



Interpolation-supplemented lattice Boltzmann simulation of thermal convection on non-uniform meshes

Ao Xu ^{a,b}, , Zheng Zhao ^a, Ben-Rui Xu ^a, Li-Sheng Jiang ^a

^a Institute of Extreme Mechanics, School of Aeronautics, Northwestern Polytechnical University, Xi'an, 710072, PR China

^b National Key Laboratory of Aircraft Configuration Design, Key Laboratory for Extreme Mechanics of Aircraft of Ministry of Industry and Information Technology, Xi'an, 710072, PR China

ARTICLE INFO

Keywords:

Lattice Boltzmann method
Thermal convection
High-performance computing

ABSTRACT

We present a systematic evaluation of an interpolation-supplemented lattice Boltzmann method (ISLBM) for simulating buoyancy-driven thermal convection on non-uniform meshes. The ISLBM extends the standard lattice Boltzmann framework by incorporating quadratic interpolation during the streaming step, enabling flexible mesh refinement near solid boundaries while maintaining algorithmic simplicity and parallel scalability. The method is implemented for a two-dimensional side-heated cavity at high Rayleigh numbers $10^6 \leq Ra \leq 10^8$, and for a three-dimensional side-heated cavity at $10^5 \leq Ra \leq 10^7$, with the Prandtl number fixed at $Pr = 0.71$. Benchmark results show that the ISLBM accurately captures thermal and velocity boundary layers, yielding Nusselt and Reynolds numbers in close agreement with high-fidelity reference data. Grid-convergence studies demonstrate nearly third-order accuracy for global quantities and about second-order for local fields. We further assess the computational performance of the in-house LBM solver against two open-source solvers: Nek5000 based on the spectral element method, and OpenFOAM based on the finite volume method. Performance metrics, including million lattice updates per second (MLUPS) and wall-clock time per dimensionless time unit (WCTpDT), indicate that the ISLBM offers one to three orders of magnitude higher efficiency in large-scale simulations. On GPU architectures, the ISLBM retains high computational performance: throughput on non-uniform meshes reaches 60%–70% of that on uniform meshes in terms of MLUPS, while the cost in WCTpDT is about three times higher. These results highlight the potential of interpolation-based LBM approaches for high-fidelity simulations of thermal convection on non-uniform meshes, providing a robust foundation for future extensions to turbulent flows.

1. Introduction

The lattice Boltzmann method (LBM) is a mesoscopic numerical approach to computational fluid dynamics [1,2], rooted in the Boltzmann kinetic theory [3,4]. Instead of directly solving the macroscopic Navier–Stokes equations, the LBM models the evolution of particle distribution functions on a discrete lattice [5,6]. From these evolution equations, macroscopic flow variables such as velocity and pressure are recovered. This framework provides access to continuum-scale hydrodynamics while retaining mesoscopic fidelity. Early developments of the LBM were inspired by lattice gas automata, but a rigorous connection to the Boltzmann equation was later established by He and Luo [7], who derived the lattice Boltzmann equation from the continuous Bhatnagar–Gross–Krook formulation. Over the past three decades, the LBM has matured into an efficient tool for simulating a wide range of complex flow phenomena. Its inherent algorithmic simplicity, local collision-streaming structure, and natural suitability for parallel

computation [8–10] make it particularly effective for simulating fluid flows and associated transport processes. As a result, the LBM has achieved broad success across diverse applications, including turbulent and multiphase flow simulations [11–19]. This success has positioned LBM as a compelling alternative to conventional Navier–Stokes-based solvers, particularly for incompressible flows in the continuum regime.

Our particular interest lies in using the LBM to simulate thermal convection, specifically the coupled fluid flow and heat transfer processes characteristic of buoyancy-driven systems [20–22]. This fundamental mechanism underpins a wide range of natural and engineered systems, including atmospheric and oceanic convection [22], thermal management in fuel cells and flow batteries [23], thermal protection in nuclear reactors [24], and thermal convection in mechanical devices subjected to vibrations [25]. Accurately simulating thermal convection at high Rayleigh numbers (Ra , defined later) remains a major computational challenge, as strong buoyancy forces significantly influence

* Corresponding author at: Institute of Extreme Mechanics, School of Aeronautics, Northwestern Polytechnical University, Xi'an, 710072, PR China.
E-mail address: axu@nwpu.edu.cn (A. Xu).

the flow and heat transfer dynamics. The difficulty arises from the formation of thin thermal boundary layers, which demand much finer resolution near solid boundaries than in the bulk flow. As a result, uniform meshes quickly become computationally prohibitive, since they enforce fine resolution across the entire domain. To overcome this limitation, it is necessary to extend the conventional LBM (typically formulated on uniform meshes) to non-uniform meshes refined near solid boundaries, where steep velocity and temperature gradients occur. Such spatially adaptive meshing enables adequate resolution of near-wall structures, such as thermal boundary layers, while preserving computational efficiency in the bulk flow region.

Several extensions of the standard LBM from uniform meshes to non-uniform meshes have been developed to resolve near-wall gradients without incurring excessive computational cost. A prominent example is the interpolation-supplemented LBM (ISLBM) [26–28], which incorporates interpolation schemes during the streaming step to enable off-lattice population transfers, thereby allowing non-uniform lattice spacing while retaining the simplicity of the standard formulation. Another widely studied approach is the finite-volume LBM [29–32], which reformulates the lattice Boltzmann equation within a control-volume framework, permitting flexible mesh structures while preserving its kinetic foundation. In addition, mesh refinement strategies, such as block-structured multi-domain methods [33,34] and multilevel (multigrid) methods [35,36], have been extensively employed to resolve localized flow features at varying spatial resolutions. These methodological advances are particularly relevant as the LBM is increasingly applied to simulate thermal convection at high Rayleigh numbers, where both large-scale circulations and thin boundary layers must be resolved. Despite their potential, the performance of these methods in terms of numerical accuracy, stability, and computational efficiency for thermally driven flows at high Rayleigh numbers have not been systematically assessed. Therefore, a comprehensive evaluation of their accuracy, numerical stability, and efficiency is essential for advancing high-fidelity LBM simulations of buoyancy-driven flows.

Due to its conceptual simplicity and its direct extension of the standard streaming step, we focus here on evaluating the ISLBM. Specifically, we implement and assess the ISLBM for simulating buoyancy-driven thermal convection at moderately high Rayleigh numbers on non-uniform meshes. Through systematic benchmarks, we aim to quantify its accuracy and efficiency, thereby laying the groundwork for its future application to turbulent thermal systems. The rest of this paper is organized as follows. In Section 2, we present the numerical details of the ISLBM for simulating thermal convection on non-uniform meshes. In Sections 3 and 4, we present results on laminar thermal convection in a two-dimensional (2-D) and a three-dimensional (3-D) side-heated cavity, respectively. In Section 5, the main findings of the present work are summarized.

2. Numerical methods

2.1. The standard LBM for simulating thermal convection on uniform meshes

We perform direct numerical simulations of thermal convection under the Boussinesq approximation. The fluid flow is assumed to be incompressible, and temperature is treated as an active scalar that influences the velocity field through buoyancy. Viscous dissipation and compression work are neglected, and all transport coefficients are assumed constant. The governing equations for the coupled fluid flow and heat transfer are

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0} \nabla P + \nu \nabla^2 \mathbf{u} + g\beta (T - T_0) \hat{\mathbf{y}} \quad (2)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha \nabla^2 T \quad (3)$$

where \mathbf{u} , P and T denote the velocity, pressure, and temperature fields, respectively. The reference density and temperature are denoted by ρ_0 and T_0 . ν , β and α are the kinematic viscosity, thermal expansion coefficient, and thermal diffusivity of the fluid, respectively. The unit vector $\hat{\mathbf{y}}$ points in the direction of gravity. To non-dimensionalize the equations, we apply the following scalings:

$$\mathbf{x}^* = \mathbf{x}/H, \quad t^* = t/\sqrt{H/(\beta g \Delta_T)}, \quad \mathbf{u}^* = \mathbf{u}/\sqrt{\beta g H \Delta_T} \quad (4)$$

$$P^* = P/(\rho_0 g \beta \Delta_T H), \quad T^* = (T - T_0)/\Delta_T$$

where Δ_T is the temperature difference between the hot and cold walls. The characteristic length scale is the cavity height H , the characteristic time scale is the free-fall time $t_f = \sqrt{H/(\beta g \Delta_T)}$, and the characteristic velocity scale is the free-fall velocity $u_f = \sqrt{H \beta g \Delta_T}$. Unless otherwise stated, dimensionless variables are denoted with a superscript star.

Using these scalings, the dimensionless governing equations become

$$\nabla \cdot \mathbf{u}^* = 0 \quad (5)$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla \mathbf{u}^* = -\nabla P^* + \sqrt{\frac{Pr}{Ra}} \nabla^2 \mathbf{u}^* + T^* \hat{\mathbf{y}} \quad (6)$$

$$\frac{\partial T^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla T^* = \sqrt{\frac{1}{Pr Ra}} \nabla^2 T^* \quad (7)$$

Two dimensionless parameters arise in the system: the Rayleigh number (Ra) and the Prandtl number (Pr), defined as

$$Ra = \frac{g \beta \Delta_T H^3}{\nu \alpha}, \quad Pr = \frac{\nu}{\alpha} \quad (8)$$

The Rayleigh number quantifies the driving buoyancy force relative to viscous and thermal diffusion, while the Prandtl number represents the ratio of momentum diffusivity to thermal diffusivity.

To solve the thermal convection problem described above, we adopt a double-distribution-function (DDF) LB model [8,37–41]. Specifically, a D2Q9 lattice in two dimensions or a D3Q19 lattice in three dimensions is employed to solve the Navier–Stokes equations for fluid flow, while a D2Q5 lattice in two dimensions or a D3Q7 lattice in three dimensions is used to solve the energy equation for heat transfer. To enhance numerical stability, a multi-relaxation-time (MRT) collision operator is employed for both the density and temperature distribution functions. The evolution equation for the density distribution function is

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) - f_i(\mathbf{x}, t) = -(\mathbf{M}^{-1} \mathbf{S})_{ij} \left[\mathbf{m}_j(\mathbf{x}, t) - \mathbf{m}_j^{(eq)}(\mathbf{x}, t) \right] + \delta_t F'_i \quad (9)$$

$(i = 0, 1, \dots, q-1)$

where f_i is the density distribution function, \mathbf{x} is the spatial position, t is the time, and δ_t is the time step. \mathbf{e}_i is the discrete velocity vector in the i th direction. For the D2Q9 lattice, $q = 9$; for the D3Q19 lattice, $q = 19$. The equilibrium moments $\mathbf{m}^{(eq)}$ are given by, for the D2Q9 lattice

$$\mathbf{m}_{D2Q9}^{(eq)} = \rho [1, -2+3(u^2+v^2), 1-3(u^2+v^2), u, -u, v, -v, u^2-v^2, uv]^T \quad (10)$$

and for the D3Q19 lattice,

$$\mathbf{m}_{D3Q19}^{(eq)} = \rho \left[1, -11+19|\mathbf{u}|^2, 3-\frac{11}{2}|\mathbf{u}|^2, u, -\frac{2}{3}u, v, -\frac{2}{3}v, w, -\frac{2}{3}w, 2u^2-v^2-w^2, -\frac{1}{2}(2u^2-v^2-w^2), v^2-w^2, -\frac{1}{2}(v^2-w^2), uv, vw, uw, 0, 0, 0 \right]^T \quad (11)$$

The forcing term F'_i on the right-hand side of Eq. (9) is computed as $\mathbf{F}' = \mathbf{M}^{-1}(\mathbf{I} - \mathbf{S}/2)\mathbf{M}\tilde{\mathbf{F}}$, where $\mathbf{M}\tilde{\mathbf{F}}$ is given by Guo et al. [42,43]. For the D2Q9 lattice,

$$\mathbf{M}\tilde{\mathbf{F}}_{D2Q9} = [0, 6\mathbf{u} \cdot \mathbf{F}, -6\mathbf{u} \cdot \mathbf{F}, F_x, -F_x, F_y, -F_y, 2uF_x - 2vF_y, uF_x + vF_y]^T \quad (12)$$

and for the D3Q19 lattice,

$$\mathbf{M}_{D3Q19}^{\tilde{}} = \begin{bmatrix} 0, & 38\mathbf{u} \cdot \mathbf{F}, & -11\mathbf{u} \cdot \mathbf{F}, & F_x, & -\frac{2}{3}F_x, & F_y, & -\frac{2}{3}F_y, & F_z, & -\frac{2}{3}F_z, \\ 4uF_x - 2vF_y - 2wF_z, & -2uF_x + vF_y + wF_z, & 2vF_y - 2wF_z, \\ -vF_y + wF_z, & uF_y + vF_x, & vF_z + wF_y, & uF_z + wF_x, & 0, & 0, & 0 \end{bmatrix}^T \quad (13)$$

Here, the body force is defined as $\mathbf{F} = \rho g \beta (T - T_0) \hat{\mathbf{y}}$. The macroscopic density ρ and velocity \mathbf{u} are obtained via $\rho = \sum_{i=0}^{q-1} f_i$, $\mathbf{u} = \frac{1}{\rho} (\sum_{i=0}^{q-1} \mathbf{e}_i f_i + \mathbf{F}/2)$.

The evolution equation for the temperature distribution function is

$$g_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) - g_i(\mathbf{x}, t) = -(\mathbf{N}^{-1} \mathbf{Q})_{ij} \left[\mathbf{n}_j(\mathbf{x}, t) - \mathbf{n}_j^{(\text{eq})}(\mathbf{x}, t) \right] \quad (i = 0, 1, \dots, q-1) \quad (14)$$

where g_i is the temperature distribution function. For the D2Q5 lattice, $q = 5$; for the D3Q7 lattice, $q = 7$. The equilibrium moments $\mathbf{n}^{(\text{eq})}$ are given by, for the D2Q5 lattice

$$\mathbf{n}_{D2Q5}^{(\text{eq})} = [T, uT, vT, a_T T, 0]^T \quad (15)$$

and for the D3Q7 lattice,

$$\mathbf{n}_{D3Q7}^{(\text{eq})} = [T, uT, vT, wT, a_T T, 0, 0]^T \quad (16)$$

Here, a_T is a parameter related to thermal diffusivity: $a_T = 20\sqrt{3}\alpha - 4$ for D2Q5 lattice, and $a_T = 42\sqrt{3}\alpha - 6$ for D3Q7 lattice. The macroscopic temperature T is then obtained as $T = \sum_{i=0}^{q-1} g_i$.

In the standard LB model described above, it is implicitly assumed that the computational mesh is uniform and that the lattice spacing matches the distance traveled by the distribution functions in a single time step. Under this assumption, the evolution of the density distribution function (Eq. (9)) can be decomposed into two sub-steps:

$$\text{Collision step : } f_i^+(\mathbf{x}, t) = f_i(\mathbf{x}, t) - (\mathbf{M}^{-1} \mathbf{S})_{ij} \left[\mathbf{m}_j(\mathbf{x}, t) - \mathbf{m}_j^{(\text{eq})}(\mathbf{x}, t) \right] + \delta_i F_i' \quad (17)$$

$$\text{Streaming step : } f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) = f_i^+(\mathbf{x}, t) \quad (18)$$

where $f_i^+(\mathbf{x}, t)$ denotes the post-collision density distribution function. Similarly, the evaluation of the temperature distribution function (i.e., Eq. (14)) is split into:

$$\text{Collision step : } g_i^+(\mathbf{x}, t) = g_i(\mathbf{x}, t) - (\mathbf{N}^{-1} \mathbf{Q})_{ij} \left[\mathbf{n}_j(\mathbf{x}, t) - \mathbf{n}_j^{(\text{eq})}(\mathbf{x}, t) \right] \quad (19)$$

$$\text{Streaming step : } g_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) = g_i^+(\mathbf{x}, t) \quad (20)$$

where $g_i^+(\mathbf{x}, t)$ represents the post-collision temperature distribution function. Further numerical details on the thermal LB method can be found in our previous works [8,9,37].

