheng and Y. Shang, A parallel stabilized finite element variational multiscale method based on fully overlapping domain decomposition for the incompressible Navier-Stokes equations, *Appl. Numer. Math.*, **159** (2020), 138–158.