2.2. The interpolation-supplemented LBM for simulating thermal convection on non-uniform meshes

In this work, we adopt the procedure proposed by He et al. [26–28] to update the distribution function f_i and g_i on a non-uniform mesh. The key idea is to approximate the values of the distribution functions at grid points via interpolation from their corresponding values at off-lattice locations. Let $\mathbf{x} = (x_i, y_j)$ denote a grid point on an arbitrary rectangular non-uniform mesh in a Cartesian coordinate system. First, the collision steps (i.e. Eqs. (17) and (19)) are performed locally at each grid point, yielding the post-collision distribution functions $f_i^+(\mathbf{x}, t)$ and $g_i^+(\mathbf{x}, t)$. Next, the virtual streaming steps (i.e. Eqs. (18) and (20)) are applied to compute the distribution functions at the off-lattice locations $\mathbf{x} + \mathbf{e}_i \delta_t$, i.e., $f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t)$ and $g_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t)$. Unlike the uniform

mesh case, where $\mathbf{x} + \mathbf{e}_i \delta_t$ coincides with a grid point, these locations are generally off-lattice in a non-uniform mesh. At such points, the values of $f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t)$ and $g_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t)$ are set equal to the post-collision values $f_i^+(\mathbf{x}, t)$ and $g_i^+(\mathbf{x}, t)$, respectively. These off-lattice values are then interpolated back to the nearest grid point \mathbf{x} , providing the updated distribution functions $f_i(\mathbf{x}, t + \delta_t)$ and $g_i(\mathbf{x}, t + \delta_t)$. In this manner, the collision and the streaming steps are repeated iteratively. To preserve second-order spatial accuracy, a higher-order quadratic interpolation scheme following the formulation proposed by He [28] is employed. This step is essential for consistent and rigorous recovery of the Navier–Stokes equations within the LBM framework.

Fig. 1 illustrates the interpolation procedure in the ISLBM for a two-dimensional non-uniform mesh. After the streaming step, the distribution functions generally arrive at off-lattice positions (see blue open circles). Quadratic interpolation stencils (see red boxes) are then constructed from neighboring post-streaming nodes to recover the distribution function at virtual nodes (see yellow triangles). The interpolated values at the virtual nodes are subsequently used to reconstruct the fluid nodes on the mesh (see black filled circles). This process completes one collision–streaming cycle on a non-uniform lattice.

In the interpolation kernel, the main contributions to the total computational cost are as follows. The first is off-lattice memory access, which requires fetching distribution values from non-contiguous global memory addresses, leading to irregular and non-coalesced reads. This constitutes the dominant bottleneck. The second is the evaluation of interpolation coefficients, which includes on-the-fly computation of quadratic interpolation weights (e.g., Lagrange polynomial coefficients) and depends on the degree of local mesh non-uniformity. The third is stencil selection and indexing, which involves identifying the appropriate neighboring nodes and determining their physical coordinates for accurate interpolation.

2.3. Boundary conditions and implementation issues

In the ISLBM framework, boundary conditions are implemented using the same kinetic boundary schemes as in uniform-mesh LBM. Specifically, the half-way bounce-back scheme is employed to enforce the no-slip velocity boundary condition, while the half-way anti-bounce-back scheme is applied for Dirichlet (constant-temperature) thermal boundaries and the half-way bounce-back scheme for Neumann (adiabatic) thermal boundaries.

At the fluid-solid interface, the no-slip velocity boundary condition is imposed using the half-way bounce-back scheme:

$$f_{\bar{i}}(\mathbf{x}, t + \delta_t) = f_i^+(\mathbf{x}, t) \quad (21)$$

where, $f_{\bar{i}}(\mathbf{x}, t)$ denotes the density distribution function corresponding to the discrete velocity $\mathbf{e}_{\bar{i}} = -\mathbf{e}_i$.

For thermal boundary conditions, the Dirichlet (constant-temperature) case is imposed via the half-way anti-bounce-back scheme:

$$g_{\bar{i}}(\mathbf{x}, t + \delta_t) = -g_i^+(\mathbf{x}, t) + \omega_i T_w \quad (22)$$

where T_w is the prescribed wall temperature. For the D2Q5 lattice, the weights are $\omega_{1-4} = (4 + a_T)/10$; for the D3Q7 lattice, the weights are $\omega_{1-6} = (6 + a_T)/21$. The Neumann (adiabatic) boundary condition is enforced using the half-way bounce-back scheme:

$$g_{\bar{i}}(\mathbf{x}, t + \delta_t) = g_i^+(\mathbf{x}, t) \quad (23)$$

where $g_{\bar{i}}(\mathbf{x}, t)$ denotes the temperature distribution function associated with the discrete velocity $\mathbf{e}_{\bar{i}}$.

We now discuss several implementation issues encountered when collecting data for post-analysis. The first concerns the distribution of grid points and the corresponding length scale. Assume convection is simulated within a cell of length L and height H . Fig. 2 illustrates typical examples of a uniform mesh and a non-uniform mesh with

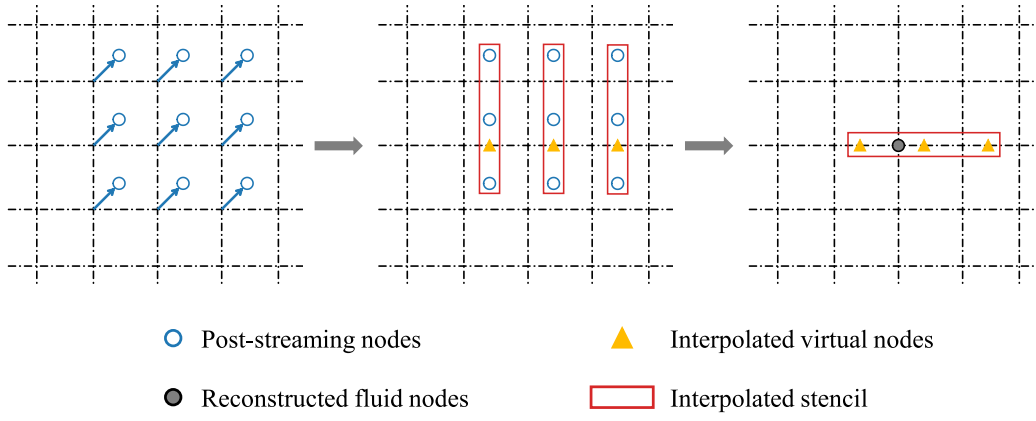


Fig. 1. Illustration of the interpolation procedure in the interpolation-supplemented lattice Boltzmann method (ISLBM) on a two-dimensional non-uniform mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Length scales in the physical system and in the LB systems with uniform and non-uniform meshes.

	Physical system	LB system with uniform mesh	LB system with non-uniform mesh
Effective cell height	H m	1	$1 - y_1^*$
Minimal resolution (1 l.u.)	–	$1/N_y$	y_1^*
Characteristic length	–	N_y l.u.	$(1/y_1^* - 1)$ l.u.

clustering near the walls. For a uniform mesh, the coordinates (taking the y -direction as an example) are specified as

$$y_0^* = 0, \quad y_{N_y+1}^* = 1, \quad (24)$$

$$y_j^* = \frac{j - 0.5}{N_y}, \quad j = 1, \dots, N_y \quad (25)$$

where the indices 1 and N_y correspond to the fluid nodes nearest to the solid walls (denoted by the black circles in Fig. 2a). The spatial resolution is uniform, with $\Delta y^* = 1/N_y = 1$ l.u., where “l.u.” denotes the lattice length unit [44]. Because the half-way bounce-back (or half-way anti-bounce-back) scheme is applied at the fluid-solid boundaries, the first layer of fluid nodes, y_1^* and $y_{N_y}^*$, are offset by 0.5 l.u. from the solid walls (denoted by the red lines at $y_0^* = 0$ and $y_{N_y+1}^* = 1$). This offset arises naturally from the half-way formulation and holds regardless of whether the mesh is uniform or non-uniform.

For non-uniform meshes, the coordinates (taking the y -direction as an example) are given by the error-function stretching [45]:

$$y_j^* = \frac{1}{2} \left\{ \frac{\text{erf} \left[a \left(\frac{j}{N_y+1} - 0.5 \right) \right]}{\text{erf}(a/2)} + 1 \right\}, \quad j = 0, 1, \dots, N_y + 1 \quad (26)$$

where the indices 1 and N_y correspond to the fluid nodes nearest to the solid walls, and the minimal spatial mesh spacing is $(\Delta y^*)_{\min} = y_1^* = 1$ l.u. Unlike the uniform case, the solid walls (denoted by the red lines in Fig. 2b) are located at $y_{\text{bottom}}^* = 0.5y_1^*$ and $y_{\text{top}}^* = 1 - 0.5y_1^*$, so the effective cell height is $1 - y_1^*$. As marked in Fig. 2(b), the offset between the boundary (red line) and the first fluid node (black circle) is 0.5 l.u. A summary of the length scales in the LB systems with uniform and non-uniform meshes is presented in Table 1.

In the stretching error function (i.e. Eq. (26)), a is a positive coefficient that controls the degree of stretching. In this work, we set $a = 1.5$ unless otherwise specified. A larger value of a produces a more pronounced S-shape in the error function; consequently, mesh points are more densely packed near the boundaries and more sparsely distributed in the interior. Conversely, a smaller value of a makes the

error function vary more linearly across the domain. As a approaches zero, the coordinate distribution y_j^* approaches j/N_{y+1} , corresponding to a uniform mesh. To ensure the robustness of our mesh design, we quantitatively evaluated its quality using two widely accepted metrics. The first metric is the growth rate, defined as the ratio of adjacent cell sizes (e.g., $\Delta x_{i+1}/\Delta x_i$). Excessive growth rates can compromise numerical accuracy and stability. In our simulations with a 513×513 mesh and stretching parameter $a = 1.5$, the maximum growth rate is 1.0044, indicating a very smooth grid transition. This value is well below the commonly accepted upper threshold of 1.2, ensuring high interpolation fidelity. The second metric is the aspect ratio, defined as the ratio of the longest to the shortest side of a computational cell. While elevated aspect ratios may be acceptable near boundary layers, where flow features are anisotropic, they are undesirable in core-flow regions. In our simulations, the maximum aspect ratio is 1.75, observed near the four boundaries. This value remains well within acceptable limits for structured grid-based simulations.

The second issue concerns the calculation of derivatives of flow variables at solid walls. The boundary schemes remain valid on non-uniform meshes, provided that the first fluid node adjacent to the wall is located 0.5 lattice units from the physical boundary. This half-spacing offset must be accounted for in derivative calculations. As shown in Fig. 2(a), for a uniform mesh the first-order derivative of a flow variable ϕ at the wall is evaluated using a one-sided second-order finite-difference scheme:

$$\frac{\partial \phi}{\partial y} \Big|_{\text{bottom}} = \frac{-8\phi_{\text{wall}} + 9\phi_1 - \phi_2}{3\Delta y}, \quad \frac{\partial \phi}{\partial y} \Big|_{\text{top}} = \frac{8\phi_{\text{wall}} - 9\phi_{N_y} + \phi_{N_y+1}}{3\Delta y} \quad (27)$$

where $\Delta y = 1$ l.u. is the minimal spatial resolution. For non-uniform meshes, the first-order derivative of ϕ at the wall is expressed as

$$\frac{\partial \phi}{\partial y} \Big|_{\text{bottom}} = \frac{-4q(q+1)\phi_{\text{wall}} + (2q+1)^2\phi_1 - \phi_2}{q(2q+1)\Delta y}, \quad \frac{\partial \phi}{\partial y} \Big|_{\text{top}} = \frac{4q(q+1)\phi_{\text{wall}} - (2q+1)^2\phi_{N_y} + \phi_{N_y+1}}{q(2q+1)\Delta y} \quad (28)$$

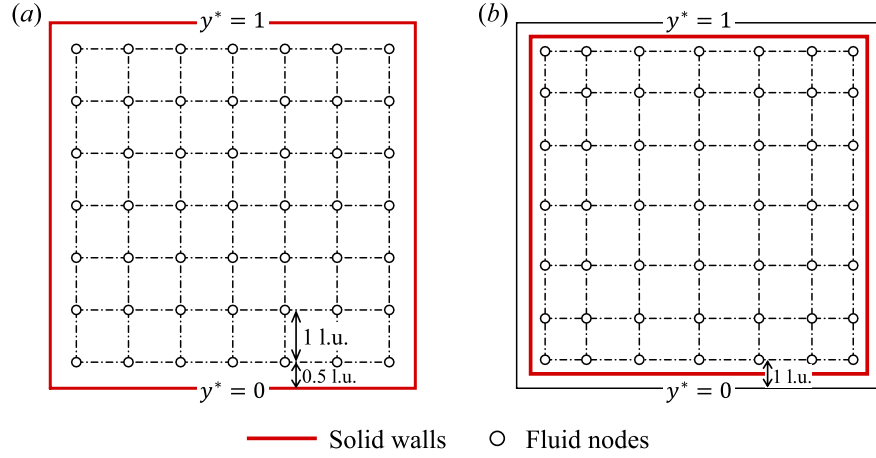


Fig. 2. Examples of (a) uniform mesh and (b) non-uniform mesh with half-way bounce-back scheme. Here, l.u. denotes the lattice length-unit [44]. The first fluid node lies 0.5 l.u. from the wall due to the half-way bounce-back/anti-bounce-back scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $q = \Delta y_2^*/\Delta y_1^* > 1$ is the ratio of the second to the first mesh spacing, with $\Delta y_2^* = y_2^* - y_1^*$ and $\Delta y_1^* = y_1^* - y_0^*$.

The third issue is the calculation of spatial averages of flow variables. Based on the midpoint rule, the weighted average of a variable ϕ on a non-uniform mesh is computed as

$$\langle \phi_j \rangle = \sum_{j=1}^{N_y} \phi_j \cdot \frac{y_{j+1}^* - y_{j-1}^*}{2} \quad (29)$$

This formulation ensures that the contribution of each data point to the overall average reflects its relative spatial extent within the domain. However, caution must be exercised when interpreting the results. As illustrated in Fig. 2(b), the vertical extent of the flow domain does not span the full unit length, since the solid walls are located at $y_{bottom}^* = 0.5y_1^*$ and $y_{top}^* = 1 - 0.5y_1^*$, due to the half-way bounce-back scheme. Consequently, the average along the vertical flow direction should be normalized by the effective cell height $1 - y_1^*$.

3. Laminar convection in a 2-D side-heated cavity

We first consider thermal convection in the canonical 2-D side-heated cavity [46]. The vertical walls are maintained at constant hot and cold temperatures, respectively, while the horizontal walls are adiabatic. All four walls enforce no-slip velocity boundary conditions. Although no exact analytical solution exists, this configuration has long served as a benchmark for coupled fluid flow and heat transfer, ever since Davis [46] reported comprehensive results on the flow structures and heat transport. A wide variety of solvers and datasets have been provided for convection at $Ra \leq 10^6$. However, relatively fewer benchmark datasets are available at $Ra \geq 10^7$, since simulating thermal flows at higher Rayleigh number requires more robust numerical methods and greater computational resources. At high Ra , the thermal and velocity boundary layers become extremely thin, demanding high near-wall resolution. This is precisely where non-uniform meshes can deliver substantial computational savings without compromising accuracy, making the side-heated cavity a natural testbed for assessing both the accuracy and efficiency of ISLBM. Previously, we provided benchmark-quality results using a fine mesh of 2049² uniform mesh points [8], which were subsequently verified by Sun and Tao [47], Chen et al. [48], Vesper et al. [49], and many others [50–58]. Here, to validate the ISLBM for simulating coupled fluid flow and heat transfer, we present results for $10^6 \leq Ra \leq 10^8$ with the Prandtl number fixed at $Pr = 0.71$. At these Rayleigh numbers, the flow remains steady and laminar in two dimensions, but the boundary layers become extremely thin, posing significant challenges for numerical resolution. Under these

conditions, the flow is considered to have reached a steady state when the following convergence criteria are satisfied:

$$\frac{\sum_i \|\mathbf{u}(x_i, t + 2000\delta_t) - \mathbf{u}(x_i, t)\|_2}{\sum_i \|\mathbf{u}(x_i, t)\|_2} < 10^{-9}, \quad (30)$$

$$\frac{\sum_i |T(x_i, t + 2000\delta_t) - T(x_i, t)|}{\sum_i |T(x_i, t)|} < 10^{-9}$$

where $\|\mathbf{u}\|_2$ denotes L^2 norm of the velocity field \mathbf{u} .

In LBM simulations, a reference velocity must be specified to fully determine the relaxation parameters, owing to the weakly compressible nature of the method. For convection problems, we adopt the free-fall velocity $u_f = \sqrt{\beta g H \Delta T}$ as the characteristic velocity, which yields the dimensionless Mach number $Ma = u_f/c_s = \sqrt{\beta g H \Delta T}/c_s$. Here, $c_s = 1/\sqrt{3}$ l.u./t.s. is the lattice speed of sound. Unless otherwise stated, we set $Ma = 0.1$ in all simulations. To approximate incompressibility, the Mach number should be kept as small as possible. In LBM simulations, however, the ratio of the dimensionless time step to the dimensionless spatial resolution is given by $dt/dx = (\delta_t/\sqrt{H/(\beta g \Delta T)})/(\delta_x/H) = \sqrt{\beta g H \Delta T} = Ma \cdot c_s$, with $\delta_x = 1$ l.u. and $\delta_t = 1$ t.s. Thus, reducing the Mach number decreases the dimensionless time step for a fixed mesh resolution, which in turn reduces computational efficiency in time marching. Therefore, our choice of $Ma = 0.1$ represents a compromise between approximating incompressibility and maintaining computational efficiency.

In Fig. 3, we present the temperature and velocity fields. At $Ra = 10^6$ (see Fig. 3a), the temperature distribution exhibits relatively thick thermal boundary layers along the hot (left) and cold (right) walls, with smooth stratification in the interior. As Ra increases to 10^7 and 10^8 (see Figs. 3b and 3c), these boundary layers become thinner, reflecting enhanced thermal gradients and stronger convective transport. The velocity fields reveal a single dominant clockwise large-scale circulation cell in all cases (see Figs. 3d–f), with elevated velocity magnitudes near the boundary layers. At higher Ra , the maximum velocities are concentrated in narrow shear layers adjacent to the sidewalls, while the streamlines indicate intensified plume activity near the vertical boundaries. The core region remains relatively quiescent and horizontally stratified. Overall, the flow structure agrees well with earlier 2-D simulations (see Fig. 10 in our previous work [8] and with many other studies [47–49]).

Figs. 4(a) and 4(b) show the horizontal distributions of the temperature T^* and vertical velocity v^* , each averaged over the mid-plane band $0.4 \leq y \leq 0.6$ for three Rayleigh numbers. This averaging minimizes end-wall effects and yields smooth profiles suitable for boundary-layer

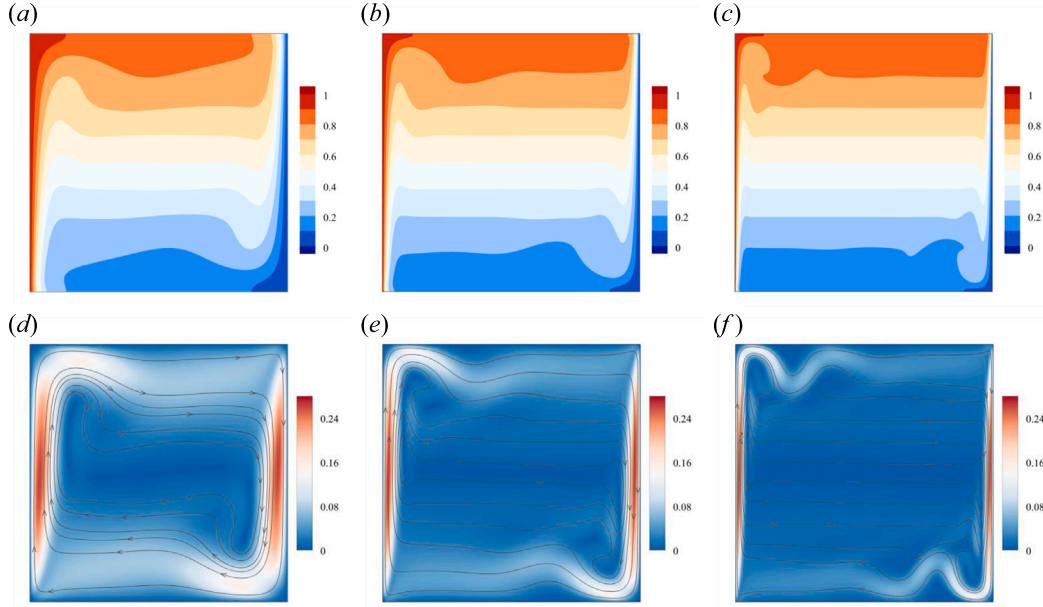


Fig. 3. Contours of (a–c) temperature field T^* and (d–f) velocity magnitude $\sqrt{u^{*2} + v^{*2}}$ (superposed with streamlines), at (a, d) $Ra = 10^6$, (b, e) $Ra = 10^7$, and (c, f) $Ra = 10^8$.

analysis. Pronounced gradients are observed near the sidewalls, reflecting the development of thermal and momentum boundary layers adjacent to the hot and cold boundaries. To examine these boundary layers in more detail, Figs. 4(c) and 4(d) present enlarged views near the hot wall ($x^* = 0$). The boundary-layer thicknesses are determined using two widely adopted approaches: the 99% criterion method and the slope method. In the 99% criterion method, the thermal boundary layer thickness δ_T is defined as the distance from the wall where T^* attains 99% of the difference between the wall and the core value, while the velocity boundary layer thickness δ_V is defined as the distance where v^* reaches 99% of its maximum magnitude. In the slope method, δ_T is obtained as the ratio of the core-to-wall temperature difference to the near-wall temperature gradient, and δ_V as the ratio of the maximum velocity to the near-wall velocity gradient. Power-law fits show that both methods yield consistent results, with $\delta_T \sim Ra^{-0.25}$ and $\delta_V \sim Ra^{-0.25}$ at $Pr = 0.71$. These results confirm that the thermal and velocity boundary layers exhibit comparable scaling with Rayleigh number.

In Table 2, we present quantitative results for heat transfer efficiency in terms of the Nusselt number, and for global flow strength in terms of the Reynolds number. Specifically, the volume-averaged Nusselt number $\langle Nu \rangle$ is defined as

$$\langle Nu \rangle = \sqrt{RaPr} \langle u^* T^* \rangle_V + 1 \quad (31)$$

the average Nusselt number along the hot wall Nu_{hot} is given by

$$Nu_{\text{hot}} = - \langle \frac{\partial T^*}{\partial x^*} \rangle_{x^*=0} \quad (32)$$

and the average Nusselt number along the vertical centerline Nu_{middle} is expressed as

$$Nu_{\text{middle}} = \langle \sqrt{RaPr} u^* T^* - \frac{\partial T^*}{\partial x^*} \rangle_{x^*=0.5} \quad (33)$$

The volume-averaged Reynolds number $\langle Re \rangle$ is defined as

$$\langle Re \rangle = \sqrt{\frac{Ra}{Pr}} \langle \sqrt{u^{*2} + v^{*2}} \rangle_V \quad (34)$$

Here, $\langle \cdot \rangle_V$ denotes a volume average, $\langle \cdot \rangle_{x^*=0}$ a line average along the hot wall, and $\langle \cdot \rangle_{x^*=0.5}$ a line average along the vertical centerline. For comparison, we also include benchmark data from previous studies: Wang et al. [59], who employed a finite-difference method to

Table 2

Benchmark solutions of the Nusselt numbers and Reynolds number. Columns from left to right indicate: Rayleigh number Ra ; flow database; grid number $N_x \times N_y$; volume-averaged Nusselt number $\langle Nu \rangle$; average Nusselt number along the hot wall Nu_{hot} ; average Nusselt number along the vertical centerline Nu_{middle} ; volume-averaged Reynolds number $\langle Re \rangle$.

Ra	Flow database	$N_x \times N_y$	$\langle Nu \rangle$	Nu_{hot}	Nu_{middle}	$\langle Re \rangle$
10^6	Present	257 ²	8.8282	8.8509	8.8277	99.6110
	Present	513 ²	8.8202	8.8317	8.8200	99.3839
	Present	1025 ²	8.8188	8.8246	8.8188	99.3576
	∞	–	8.8186	8.8206	8.8186	99.3542
	p	–	2.6	1.5	2.6	3.1
	GCI (%)	–	0.0037	0.058	0.0033	0.0043
	Wang [59]	128 ²	8.830	–	–	99.11
	Le Quéré [60]	72 ²	–	8.825	8.825	–
	Contrino [39]	2043 ²	8.8252	8.8252	8.8252	–
	Present	513 ²	16.5245	16.5461	16.5243	230.6601
10^7	Present	1025 ²	16.5133	16.5242	16.5133	230.2448
	Present	2049 ²	16.5114	16.5169	16.5114	230.1948
	∞	–	16.5110	16.5132	16.5110	230.1879
	p	–	2.5	1.6	2.6	3.1
	GCI (%)	–	0.0030	0.028	0.0029	0.0037
	Xu [8]	2049 ²	16.52414	16.52229	16.52428	–
	Wang [59]	256 ²	16.528	–	–	229.70
	Le Quéré [60]	80 ²	–	16.523	16.523	–
	Contrino [39]	2043 ²	16.5231	16.5233	16.5232	–
	Present	513 ²	30.3130	30.3503	30.3125	542.9352
10^8	Present	1025 ²	30.2225	30.2423	30.2223	537.3420
	Present	2049 ²	30.2072	30.2172	30.2072	536.6449
	∞	–	30.2041	30.2096	30.2041	536.5456
	p	–	2.6	2.1	2.6	3.0
	GCI (%)	–	0.013	0.031	0.013	0.023
	Xu [8]	2049 ²	30.23013	30.22650	30.23050	–
	Wang [59]	256 ²	30.242	–	–	534.23
	Le Quéré [60]	128 ²	–	30.225	30.225	–
	Contrino [39]	2043 ²	30.2255	30.2268	30.2261	–
	Present	513 ²	30.3130	30.3503	30.3125	542.9352

solve low-Mach-number equations; Le Quéré [60], who used a pseudo-spectral Chebyshev algorithm; Contrino et al. [39], who applied an MRT-LBM with a forcing term split using the Strang–Marchuk scheme; and our previous work [8], which used the LBM with Guo’s forcing scheme [42,43,61] on a uniform mesh.

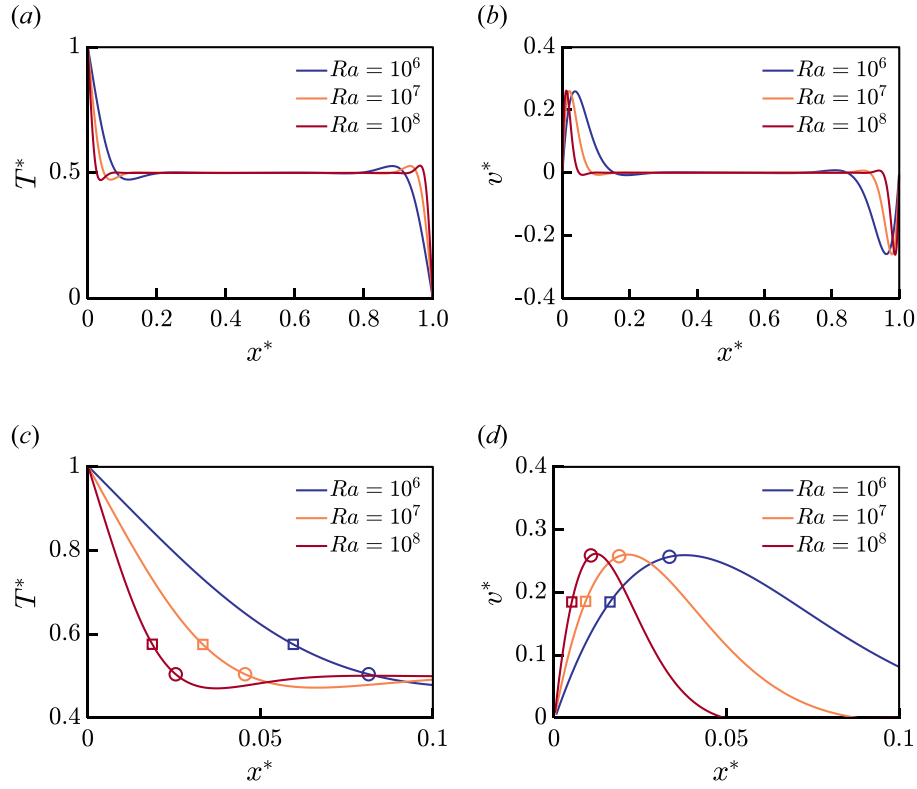


Fig. 4. Horizontal distributions of vertically averaged (a) temperature T^* and (b) vertical velocity v^* . Panels (c, d) show enlarged views near the hot wall ($x^* = 0$), where open circles denote boundary layer locations determined by the 99% criterion method and open squares denote those determined by the slope method.

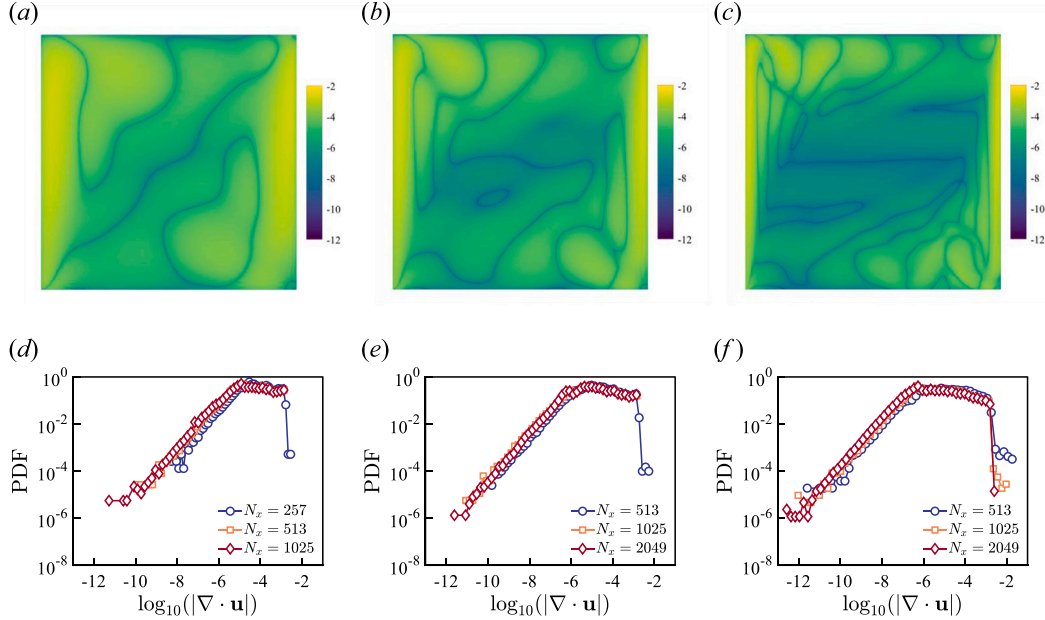


Fig. 5. (a–c) Contours of the logarithm of velocity divergence magnitude, $\log_{10}(|\nabla \cdot \mathbf{u}|)$. (d–f) Probability density functions (PDFs) of $\log_{10}(|\nabla \cdot \mathbf{u}|)$, obtained over the entire cell, at (a, d) $Ra = 10^6$, (b, e) $Ra = 10^7$ and (c, f) $Ra = 10^8$.

The comparison in Table 2 shows that the present data converge toward the asymptotic value (discussed later) and are in good agreement with previous studies. An interesting observation is that the present Nusselt numbers are slightly lower than those reported in our earlier work [8], with deviations of less than 0.1%. We attribute this small discrepancy to the weak compressibility of the current LB model. In our previous work [8], only density fluctuation $\delta\rho$ were included in certain components of the equilibrium moments $\mathbf{m}^{(eq)}$ (see Eq. (3) in [8]), which provided a good approximation for incompressible flows

in steady state. In the present study, however, we incorporate the full density ρ in the equilibrium moments $\mathbf{m}^{(eq)}$, resulting in a weakly compressible formulation that enhances numerical stability. As also reported by Wang et al. [59] and Wen et al. [62,63], weak compressibility leads to a slight decrease in Nu and a slight increase in Re , without significantly affecting the scaling exponents with respect to Ra .

To assess the compressibility effects intrinsic to the LBM employed in this study, we examine both the spatial distribution and statistical

properties of the velocity divergence field. Although the LBM formulation is nearly incompressible, the simulations operate in a weakly compressible regime with a Mach number of $Ma = 0.1$, necessitating a careful evaluation of divergence behavior. Figs. 5(a–c) show contour maps of the base-10 logarithm of the velocity divergence magnitude, $\log_{10}(|\nabla \cdot \mathbf{u}|)$, computed over the entire domain. In all cases, the divergence field remains predominantly low, with most of the domain exhibiting values below 10^{-4} , consistent with the nearly incompressible character of the formulation. At $Ra = 10^6$ (see Fig. 5a), the divergence field is characterized by smoother structures, reflecting weak localized deviations from incompressibility. As Ra increases to 10^7 (see Fig. 5b), finer features emerge, with localized regions of elevated divergence magnitude appearing intermittently throughout the domain. At the highest Rayleigh number considered, $Ra = 10^8$ (see Fig. 5c), the divergence field develops elongated and filamentary structures aligned with regions of intensified thermal plumes and sharp velocity gradients. Although local compressibility effects are slightly amplified at higher Rayleigh numbers, they remain small in magnitude and spatially confined. This ensures that the weakly compressible LBM faithfully reproduces the incompressible flow characteristics.

To quantitatively assess the divergence statistics, Figs. 5(d–f) show the probability density functions (PDFs) of $\log_{10}(|\nabla \cdot \mathbf{u}|)$, evaluated at three spatial resolutions for each Rayleigh number. In all cases, the PDFs peak within $|\nabla \cdot \mathbf{u}| \in [10^{-7}, 10^{-3}]$, with long tails extending down to 10^{-11} , again consistent with a nearly incompressible flow field. As resolution increases, the PDFs converge, particularly near the most probable values. In addition, a modest increase in the domain-averaged divergence is observed when Ra increases from 10^6 to 10^8 . This trend is physically reasonable, as stronger thermal forcing steepens velocity and temperature gradients, thereby amplifying local compressibility effects in weakly compressible formulations. Similar divergence characteristics have also been reported in high-fidelity incompressible simulations. For example, Ostilla-Mónico et al. [64] observed locally non-solenoidal fields with residual divergence of order of $O(10^{-3})$ in multiple-resolution incompressible Navier–Stokes simulations. To the best of our knowledge, such low levels of divergence do not introduce significant qualitative or quantitative deviations, even in turbulent convection regimes.

Our non-uniform mesh is systematically refined to preserve geometric similarity while halving the effective mesh spacing at each refinement level. To assess convergence, we employ Richardson extrapolation using three successively refined meshes. For $Ra = 10^6$, the grid resolutions are $(N_{\text{coarse}}, N_{\text{medium}}, N_{\text{fine}}) = (257, 513, 1025)$, and for $Ra = 10^7$ and 10^8 , they are $(N_{\text{coarse}}, N_{\text{medium}}, N_{\text{fine}}) = (513, 1025, 2049)$. Throughout, the effective mesh spacing follows $\Delta x_i \propto 1/N_i$, with refinement ratio $r = \Delta x_{\text{coarse}}/\Delta x_{\text{medium}} = \Delta x_{\text{medium}}/\Delta x_{\text{fine}} = 2$. Let $F(\Delta x)$ denote a flow quantity computed on mesh spacing Δx . The observed order of convergence p is then estimated as

$$p = \frac{\ln \left(\left| \frac{F(\Delta x_{\text{coarse}}) - F(\Delta x_{\text{medium}})}{F(\Delta x_{\text{medium}}) - F(\Delta x_{\text{fine}})} \right| \right)}{\ln(r)} \quad (35)$$

The corresponding asymptotic, grid-independent value is obtained as

$$F(1/\infty) \approx \frac{r^p F(\Delta x_{\text{fine}}) - F(\Delta x_{\text{medium}})}{r^p - 1} \quad (36)$$

Table 2 reports the estimated convergence orders and extrapolated asymptotic values of the Nusselt and Reynolds numbers. Although the table lists four significant digits for readability, all computations were performed with at least fourteen-digit precision. The interpolation-supplemented LBM solver achieves nearly third-order spatial accuracy in domain-integrated quantities, such as volume-averaged Nusselt and Reynolds numbers. This enhanced accuracy arises because the integration process inherently damps local discretization errors. To further examine this hypothesis, Table 3 reports convergence orders for local flow variables evaluated at the cell center. The vorticity field, being a derivative of velocity, exhibits close to third-order accuracy, while

Table 3

Convergence behavior of flow variables at cell center. The columns from left to right indicate the following: Rayleigh number Ra ; absolute value of vorticity $|\omega_c|$; temperature T_c ; horizontal velocity component u_c ; vertical velocity component v_c .

Ra	Grid	$ \omega_c $	T_c	u_c	v_c
10^6	257 ²	0.1175	0.49945	-1.2163×10^{-4}	1.8504×10^{-5}
	513 ²	0.1158	0.49947	-1.3013×10^{-4}	1.9377×10^{-5}
	1025 ²	0.1156	0.49948	-1.3239×10^{-4}	1.9551×10^{-5}
	∞	0.1155	0.49948	-1.3320×10^{-4}	1.9594×10^{-5}
	p	2.6	2.0	1.9	2.3
	GCI (%)	0.054	0.00055	0.77	0.27
10^7	513 ²	0.1088	0.49930	-8.0524×10^{-5}	8.2430×10^{-6}
	1025 ²	0.1076	0.49934	-8.3466×10^{-5}	8.3163×10^{-6}
	2049 ²	0.1075	0.49934	-8.4272×10^{-5}	8.3330×10^{-6}
	∞	0.1074	0.49935	-8.4576×10^{-5}	8.3379×10^{-6}
	p	2.7	2.0	1.9	2.1
	GCI (%)	0.037	0.00068	0.45	0.073
10^8	513 ²	4.1046×10^{-2}	0.49911	-3.7830×10^{-5}	3.1377×10^{-6}
	1025 ²	3.9075×10^{-2}	0.49922	-4.2857×10^{-5}	2.7224×10^{-6}
	2049 ²	3.8864×10^{-2}	0.49925	-4.4429×10^{-5}	2.6433×10^{-6}
	∞	3.8838×10^{-2}	0.49926	-4.5144×10^{-5}	2.6248×10^{-6}
	p	3.2	2.0	1.7	2.4
	GCI (%)	0.082	0.0024	2.0	0.88

local temperature and velocity converge at approximately second-order accuracy.

In addition to assessing the observed order of convergence and extrapolated asymptotic values, we further quantified the discretization uncertainty using the Grid Convergence Index (GCI), following the ASME procedure [65]. For three systematically refined meshes with a refinement ratio of r , the GCI on the fine mesh is computed as

$$\text{GCI}_{\text{fine}} = \frac{F_s}{r^p - 1} \left| \frac{F(\Delta x_{\text{fine}}) - F(\Delta x_{\text{medium}})}{F(\Delta x_{\text{fine}})} \right| \times 100\%, \quad (37)$$

where p is the observed convergence order and $F_s = 1.25$ is the safety factor recommended for three-grid studies. The resulting GCI values are typically below 0.1% for global quantities such as the Nusselt and Reynolds numbers (see Table 2), whereas local velocity components exhibit somewhat larger values, up to $\sim 2\%$ (see Table 3), reflecting their higher sensitivity to grid resolution. These results demonstrate that the fine-mesh solutions lie within the asymptotic range of convergence and that the numerical uncertainty in the reported quantities is negligible.

We evaluate the computational efficiency of our in-house LBM solver by comparing it against two widely used open-source solvers: Nek5000 (version v19.0), which is based on the spectral element method (SEM) [66], and OpenFOAM (version 8), which employs the finite volume method (FVM) [67].

The Nek5000 solver extends the standard finite element method to higher-order polynomial basis functions [66]. In this study, the polynomial order for both velocity and pressure is set to $N = 8$, following the Pn–Pn formulation. This choice is consistent with values commonly adopted in high-fidelity turbulent flow simulations. The Pn–Pn formulation employs a time-splitting scheme that decouples the pressure and velocity fields into three distinct substeps. The pressure Poisson equation is solved using the preconditioned Generalized Minimal Residual (GMRES) method, while the Helmholtz equations for the velocity components are solved using the preconditioned Conjugate Gradient (CG) method. For temporal discretization, a semi-implicit scheme is employed. The viscous term is discretized implicitly using a second-order backward difference scheme, while the nonlinear convection term is discretized explicitly using a second-order extrapolation scheme. The energy equation, governing temperature via a convection–diffusion formulation, is handled analogously: the transient and diffusion terms are treated implicitly with a second-order backward difference scheme, while the convection term is treated explicitly using a second-order extrapolation scheme.

The OpenFOAM solver enforces the integral form of the conservation equations on each control volume. In this study, we employed the transient *buoyantPimpleFoam* solver [67]. The Navier–Stokes equations were solved using the PIMPLE pressure–velocity coupling algorithm. Within the inner iterations, the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) scheme was applied for pressure correction, while the outer iterations used the Pressure Implicit with Splitting of Operators (PISO) scheme to handle transient terms, thereby improving convergence and numerical stability. For temporal discretization, a second-order implicit backward scheme was adopted. Spatial discretization was carried out within the Gauss linear framework, in which volume integrals are evaluated by Gaussian quadrature and face fluxes are approximated using linear interpolation. The pressure equation in the resulting linear systems was solved using the Geometric–Algebraic Multi-Grid (GAMG) method to accelerate convergence for large-scale problems. The velocity and energy equations were solved using the Preconditioned Bi-Conjugate Gradient Stabilized (PBiCGStab) method, which efficiently handles non-symmetric sparse matrices while maintaining both stability and accuracy.

As a performance metric, we adopt the million lattice updates per second (MLUPS), defined as:

$$\text{MLUPS} = \frac{\text{grid size} \times \text{iteration steps}}{\text{running time} \times 10^6} \quad (38)$$

A higher MLUPS value indicates greater computational efficiency. The use of simple geometry and boundary conditions in the side-heated cavity enables direct and fair comparisons with Nek5000 and OpenFOAM, while minimizing discrepancies related to solver-specific geometrical handling or mesh preprocessing. This ensures that the reported performance metrics primarily reflect the core algorithmic efficiency, rather than differences in geometry setup. Figs. 6(a) and 6(b) show MLUPS as a function of grid number on uniform and non-uniform meshes, respectively. The simulations were carried at $Ra = 10^8$ and $Pr = 0.71$, on AMD EPYC 9135 CPUs. For non-uniform meshes, the stretching coefficient in the error-function (i.e. Eq. (26)) was set to $a = 2.1$. Increasing the grid resolution from 1024^2 to 8192^2 has minimal impact on the performance of the in-house LBM solver. In contrast, Nek5000 shows a slight decline, while OpenFOAM exhibits a significant drop in efficiency. To further emphasize the relative differences, Figs. 6(c) and 6(d) present MLUPS values normalized by those of the in-house LBM solver at each grid size. The results clearly show that Nek5000 becomes up to two orders of magnitude slower than LBM at the largest grid size of 8192^2 . For OpenFOAM, the disparity is even more pronounced, reaching up to three orders of magnitude slower at the same resolution. The observed performance advantage of the LBM stems from its reliance almost entirely on local operations (collision and streaming) with no global matrix solves or complex stencil dependencies. This design yields excellent memory locality and computational regularity, allowing near-ideal scaling across CPU cores until memory-bandwidth limits are reached. In contrast, SEM and FVM require global or semi-global solves for the pressure field and other coupled variables, involving sparse matrix–vector products, iterative solvers, and halo exchanges. These introduce higher communication costs and memory-access irregularity. Consequently, the orders-of-magnitude speedup of LBM becomes most pronounced at large grid sizes, where communication overhead and global-solve cost dominate in SEM/FVM frameworks.

For steady problems, different solvers may require varying numbers of time steps to converge, depending on the temporal discretization scheme employed and the inherent numerical damping of each solver. In unsteady simulations, however, it is often necessary to quantify the computational cost of advancing the solution over a prescribed dimensionless physical time (e.g., 1, 10, or 100 free-fall time). This requirement is particularly relevant in large-scale turbulent convection studies, where long integrations are needed to achieve statistical stationarity and to accumulate converged averages. To this end, we introduce the metric WCTpDT (Wall-Clock Time per Dimensionless

Time), defined as:

$$\text{WCTpDT} = \frac{\text{measured wall-clock time}}{\text{simulated dimensionless time}} \quad [\text{s}] \quad (39)$$

which quantifies the wall-clock time required to simulate one unit of dimensionless time on a given solver and hardware configuration. A lower WCTpDT indicates greater efficiency, as fewer computational resources are required to advance the solution over one unit of dimensionless time. Figs. 7(a) and 7(b) show WCTpDT as a function of grid number for uniform and non-uniform meshes, respectively. On uniform meshes, the LBM solver consistently achieves the lowest WCTpDT across all tested resolutions. For non-uniform meshes, OpenFOAM shows a modest advantage at the coarsest resolution, but as grid size increases, the LBM solver demonstrates superior time-advancement efficiency. This advantage becomes even clearer when WCTpDT is normalized by the LBM results, as shown in Figs. 7(c) and 7(d). It is worth noting that the Courant–Friedrichs–Lewy (CFL) numbers adopted were 0.5 for Nek5000 and 0.9 for OpenFOAM, allowing these solvers to employ relatively larger time steps than LBM. On the uniform mesh, the effective nondimensional timestep is $dt/dx = Ma \cdot c_s \approx 0.0577$ for LBM, compared with $dt/dx \approx 0.8$ for Nek5000 and $dt/dx \approx 1.2$ for OpenFOAM. While larger timesteps should, in principle, reduce the cost per simulated physical time, Nek5000 and OpenFOAM exhibit substantially higher per-step overhead. As a result, their WCTpDT values are roughly an order of magnitude larger than those of LBM. This explains why, despite LBM attaining two to three orders of magnitude higher MLUPS values than Nek5000 and OpenFOAM, the relative advantage in WCTpDT is smaller. Nevertheless, as the grid resolution increases, the efficiency gap widens and the LBM solver becomes increasingly advantageous.

To further assess the computational efficiency of the in-house LBM solver on modern GPU architectures, additional performance tests were conducted on an NVIDIA A100 GPU. Fig. 8 compares the solver's performance on uniform and non-uniform meshes in terms of MLUPS and WCTpDT across different grid sizes. As shown in Fig. 8(a), on all tested grids, MLUPS values on uniform meshes consistently exceed those on non-uniform meshes. This disparity stems primarily from the additional interpolation operations required for off-lattice streaming on non-uniform grids, which introduce irregular memory access patterns and reduce throughput. The normalized MLUPS (see Fig. 8b) shows that non-uniform grids achieve only 60%–70% of the performance of uniform grids, depending on resolution. The quadratic interpolation step, particularly its memory-bound indirect access pattern, is the main source of this performance degradation. Uniform meshes permit fully coalesced and predictable memory access, whereas non-uniform interpolation requires fetching data from spatially non-contiguous memory locations, leading to irregular global memory access and reduced throughput. The collision step, in contrast, is compute-bound and consistently sustains high arithmetic throughput regardless of mesh type. Consequently, the observed 30%–40% drop in MLUPS for non-uniform grids reflects interpolation overhead rather than any intrinsic inefficiency of the LBM algorithm. WCTpDT increases nearly linearly with grid size for both mesh types (see Fig. 8c), while the relative overhead associated with non-uniform meshes remains stable at about a factor of three across resolutions (see Fig. 8d). These results confirm that, despite modest penalties from interpolation overhead, the in-house LBM solver retains excellent scalability and GPU efficiency on both mesh types. This enables practical large-scale simulations on contemporary accelerator hardware. It is also worth noting that our LBM solver on GPUs employs double-precision floating point arithmetic to ensure accuracy.

We also assess the economic benefits of the LBM solver on GPU platforms through a simple cost estimation. Consider a simulation at $Ra = 10^8$ and $Pr = 0.71$ using a 8193^2 non-uniform mesh with a stretching coefficient of $a = 2.1$. On an NVIDIA A100 GPU, the wall-clock time required to advance the simulation by 1 t_f (i.e. WCTpDT) is $1.5722 \times$

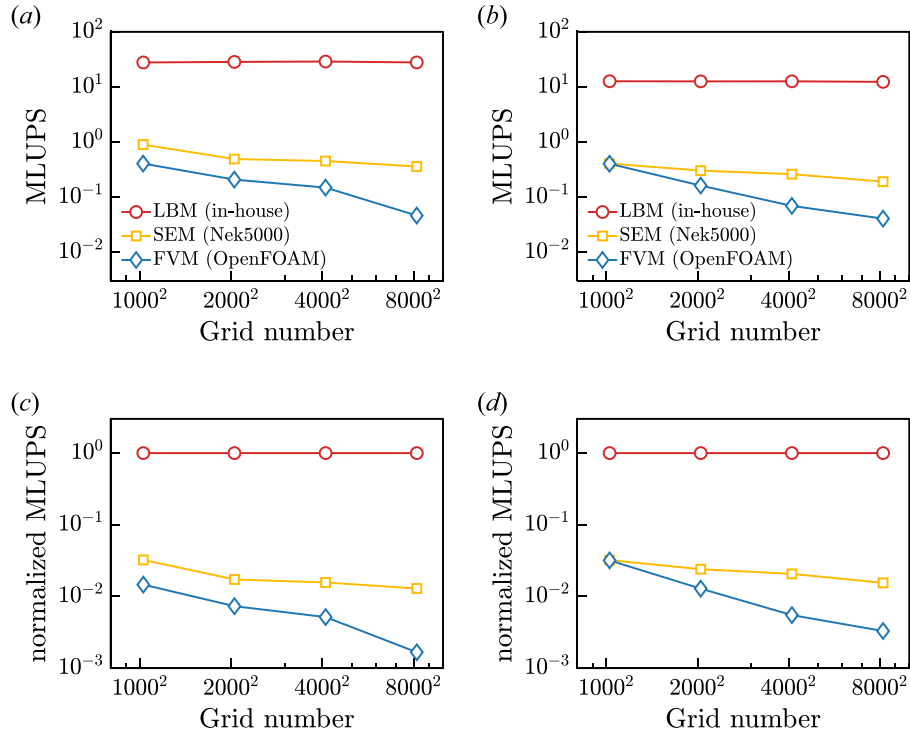


Fig. 6. Performance comparison of three solvers in terms of million lattice updates per second (MLUPS): an in-house solver using the lattice Boltzmann method (LBM), Nek5000 using the spectral element method (SEM), and OpenFOAM using the finite volume method (FVM). (a, b) Absolute MLUPS as a function of grid number; (c, d) MLUPS normalized by the corresponding LBM values, on (a, c) uniform and (b, d) non-uniform meshes.

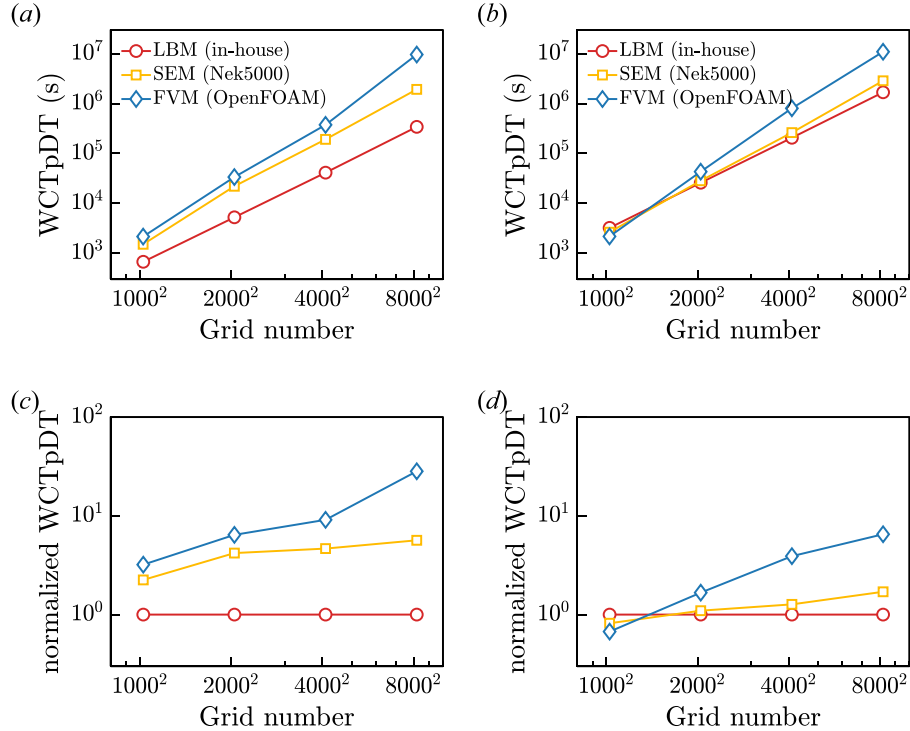


Fig. 7. Performance comparison of in-house LBM solver, Nek5000 and OpenFOAM in terms of wall-clock time per dimensionless time (WCTpDT). (a, b) Absolute WCTpDT as a function of grid number; (c, d) WCTpDT normalized by the corresponding LBM values, on (a, c) uniform and (b, d) non-uniform meshes.

10^4 s \approx 4.37 h. Assuming a nominal rental cost of 7 RMB per GPU-hour, the corresponding cost per t_f is $4.37 \text{ h} \times 7 \text{ RMB/h} \approx 30.57 \text{ RMB}$. By contrast, CPU-based solvers requires significantly longer wall-clock times. For Nek5000, the time per t_f is about $2.871852 \times 10^6 \text{ s} \approx 797.74 \text{ h}$,

yielding a cost of $797.74 \text{ h} \times 0.1 \text{ RMB/h} \approx 79.77 \text{ RMB}$, where we assume a conservative rate of 0.1 RMB per CPU-core hour. For OpenFOAM, the required time increases to $1.0981646 \times 10^7 \text{ s} \approx 3050.5 \text{ h}$, yielding $3050.5 \text{ h} \times 0.1 \text{ RMB/h} \approx 305.05 \text{ RMB}$. It is important to emphasize

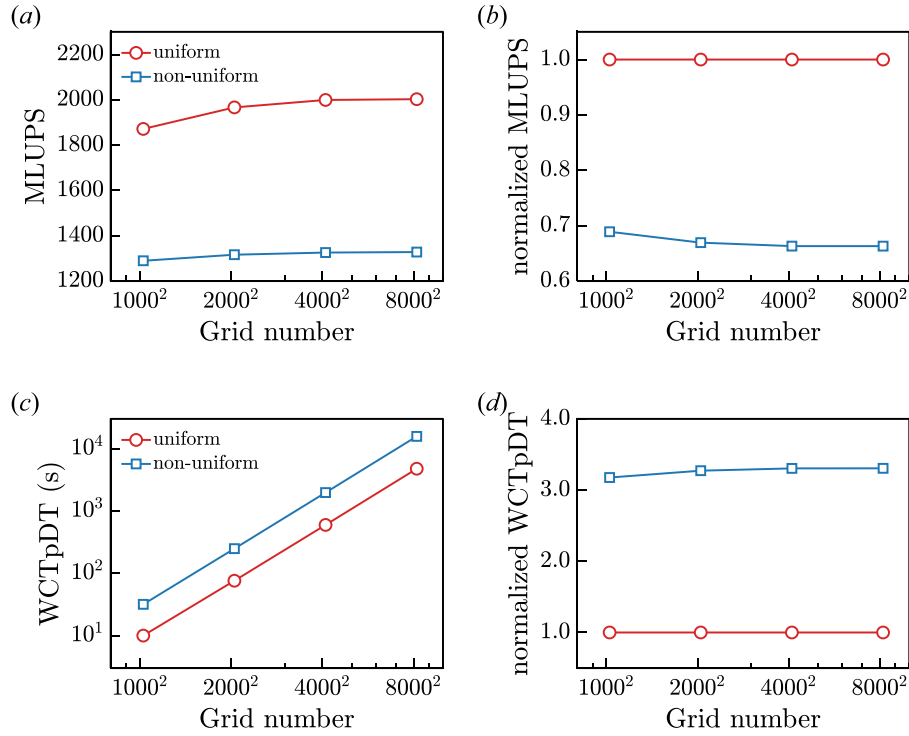


Fig. 8. GPU performance of the in-house LBM solver on an NVIDIA A100 for uniform and non-uniform meshes. (a) Absolute MLUPS as a function of grid number, (b) MLUPS normalized by the corresponding uniform-mesh performance. (c) Absolute WCTpDT as a function of grid number, (d) WCTpDT normalized by the corresponding uniform-mesh value.

that both Nek5000 and OpenFOAM typically run on multi-core CPU architectures, where scalability is often limited by communication overhead and memory contention. Consequently, the effective performance falls short of ideal expectations, further increasing wall-clock times and overall computational cost. These considerations highlight the substantial economic advantage of the GPU-based LBM solver, an advantage that becomes increasingly pronounced in large-scale simulations.

4. Laminar thermal convection in a 3-D side-heated cavity

We next consider thermal convection in the canonical 3-D side-heated cavity [37]. The left and right vertical walls are maintained at constant hot and cold temperatures, respectively, while the other four walls are adiabatic. All six walls impose no-slip velocity boundary conditions. In our previous work [37], we provided results obtained on a uniform mesh of 257³ grid points, which were subsequently verified by Sun and Tao [47], Chen et al. [48], Ren et al. [68,69], Yigit et al. [70], Vesper et al. [49], and many others [55,56,71–73]. Here, to validate the ISLBM for simulating coupled fluid flow and heat transfer in 3-D, we present simulation results for $10^5 \leq Ra \leq 10^7$, with the Prandtl number fixed at 0.71. Under these conditions, the flow reaches a steady state once the same convergence criteria as in the 2-D case are satisfied (see Eq. (30)).

Fig. 9 illustrates the temperature fields T^* , where contour slices in the vertical planes highlight the evolution of thermal structures with increasing buoyancy forcing. At $Ra = 10^5$ (see Fig. 9a), the flow is characterized by relatively smooth and broad thermal gradients with thick boundary layers. As Ra increases to 10^6 (see Fig. 9b), convection becomes more vigorous, leading to thinner boundary layers and sharper vertical temperature gradients, indicative of enhanced plume activity. At $Ra = 10^7$ (see Fig. 9c), convective transport dominates: the temperature field exhibits well-defined plume structures, pronounced vertical mixing, and significantly thinner boundary layers. These visualizations demonstrate the progressive transition from conduction-dominated to convection-dominated regimes as Ra increases. The overall structure

agrees well with earlier 3-D simulations (see Fig. 10 in our previous work [37] and with many other studies [47,48]).

Figs. 10(a) and 10(b) show the horizontal distributions of the temperature T^* and vertical velocity v^* , each averaged over the mid-plane band $0.4 \leq y \leq 0.6$ for three Rayleigh numbers. Both quantities exhibit clear gradients near the sidewalls, indicating the formation of thermal and momentum boundary layers. Enlarged views near the hot wall ($x^* = 0$) are provided in Figs. 10(c) and 10(d), where open circles and squares denote the boundary layer thicknesses determined using the 99% criterion and slope methods, respectively. From these measurements, the scaling relations are obtained as $\delta_T \sim Ra^{-0.26}$ for the thermal boundary layer and $\delta_v \sim Ra^{-0.23}$ for the velocity boundary layer, with consistent results from both methods.

In Table 4, we present quantitative results for heat transfer efficiency in terms of the Nusselt number, and for global flow strength in terms of the Reynolds number. Specifically, the volume-averaged Nusselt number $\langle Nu \rangle$ is defined as

$$\langle Nu \rangle = \sqrt{RaPr(u^*T^*)}_V + 1 \quad (40)$$

and the average Nusselt numbers along the hot and cold walls, Nu_{hot} and Nu_{cold} , are given by

$$Nu_{\text{hot}} = -\langle \frac{\partial T^*}{\partial x^*} \rangle_{\text{hot}}, \quad Nu_{\text{cold}} = -\langle \frac{\partial T^*}{\partial x^*} \rangle_{\text{cold}} \quad (41)$$

The volume-averaged Reynolds number $\langle Re \rangle$ is defined as

$$\langle Re \rangle = \sqrt{\frac{Ra}{Pr} \langle \sqrt{u^{*2} + v^{*2}} \rangle}_V \quad (42)$$

Here, $\langle \cdot \rangle_V$ denotes a volume average, while $\langle \cdot \rangle_{\text{hot/cold}}$ denotes an area average along the hot or cold wall. For comparison, we also include data from previous studies, such as Fusegi et al. [74], who employed a control-volume-based finite difference method with a strongly implicit scheme; Tric et al. [75], who used a pseudo-spectral Chebyshev algorithm based on the projection–diffusion method; and our earlier work [37], which employed the LBM with Guo’s forcing scheme [42, 43,61] on a uniform mesh.

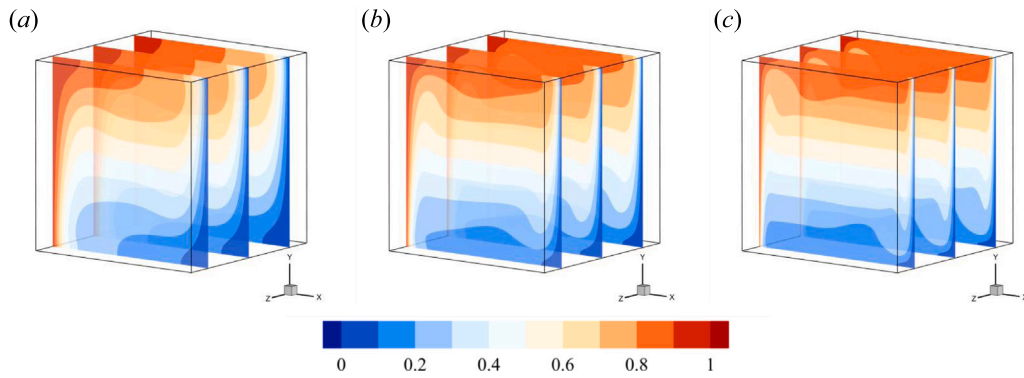


Fig. 9. Temperature slices T^* at (a) $Ra = 10^5$, (b) $Ra = 10^6$, and (c) $Ra = 10^7$.

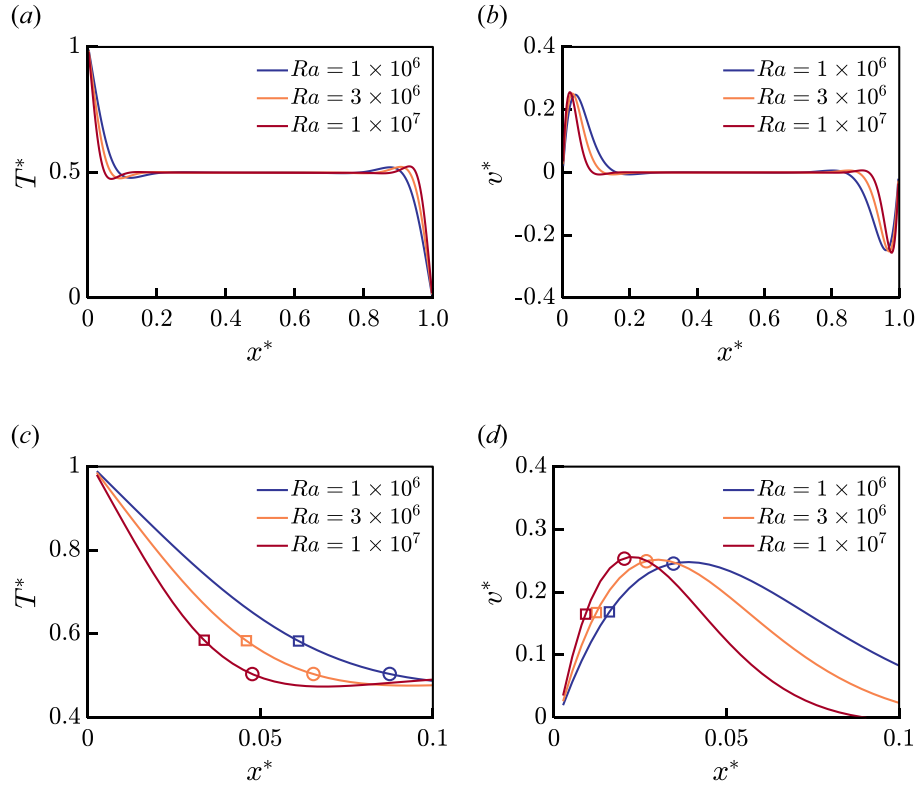


Fig. 10. Horizontal distributions of vertically averaged (a) temperature T^* and (b) vertical velocity v^* . Panels (c, d) show enlarged views near the hot wall ($x^* = 0$), where open circles denote boundary layer locations determined by the 99% criterion method, and open squares denote boundary layer locations determined by the slope method.

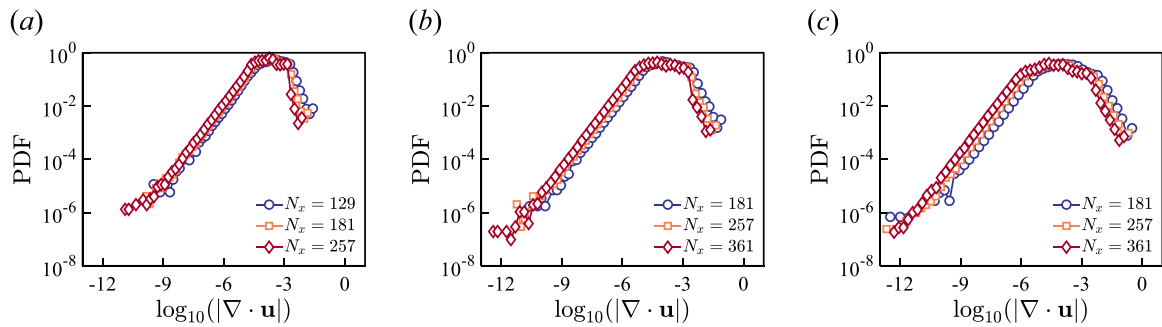


Fig. 11. PDFs of $\log_{10}(|\nabla \cdot \mathbf{u}|)$ obtained over the entire domain, for (a) $Ra = 10^5$, (b) $Ra = 10^6$, and (c) $Ra = 10^7$.

The results in Table 4 demonstrate good agreement with previous studies, with most values converge toward the corresponding asymptotic limits. To ensure systematic refinement, the non-uniform mesh

was constructed to preserve geometric similarity while reducing the effective mesh spacing. Richardson extrapolation was then applied using three successively refined meshes: for $Ra = 10^5$, the grid numbers

Table 4

Benchmark solutions of the Nusselt numbers and Reynolds number. The columns from left to right indicate the following: Rayleigh number Ra ; flow database; grid number $N_x \times N_y \times N_z$; volume-averaged Nusselt number $\langle Nu \rangle$; average Nusselt number along the hot wall Nu_{hot} ; average Nusselt number along the cold wall Nu_{cold} ; volume-averaged Reynolds number $\langle Re \rangle$.

Ra	Flow database	$N_x \times N_y \times N_z$	$\langle Nu \rangle$	Nu_{hot}	Nu_{cold}	$\langle Re \rangle$
10^5	Present	129^3	4.3406	4.3624	4.3624	39.4521
	Present	181^3	4.3363	4.3520	4.3520	39.3768
	Present	257^3	4.3346	4.3457	4.3457	39.3500
		∞	4.3334	4.3362	4.3362	39.3352
	p		2.6	1.5	1.5	3.0
	GCI (%)		0.03	0.3	0.3	0.05
	Fusegi [74]	62^3	–	4.361	–	–
	Tric [75]	81^3	–	–	4.3370	–
	Present	181^3	8.6601	8.6931	8.6931	97.4828
	Present	257^3	8.6441	8.6669	8.6668	97.0075
10^6	Present	361^3	8.6379	8.6540	8.6540	96.8464
		∞	8.6341	8.6419	8.6419	96.7638
	p		2.8	2.1	2.1	3.1
	GCI (%)		0.055	0.18	0.18	0.11
	Xu [37]	257^3	–	8.64345	8.64342	–
	Fusegi [74]	62^3	–	8.770	–	–
	Tric [75]	81^3	–	–	8.6407	–
	Present	181^3	16.5341	16.6161	16.6158	237.3925
	Present	257^3	16.4179	16.4681	16.4680	230.7625
	Present	361^3	16.3670	16.4001	16.4001	228.4148
10^7		∞	16.3274	16.3422	16.3422	227.1278
	p		2.4	2.3	2.3	3.0
	GCI (%)		0.30	0.44	0.44	0.70
	Xu [37]	257^3	–	16.40322	16.40285	–
	Tric [75]	111^3	–	–	16.3427	–

are $(N_{coarse}, N_{medium}, N_{fine}) = (129, 181, 361)$; for $Ra = 10^6$ and 10^7 , they are $(N_{coarse}, N_{medium}, N_{fine}) = (181, 257, 361)$. Throughout, we take $\Delta x_i \propto 1/N_i$, with a refinement ratio $r = \Delta x_{coarse}/\Delta x_{medium} = \Delta x_{medium}/\Delta x_{fine} \approx 1.41$. Table 4 reports the estimated convergence orders and Richardson-extrapolated asymptotic values of the Nusselt and Reynolds numbers. The included GCI values confirm that numerical uncertainty is generally small, typically below 1%. Larger uncertainties are observed for local variables, reflecting the greater difficulty of achieving convergence in these quantities. Nevertheless, these results validate the robustness of the grid-refinement study. The ISLBM solver achieves nearly third-order spatial accuracy for volume-averaged Nusselt and Reynolds numbers, owing to the error-canceling effect of domain integration. To further examine this behavior, Table 5 summarizes flow variables evaluated at the cell center. Consistent with the 2-D case, these local flow variables converge at approximately second order.

As in the 2-D case, compressibility effects are assessed by examining the divergence field in three dimensions. Fig. 11 shows the PDFs of $\log_{10}(|\nabla \cdot \mathbf{u}|)$ at $Ra = 10^5$, 10^6 , and 10^7 , evaluated at three different grid resolutions. Across all Rayleigh numbers, the PDFs peak between $|\nabla \cdot \mathbf{u}| \in [10^{-6}, 10^{-4}]$ with long tails extending toward 10^{-11} , indicating that the flow remains nearly solenoidal. With increasing resolution, the distributions collapse onto each other, demonstrating numerical convergence of the divergence statistics. At higher Rayleigh numbers, the PDFs broaden slightly and the mean divergence increases modestly, reflecting stronger thermal forcing and sharper gradients that locally amplify compressibility effects. Nevertheless, these deviations remain small in magnitude. Taken together with the 2-D results, these findings confirm that the weakly compressible LBM preserves the incompressible character of thermal convection in both two and three dimensions, even at high Rayleigh numbers.

To extend the performance assessment to three dimensions, we adopt MLUPS as the performance metric. Fig. 12 shows results for 3-D simulations at $Ra = 10^7$ and $Pr = 0.71$ on both uniform and

Table 5

Convergence behavior of flow variables at the cell center. The columns from left to right indicate the following: Rayleigh number Ra ; grid number $N_x \times N_y \times N_z$; absolute value of vorticity magnitude $|\omega_c|$; temperature T_c ; horizontal velocity u_c ; vertical velocity v_c .

Ra	Grid	$ \omega_c $	T_c	u_c	v_c
10^5	129^3	0.2565	0.49948	-2.1342×10^{-4}	1.3056×10^{-4}
	181^3	0.2560	0.49949	-2.2689×10^{-4}	1.2423×10^{-4}
	257^3	0.2557	0.49950	-2.3377×10^{-4}	1.2119×10^{-4}
	∞	0.2548	0.49950	-2.4093×10^{-4}	1.1837×10^{-4}
	p	0.9	2.0	2.0	2.1
	GCI (%)	0.5	0.002	4.0	3.0
10^6	181^3	0.1419	0.49942	-1.2268×10^{-4}	1.4722×10^{-5}
	257^3	0.1383	0.49946	-1.3361×10^{-4}	1.6437×10^{-5}
	361^3	0.1370	0.49948	-1.3933×10^{-4}	1.7106×10^{-5}
	∞	0.1363	0.49949	-1.4563×10^{-4}	1.7532×10^{-5}
	p	3.0	2.1	1.9	2.1
	GCI (%)	0.63	0.0039	5.7	3.1
10^7	181^3	0.1296	0.49898	-7.1568×10^{-5}	6.1396×10^{-6}
	257^3	0.1149	0.49917	-7.3572×10^{-5}	7.4064×10^{-6}
	361^3	0.1094	0.49926	-7.8387×10^{-5}	7.7847×10^{-6}
	∞	0.1060	0.49934	–	7.9458×10^{-6}
	p	2.8	2.3	–	3.5
	GCI (%)	3.9	0.019	–	2.6

non-uniform meshes executed on AMD EPYC 9135 CPUs. For non-uniform meshes, the stretching coefficient $a = 2.1$. As seen in Figs. 12(a) and 12(b), the absolute MLUPS values reveal that the in-house LBM solver consistently sustains throughput at $O(10)$ MLUPS across all tested grid sizes, while Nek5000 and OpenFOAM are two to three orders of magnitude slower and exhibit decreasing performance as the grid size increases. On the largest grids, the throughput of Nek5000 and OpenFOAM drops to near $O(10^{-1})$ MLUPS, whereas the in-house LBM solver maintains nearly constant performance, demonstrating resilience to increasing problem size. The normalized MLUPS results in Figs. 12(c) and 12(d) further highlight the relative efficiency of the three solvers: Nek5000 and OpenFOAM achieve at most a few percent of the LBM performance, with the performance gap widening at higher resolutions, particularly on non-uniform meshes. By contrast, the relative advantage of the LBM solver remains robust or even improves as the grid size increases.

Figs. 13(a) and 13(b) further show the absolute WCTpDT values as functions of grid number on uniform and non-uniform meshes, respectively. On uniform meshes, the in-house LBM solver consistently achieves shorter runtimes than Nek5000, while OpenFOAM is slightly more efficient at the coarsest grid but quickly loses this advantage as the resolution increases, eventually becoming the most expensive solver. On non-uniform meshes, OpenFOAM again performs best at the smallest grid, but its cost grows steeply with resolution, surpassing both LBM and Nek5000 by more than an order of magnitude at the largest grids. Nek5000, by contrast, scales more smoothly and maintains a modest runtime advantage over LBM across all non-uniform grid sizes. The normalized results in Figs. 13(c) and 13(d) further highlight these trends. On uniform meshes, Nek5000 requires approximately 3–5 times the runtime of LBM per unit dimensionless time, while OpenFOAM starts faster at coarse grids but becomes significantly more expensive at higher resolutions. On non-uniform meshes, Nek5000 consistently outperforms LBM, requiring only about 50%–80% of its runtime, whereas OpenFOAM undergoes a sharp transition from being the fastest solver at small grids to the least efficient at the large grids, exceeding LBM's cost by a factor of 5–6. Although Nek5000 and OpenFOAM employ larger CFL numbers (0.5 and 0.9, respectively) than the LBM solver, which should in principle reduce their cost per simulated physical time, their substantially higher per-step computational overhead offsets this benefit. Consequently, WCTpDT values for both remain larger, particularly at high resolutions. As the grid size increases, the efficiency

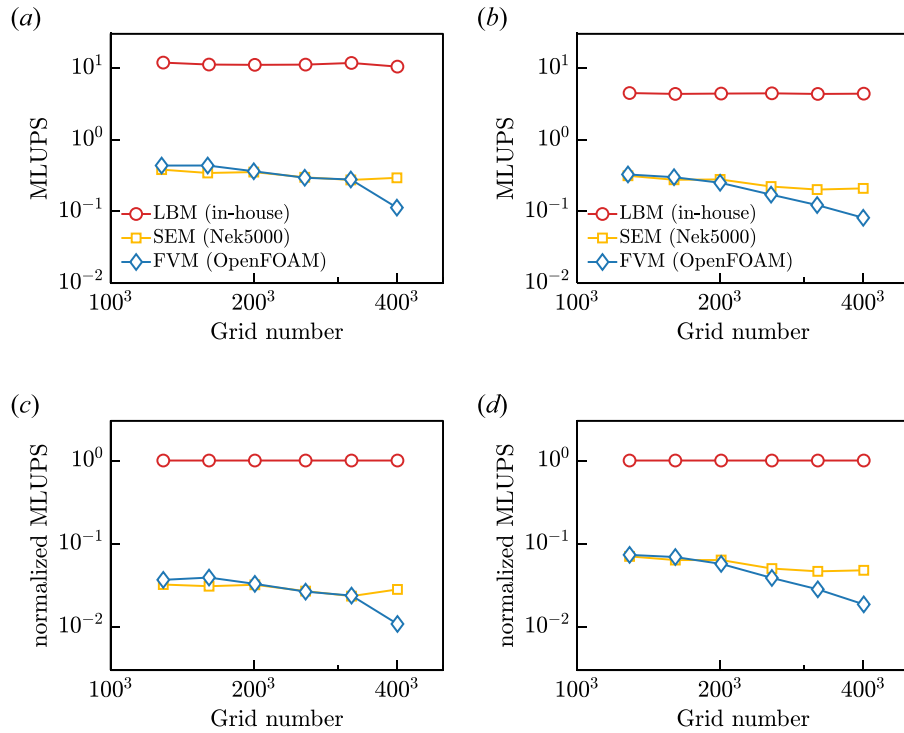


Fig. 12. Performance comparison of in-house LBM solver, Nek5000 and OpenFOAM in terms of MLUPS. (a, b) Absolute MLUPS as a function of grid number, (c, d) MLUPS normalized by the corresponding LBM solver, on (a, c) uniform and (b, d) non-uniform meshes.

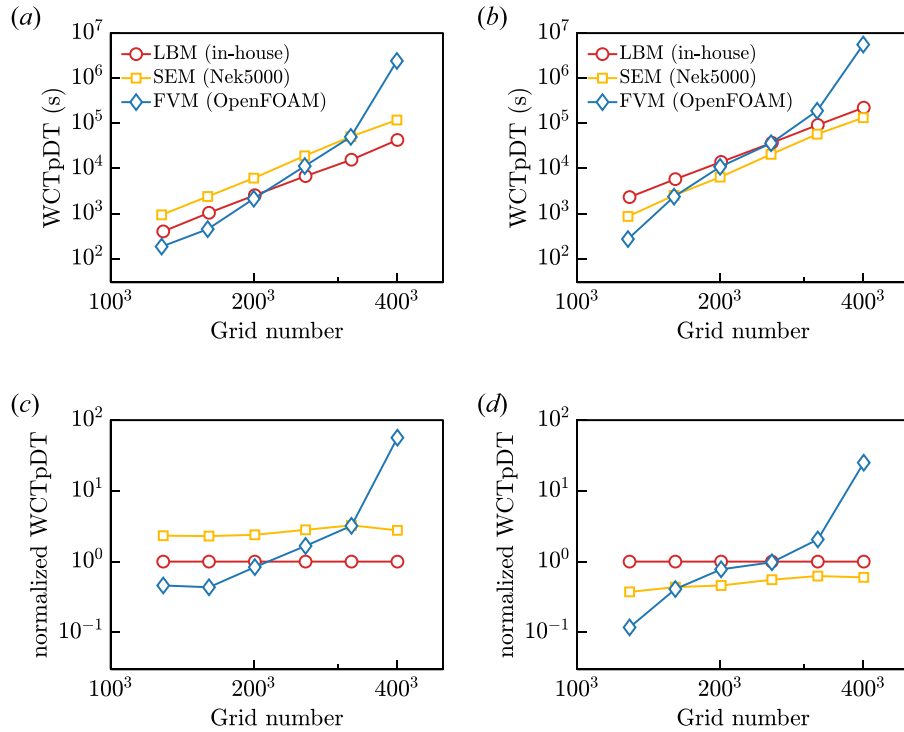


Fig. 13. Performance comparison of in-house LBM solver, Nek5000 and OpenFOAM solvers in terms of WCTpDDT. (a, b) Absolute WCTpDDT as a function of grid number, (c, d) the WCTpDDT normalized by the corresponding LBM solver, on (a, c) uniform and (b, d) non-uniform meshes.

gap widens further, establishing the LBM solver as a favorable choice for advancing large-scale 3-D simulations.

To further assess the computational efficiency of the in-house 3-D LBM solver on modern GPU architecture, we performed additional tests on an NVIDIA A100 GPU. Fig. 14 compares the solver's performance

for uniform and non-uniform meshes. As shown in Fig. 14(a), the MLUPS values on uniform meshes consistently exceed those on non-uniform meshes across all tested grid sizes. The normalized MLUPS in Fig. 14(b) indicates that non-uniform grids achieve only about 60%–65% of the performance of uniform grids, with a slight downward

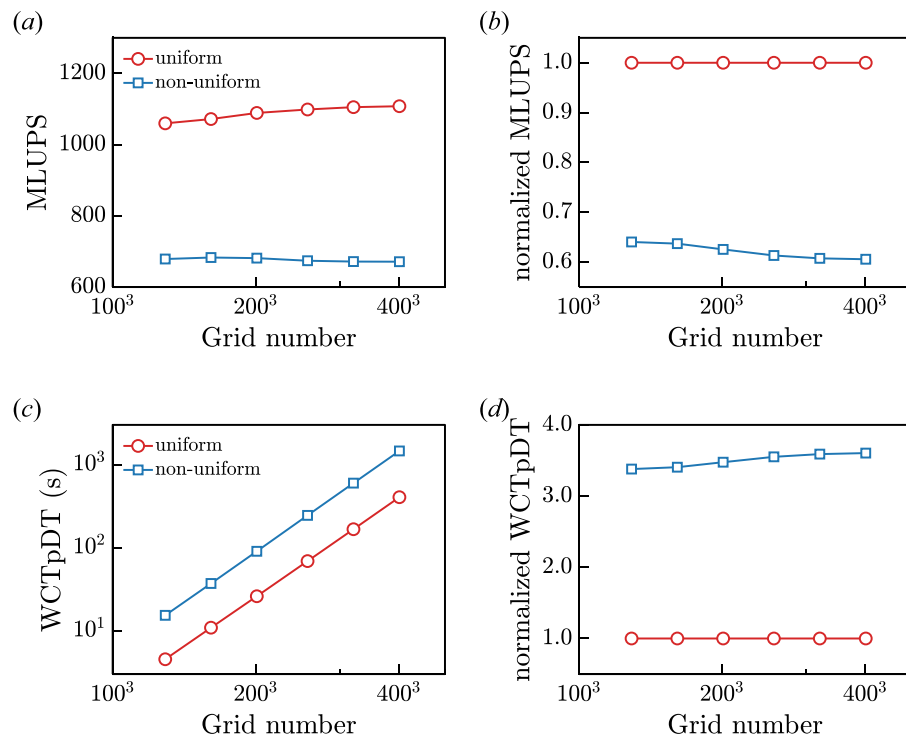


Fig. 14. GPU performance of the in-house LBM solver on an NVIDIA A100 for uniform and non-uniform meshes: (a) Absolute MLUPS as a function of grid number, (b) MLUPS normalized by the uniform-mesh performance at each grid; (c) WCTpDT as a function of grid number; (d) WCTpDT normalized by the corresponding value for uniform meshes.

trend as resolution increases. Fig. 14(c) shows that the WCTpDT grows nearly linearly with grid size for both mesh types, while Fig. 14(d) confirms that the relative overhead of non-uniform meshes remains close to a factor of three across resolutions. Compared with the 2-D case, the relative throughput loss is similar, but the absolute runtime penalty is more pronounced in 3-D, reflecting the higher computational and memory demands of large-scale simulations. Overall, the solver retains excellent scalability and GPU efficiency in three dimensions, ensuring its applicability to practical large-scale problems even when non-uniform discretizations are employed.

For 3-D simulations, the cost estimation is as follows. Consider a simulation at $Ra = 10^7$ and $Pr = 0.71$ using a 400^3 non-uniform mesh with a stretching coefficient of $a = 2.1$. On an NVIDIA A100 GPU, the wall-clock time required to advance the simulation by $1 t_f$ (i.e. WCTpDT) is 1.456×10^3 s ≈ 0.40 h. Thus, the corresponding cost per t_f is $0.40 \text{ h} \times 7 \text{ RMB/h} \approx 2.83 \text{ RMB}$. For Nek5000, the time per t_f is about 1.33484×10^5 s ≈ 37.08 h, yielding a cost of $37.08 \text{ h} \times 0.1 \text{ RMB/h} \approx 3.71 \text{ RMB}$. For OpenFOAM, the time per t_f is about 5.576363×10^6 s ≈ 1549.0 h, yielding $1549.0 \text{ h} \times 0.1 \text{ RMB/h} \approx 154.90 \text{ RMB}$. When Nek5000 and OpenFOAM run on multi-core CPU architectures, scalability is limited by communication overhead and memory contention, and their effective performance falls short of ideal expectations, resulting in increased wall-clock times and overall computational cost. We expect greater economic advantage of the GPU-based LBM solver in large-scale simulations.

5. Conclusion

In this study, we systematically evaluated an interpolation-supplemented lattice Boltzmann method (ISLBM) for simulating buoyancy-driven thermal convection on non-uniform meshes. The ISLBM enables local mesh refinement near solid boundaries to resolve steep thermal and velocity gradients, while retaining the computational efficiency of the standard LBM in the bulk flow. By incorporating quadratic interpolation during the streaming step, the ISLBM achieves

nearly third-order accuracy for global quantities and about second-order for local fields. The method was validated through benchmark simulations of laminar thermal convection in a side-heated cavity at Rayleigh numbers $10^6 \leq Ra \leq 10^8$ in 2-D and $10^5 \leq Ra \leq 10^7$ in 3-D, demonstrating excellent agreement with high-fidelity reference data. The results demonstrate second-order spatial accuracy for local temperature and velocity fields, and nearly third-order accuracy for vorticity and volume-integrated quantities. Furthermore, a careful assessment of compressibility effects confirmed that the solver remains within a nearly incompressible regime, with only minimal numerical divergence even at elevated Rayleigh numbers.

We also assessed the computational performance of the in-house LBM solver by benchmarking it against two widely used open-source solvers: Nek5000 based on the spectral element method, and OpenFOAM based on the finite volume method. Using performance metrics such as MLUPS and WCTpDT, our results consistently demonstrate that the ISLBM outperforms these solvers in large-scale simulations. When deployed on GPU architecture, the ISLBM maintains high computational performance. In terms of MLUPS, the throughput on non-uniform meshes reaches approximately 60%–70% of that achieved on uniform meshes; in terms of WCTpDT, however, the computational cost on non-uniform meshes is about three times higher than that on uniform meshes, mainly due to the overhead of quadratic interpolation and irregular memory access patterns. Despite this penalty, the solver exhibits excellent scalability and GPU efficiency, enabling practical large-scale simulations on contemporary accelerator hardware.

While the ISLBM was originally proposed by He et al. [26–28], to the best of our knowledge no prior work has conducted a systematic and quantitative evaluation of its performance for thermally driven flows at high Rayleigh numbers on non-uniform meshes. This study is, to our knowledge, the first to: (i) implement quadratic interpolation-based streaming to simulate thermal convection on non-uniform meshes at high Rayleigh numbers; (ii) quantitatively evaluate accuracy, convergence order, divergence control, and computational efficiency; and (iii)

benchmark the ISLBM against high-fidelity solvers (Nek5000 and OpenFOAM) using rigorous MLUPS and WCTpDT metrics, including GPU-based performance tests. Our results demonstrate that the interpolation-supplemented LBM provides a robust and efficient framework for simulating thermally driven flows on non-uniform meshes. Its demonstrated accuracy, stability, computational efficiency, and scalability make it a promising candidate for future extensions to fully turbulent convection at extreme Rayleigh numbers.

CRedit authorship contribution statement

Ao Xu: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Zheng Zhao:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Ben-Rui Xu:** Validation, Software, Investigation, Data curation. **Li-Sheng Jiang:** Validation, Software, Investigation, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Given his role as Associate Editor and Managing Editor of the International Journal of Heat and Mass Transfer, Ao Xu had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) through grants nos. 12272311, 12388101; the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001). The authors acknowledge the Beijing Beilong Super Cloud Computing Co., Ltd for providing HPC resources that have contributed to the research results reported within this paper (URL: <http://www.blsc.cn/>).

Data availability

Data will be made available on request.

References

- [1] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1) (1998) 329–364, <http://dx.doi.org/10.1146/annurev.fluid.30.1.329>.
- [2] C.K. Aidun, J.R. Clausen, Lattice-Boltzmann method for complex flows, *Annu. Rev. Fluid Mech.* 42 (1) (2010) 439–472, <http://dx.doi.org/10.1146/annurev.fluid-121108-145519>.
- [3] K. Xu, Direct modeling for computational fluid dynamics, *Acta Mech. Sin.* 31 (3) (2015) 303–318, <http://dx.doi.org/10.1007/s10409-015-0453-2>.
- [4] K. Xu, *Direct Modeling for Computational Fluid Dynamics*, vol. 4, World Scientific, Singapore, 2015.
- [5] Y.H. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier–Stokes equation, *Europhys. Lett.* 17 (6) (1992) 479–484, <http://dx.doi.org/10.1209/0295-5075/17/6/001>.
- [6] H. Chen, S. Chen, W.H. Matthaeus, Recovery of the Navier–Stokes equations using a lattice-gas Boltzmann method, *Phys. Rev. A* 45 (8) (1992) R5339–R5342, <http://dx.doi.org/10.1103/PhysRevA.45.R5339>.
- [7] X. He, L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E* 56 (6) (1997) 6811–6817, <http://dx.doi.org/10.1103/PhysRevE.56.6811>.
- [8] A. Xu, L. Shi, T.S. Zhao, Accelerated lattice Boltzmann simulation using GPU and OpenACC with data management, *Int. J. Heat Mass Transfer* 109 (2017) 577–588, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2017.02.032>.
- [9] A. Xu, B.T. Li, Multi-GPU thermal lattice Boltzmann simulations using OpenACC and MPI, *Int. J. Heat Mass Transfer* 201 (2023) 123649, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.123649>.
- [10] A. Xu, B.T. Li, Particle-resolved thermal lattice Boltzmann simulation using OpenACC on multi-GPUs, *Int. J. Heat Mass Transfer* 218 (2024) 124758, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2023.124758>.
- [11] L. Chen, Q. Kang, Y. Mu, Y. He, W. Tao, A critical review of the pseudo-potential multiphase lattice Boltzmann model: Methods and applications, *Int. J. Heat Mass Transfer* 76 (2014) 210–236, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2014.04.032>.
- [12] Q. Li, K.H. Luo, Q. Kang, Y. He, Q. Chen, Q. Liu, Lattice Boltzmann methods for multiphase flow and phase-change heat transfer, *Prog. Energy Combust. Sci.* 52 (2016) 62–105, <http://dx.doi.org/10.1016/j.pecs.2015.10.001>.
- [13] E. Calzavarini, Eulerian–Lagrangian fluid dynamics platform: the ch4-project, *Softw. Impacts* 1 (2019) 100002, <http://dx.doi.org/10.1016/j.simpa.2019.100002>.
- [14] V. Patočka, E. Calzavarini, N. Tosi, Settling of inertial particles in turbulent Rayleigh–Bénard convection, *Phys. Rev. Fluids* 5 (11) (2020) 114304, <http://dx.doi.org/10.1103/PhysRevFluids.5.114304>.
- [15] V. Patočka, N. Tosi, E. Calzavarini, Residence time of inertial particles in 3D thermal convection: Implications for magma reservoirs, *Earth Planet. Sci. Lett.* 591 (2022) 117622, <http://dx.doi.org/10.1016/j.epsl.2022.117622>.
- [16] H. Wu, K. Karzhaubayev, J. Shen, L.-P. Wang, Direct numerical simulation of three-dimensional particle-laden thermal convection using the lattice Boltzmann method, *Comput. & Fluids* 276 (2024) 106268, <http://dx.doi.org/10.1016/j.compfluid.2024.106268>.
- [17] Y. Zhang, K. Luo, H. Yi, A. Liu, J. Wu, Application of large eddy simulation models to electroconvection turbulence study with lattice Boltzmann method, *Phys. Rev. Fluids* 9 (8) (2024) 083703, <http://dx.doi.org/10.1103/PhysRevFluids.9.083703>.
- [18] X. Liu, Z.-X. Tong, Y.-L. He, Insight into enthalpy-based lattice Boltzmann method for solid-liquid phase change without numerical diffusion, *Int. J. Heat Mass Transfer* 239 (2025) 126534, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2024.126534>.
- [19] X. Liu, Z.-X. Tong, Y.-L. He, K.H. Luo, S. Du, Multi-GPU accelerated lattice Boltzmann modelling of melting performance for phase change material-metal foam composite considering interfacial thermal resistance, *Int. J. Heat Mass Transfer* 252 (2025) 127475, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2025.127475>.
- [20] K.-Q. Xia, Current trends and future directions in turbulent thermal convection, *Theor. Appl. Mech. Lett.* 3 (5) (2013) 052001, <http://dx.doi.org/10.1063/2.1305201>.
- [21] K.-Q. Xia, S.-D. Huang, Y.-C. Xie, L. Zhang, Tuning heat transport via coherent structure manipulation: recent advances in thermal turbulence, *Natl. Sci. Rev.* 10 (6) (2023) nwad012, <http://dx.doi.org/10.1093/nsr/nwad012>.
- [22] K.-Q. Xia, K.L. Chong, G.-Y. Ding, L. Zhang, Some fundamental issues in buoyancy-driven flows with implications for geophysical and astrophysical systems, *Acta Mech. Sin.* 41 (1) (2024) 324287, <http://dx.doi.org/10.1007/s10409-024-24287-x>.
- [23] A. Xu, W. Shyy, T.S. Zhao, Lattice Boltzmann modeling of transport phenomena in fuel cells and flow batteries, *Acta Mech. Sin.* 33 (3) (2017) 555–574, <http://dx.doi.org/10.1007/s10409-017-0667-6>.
- [24] M.-J. Ni, J.C. Yang, J. Zhang, Magnetohydrodynamics in liquid metal interfacial flows, *Appl. Mech. Rev.* (2025) 1–105, <http://dx.doi.org/10.1115/1.4067935>.
- [25] B.-F. Wang, Q. Zhou, C. Sun, Vibration-induced boundary-layer destabilization achieves massive heat-transport enhancement, *Sci. Adv.* 6 (21) (2020) eaaz8239, <http://dx.doi.org/10.1126/sciadv.aaz8239>.
- [26] X. He, L.S. Luo, M. Dembo, Some progress in lattice Boltzmann method. Part I. Nonuniform mesh grids, *J. Comput. Phys.* 129 (2) (1996) 357–363, <http://dx.doi.org/10.1006/jcph.1996.0255>.
- [27] X. He, L.S. Luo, M. Dembo, Some progress in the lattice Boltzmann method: Reynolds number enhancement in simulations, *Physica A* 239 (1–3) (1997) 276–285, [http://dx.doi.org/10.1016/S0378-4371\(96\)00486-4](http://dx.doi.org/10.1016/S0378-4371(96)00486-4).
- [28] X. He, Error analysis for the interpolation-supplemented lattice-Boltzmann equation scheme, *Internat. J. Modern Phys. C* 8 (04) (1997) 737–745, <http://dx.doi.org/10.1142/S012918319700062X>.
- [29] F. Nannelli, S. Succi, The lattice Boltzmann equation on irregular lattices, *J. Stat. Phys.* 68 (3) (1992) 401–407, <http://dx.doi.org/10.1007/BF01341755>.
- [30] G. Peng, H. Xi, C. Duncan, S.-H. Chou, Finite volume scheme for the lattice Boltzmann method on unstructured meshes, *Phys. Rev. E* 59 (4) (1999) 4675–4682, <http://dx.doi.org/10.1103/PhysRevE.59.4675>.
- [31] H. Xi, G. Peng, S.-H. Chou, Finite-volume lattice Boltzmann method, *Phys. Rev. E* 59 (5) (1999) 6202–6205, <http://dx.doi.org/10.1103/PhysRevE.59.6202>.
- [32] S. Ubertini, G. Bella, S. Succi, Lattice Boltzmann method on unstructured grids: Further developments, *Phys. Rev. E* 68 (1) (2003) 016701, <http://dx.doi.org/10.1103/PhysRevE.68.016701>.

- [33] O. Filippova, D. Hänel, Grid refinement for lattice-BGK models, *J. Comput. Phys.* 147 (1) (1998) 219–228, <http://dx.doi.org/10.1006/jcph.1998.6089>.
- [34] D. Yu, R. Mei, W. Shyy, A multi-block lattice Boltzmann method for viscous fluid flows, *Internat. J. Numer. Methods Fluids* 39 (2) (2002) 99–120, <http://dx.doi.org/10.1002/flid.280>.
- [35] J. Tölke, M. Krafczyk, E. Rank, A multigrid-solver for the discrete Boltzmann equation, *J. Stat. Phys.* 107 (1) (2002) 573–591, <http://dx.doi.org/10.1023/A:1014551813787>.
- [36] D.J. Mavriplis, Multigrid solution of the steady-state lattice Boltzmann equation, *Comput. & Fluids* 35 (8–9) (2006) 793–804, <http://dx.doi.org/10.1016/j.compfluid.2005.07.020>.
- [37] A. Xu, L. Shi, H.D. Xi, Lattice Boltzmann simulations of three-dimensional thermal convective flows at high Rayleigh number, *Int. J. Heat Mass Transfer* 140 (2019) 359–370, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2019.06.002>.
- [38] J. Wang, D. Wang, P. Lallemand, L.S. Luo, Lattice Boltzmann simulations of thermal convective flows in two dimensions, *Comput. Math. Appl.* 65 (2) (2013) 262–286, <http://dx.doi.org/10.1016/j.camwa.2012.07.001>.
- [39] D. Contrino, P. Lallemand, P. Asinari, L.S. Luo, Lattice-Boltzmann simulations of the thermally driven 2D square cavity at high Rayleigh numbers, *J. Comput. Phys.* 275 (2014) 257–272, <http://dx.doi.org/10.1016/j.jcp.2014.06.047>.
- [40] Z. Chai, T.S. Zhao, Lattice Boltzmann model for the convection-diffusion equation, *Phys. Rev. E* 87 (6) (2013) 063309, <http://dx.doi.org/10.1103/PhysRevE.87.063309>.
- [41] Z. Chai, B. Shi, Multiple-relaxation-time lattice Boltzmann method for the Navier–Stokes and nonlinear convection-diffusion equations: Modeling, analysis, and elements, *Phys. Rev. E* 102 (2) (2020) 023306, <http://dx.doi.org/10.1103/PhysRevE.102.023306>.
- [42] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *Phys. Rev. E* 65 (4) (2002) 046308, <http://dx.doi.org/10.1103/PhysRevE.65.046308>.
- [43] Z. Guo, C. Zheng, Analysis of lattice Boltzmann equation for microscale gas flows: Relaxation times, boundary conditions and the Knudsen layer, *Int. J. Comput. Fluid Dyn.* 22 (7) (2008) 465–473, <http://dx.doi.org/10.1080/10618560802253100>.
- [44] H. Huang, M.C. Sukop, X.-Y. Lu, *Multiphase Lattice Boltzmann Methods: Theory and Application*, John Wiley & Sons, 2015.
- [45] A. Ceci, S. Pirozzoli, Natural grid stretching for DNS of compressible wall-bounded flows, *J. Comput. Phys.: X* 17 (2022) 100128, <http://dx.doi.org/10.1016/j.jcp.2023.100128>.
- [46] G. de Vahl Davis, Natural convection of air in a square cavity: A benchmark numerical solution, *Internat. J. Numer. Methods Fluids* 3 (3) (1983) 249–264, <http://dx.doi.org/10.1002/flid.1650030305>.
- [47] D.L. Sun, W.Q. Tao, Performance analysis and comparison of ideal and simpler algorithms for incompressible fluid flow and heat transfer problems, *Int. J. Heat Mass Transfer* 205 (2023) 123896, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2023.123896>.
- [48] Q. Chen, Y. Xuan, Q. Lin, Y. Han, Evolutions of the electromagnetic signatures induced by the propagating wake behind a submerged body, *Int. J. Heat Mass Transfer* 194 (2022) 123105, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.123105>.
- [49] J.E. Vesper, S.C. Tietjen, M. Chakkingal, S. Kenjereš, Numerical analysis of effects of fins and conductive walls on heat transfer in side heated cavities—Onset of three-dimensional phenomena in natural convection, *Int. J. Heat Mass Transfer* 183 (2022) 122033, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2021.122033>.
- [50] Y. Ma, Z. Yang, Simplified and highly stable thermal lattice Boltzmann method simulation of hybrid nanofluid thermal convection at high Rayleigh numbers, *Phys. Fluids* 32 (1) (2020) 012009, <http://dx.doi.org/10.1063/1.5139092>.
- [51] P. Yang, C. Huang, Z. Zhang, T. Long, M. Liu, Simulating natural convection with high Rayleigh numbers using the smoothed particle hydrodynamics method, *Int. J. Heat Mass Transfer* 166 (2021) 120758, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2020.120758>.
- [52] Z. Zhao, M. Wen, W. Li, A coupled gas-kinetic BGK scheme for the finite volume lattice Boltzmann method for nearly incompressible thermal flows, *Int. J. Heat Mass Transfer* 164 (2021) 120584, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2020.120584>.
- [53] C. Ma, J. Wu, L. Yang, H. Dong, A coupled high-order implicit-explicit flux reconstruction lattice Boltzmann method for nearly incompressible thermal flows, *Int. J. Heat Mass Transfer* 187 (2022) 122575, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.122575>.
- [54] C. Ma, J. Wu, X. Gu, L. Yang, High-order flux reconstruction thermal lattice Boltzmann flux solver for simulation of incompressible thermal flows, *Phys. Rev. E* 106 (3) (2022) 035301, <http://dx.doi.org/10.1103/PhysRevE.106.035301>.
- [55] Q. Xu, R. Li, M. Xu, High-performance implementation of parallel semi-implicit method for pressure linked equations solver on CPU+GPU platform, *Int. J. Heat Mass Transfer* 182 (2022) 121976, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2021.121976>.
- [56] J. Lu, C. Dai, P. Yu, Analysis and reconstruction of the simplified thermal lattice Boltzmann method, *Int. J. Heat Mass Transfer* 187 (2022) 122576, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.122576>.
- [57] A. Nee, Evaluation of hybrid lattice Boltzmann models for laminar and turbulent natural convection, *Int. Commun. Heat Mass Transfer* 162 (2025) 108635, <http://dx.doi.org/10.1016/j.icheatmasstransfer.2025.108635>.
- [58] X. Feng, J. Zeng, Y. Ge, F. Liu, T. Yu, Q. Liu, An improved multiple-relaxation-time lattice Boltzmann model for incompressible thermal flows in two dimensions, *Int. J. Heat Mass Transfer* 240 (2025) 126614, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2024.126614>.
- [59] Q. Wang, S.N. Xia, R. Yan, D.J. Sun, Z.H. Wan, Non-Oberbeck-Boussinesq effects due to large temperature differences in a differentially heated square cavity filled with air, *Int. J. Heat Mass Transfer* 128 (2019) 479–491, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2018.06.079>.
- [60] P.L. Quéré, Accurate solutions to the square thermally driven cavity at high Rayleigh number, *Comput. & Fluids* 20 (1) (1991) 29–41, [http://dx.doi.org/10.1016/0045-7930\(91\)90025-D](http://dx.doi.org/10.1016/0045-7930(91)90025-D).
- [61] Z. Chai, T.S. Zhao, Effect of the forcing term in the multiple-relaxation-time lattice Boltzmann equation on the shear stress or the strain rate tensor, *Phys. Rev. E* 86 (1) (2012) 016705, <http://dx.doi.org/10.1103/PhysRevE.86.016705>.
- [62] X. Wen, L.-P. Wang, Z. Guo, An efficient discrete unified gas-kinetic scheme for compressible thermal flows, *Phys. Fluids* 36 (2) (2024) 026123, <http://dx.doi.org/10.1063/5.0188092>.
- [63] X. Wen, W. Lin, W. Wang, Y. Chen, K. Li, L.-P. Wang, An immersed boundary-discrete unified gas-kinetic scheme for Non-Oberbeck–Boussinesq thermal convection with curved surfaces, *Int. J. Heat Mass Transfer* 247 (2025) 127100, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2025.127100>.
- [64] R. Ostilla-Mónico, Y. Yang, E.P.V.D. Poel, D. Lohse, R. Verzicco, A multiple-resolution strategy for direct numerical simulation of scalar turbulence, *J. Comput. Phys.* 301 (2015) 308–321, <http://dx.doi.org/10.1016/j.jcp.2015.08.031>.
- [65] I.B. Celik, U. Ghia, P.J. Roache, C.J. Freitas, H. Coleman, P.E. Raad, Procedure for estimation and reporting of uncertainty due to discretization in CFD applications, *J. Fluids Eng.* 130 (7) (2008) 078001, <http://dx.doi.org/10.1115/1.2960953>.
- [66] P. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations, *J. Comput. Phys.* 133 (1) (1997) 84–101, <http://dx.doi.org/10.1006/jcph.1997.5651>.
- [67] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Comput. Phys.* 12 (6) (1998) 620–631, <http://dx.doi.org/10.1063/1.168744>.
- [68] B. Ren, C.-G. Li, M. Tsubokura, Laminar natural convection in a square cavity with 3D random roughness elements considering the compressibility of the fluid, *Int. J. Heat Mass Transfer* 173 (2021) 121248, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2021.121248>.
- [69] B. Ren, C.-G. Li, M. Tsubokura, The effects of irregular roughness with different surface power spectrums on the heat transfer of natural convection in enclosures, *Int. Commun. Heat Mass Transfer* 141 (2023) 106581, <http://dx.doi.org/10.1016/j.icheatmasstransfer.2022.106581>.
- [70] S. Yigit, J. Hasslberger, N. Chakraborty, M. Klein, Effects of Rayleigh–Bénard convection on spectra of viscoplastic fluids, *Int. J. Heat Mass Transfer* 147 (2020) 118947, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2019.118947>.
- [71] Y. Su, T. Ng, J.H. Davidson, Three dimensional mesoscopic scale simulations of buoyancy driven flow and heat mass transfer through randomly packed fiber boards, *Int. J. Heat Mass Transfer* 143 (2019) 118522, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2019.118522>.
- [72] C. Ma, J. Wu, X. Gu, L. Yang, A three-dimensional high-order flux reconstruction lattice Boltzmann flux solver for incompressible laminar and turbulent flows, *Comput. & Fluids* 265 (2023) 106008, <http://dx.doi.org/10.1016/j.compfluid.2023.106008>.
- [73] W. Schupbach, K. Premnath, Fokker–Planck model-based central moment lattice Boltzmann method for effective simulations of thermal convective flows, *Energies* 18 (5) (2025) 1890, <http://dx.doi.org/10.3390/en18081890>.
- [74] T. Fusegi, J.M. Hyun, K. Kuwahara, B. Farouk, A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure, *Int. J. Heat Mass Transfer* 34 (6) (1991) 1543–1557, [http://dx.doi.org/10.1016/0017-9310\(91\)90295-P](http://dx.doi.org/10.1016/0017-9310(91)90295-P).
- [75] E. Tric, G. Labrosse, M. Betrouni, A first incursion into the 3D structure of natural convection of air in a differentially heated cubic cavity, from accurate numerical solutions, *Int. J. Heat Mass Transfer* 43 (21) (2000) 4043–4056, [http://dx.doi.org/10.1016/S0017-9310\(00\)00037-5](http://dx.doi.org/10.1016/S0017-9310(00)00037-5).