PARALLEL DIRECT METHOD OF DNS FOR TWO-DIMENSIONAL TURBULENT RAYLEIGH-BÉNARD CONVECTION

Y. Bao* **J. Luo** **M. Ye**

Department of Mechanics
Sun Yat-sen University
Guangzhou, China

ABSTRACT

A highly efficient parallelization scheme of direct numerical simulation (DNS) for two-dimensional Rayleigh-Bénard convection is presented. By introducing the parallel diagonal dominant (PDD) algorithm to solve the pressure Poisson equation and adjusting the domain decomposition accordingly, all-to-all communication as the usual obstacle to parallel computing can be eliminated. Excellent strong scaling and weak scaling for the parallel efficiency are achieved. Numerical results show that very complex structures in flow exist at very high *Ra* numbers. The required high resolution both in space and in time can be obtained by the present method at low parallel overhead.

Keywords: PDM-DNS, Parallel computation, Rayleigh-Bénard convection, Turbulence.

1. INTRODUCTION

Thermal convection can be widely seen in nature and industry. Understanding of thermal convection holds the key to global climate change, ocean circulation, reactor design and industry cooling [1]. Rayleigh-Bénard convection (RBC) is a paradigmatic thermal convection problem where a fluid container with two horizontal plates is heated from below and cooled from above. The driving of the system is characterized by the dimensionless Rayleigh number *Ra*, which is proportional to the cubic of height and the temperature difference between the top and bottom plates. When the driving is high enough, i.e., *Ra* exceeds 10^7, the flow develops into turbulence [2].

The main interest in turbulent RBC is to understand how the heat transfer, expressed by the dimensionless Nusselt number *Nu*, scales with the Rayleigh number. The *Nu*-*Ra* scaling is well predicted by the GL theory [3-6] up to a critical Rayleigh number Ra^*, above which the system would theoretically enter the "ultimate regime" [7]. Due to the difficulties in achieving high enough Rayleigh numbers, only a few experiments cover the "ultimate regime" and discrepancies exist between these experiments [8].

Direct numerical simulation (DNS) plays an important role in the investigation of RBC. It has the advantages of fully accessibility to variables, exact satisfaction to boundary conditions and arbitrarily chosen parameters. However, computational work grows exponentially with the Rayleigh number and well-resolved RBC of very high *Ra* remains a challenging task for simulations.

In the last decade, numerical simulations of RBC commonly adopted cylindrical configurations [9, 10] to replicate the set-ups of most experiments [11]. However, the singularity at the axis of cylindrical coordinates requires special treatments, e.g., implicit scheme in azimuthal direction [10], which unavoidably affect the overall computational performance, and especially the parallelization of the code.

The dynamics of turbulent convection in a cylinder are known to be much different from that in a rectangular box [12, 13]. Many investigations were done towards the latter case in the recent years. Chong *et al*. [14, 15] show that rectangular box with small aspect-ratio could cause plume condensation which results in heat transport enhancement. The scaling of *Nu*-*Ra* for small aspect-ratio also behaves differently [16].

Moreover, the potential of simple geometry to simulate RBC at lower computational cost is also desirable. Ostilla-Monico *et al*. [17] proposed a multiple-resolution strategy, which can be readily applied to rectangular RBC. Most recently, a pencil distributed finite difference code [18] was presented as a scalable parallelization scheme to simulate rectangular RBC. In their code, the viscous terms in vertical direction are integrated implicitly, and the pencils are aligned with the same direction to reduce all-to-all communications. Even so, it needs 6 instances of all-to-all communications per timestep.

In this paper, we present a highly efficient method dedicated to high resolution simulation of two-dimensional rectangular RBC. Extension to

* Corresponding author (stsby@mail.sysu.edu.cn)

DOI : 10.1017/jmech.2017.54
Copyright © 2017 The Society of Theoretical and Applied Mechanics
https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

three-dimensional simulations is straightforward. Section 2 describes the numerical scheme in detail and the choice of domain decomposition suitable for the parallel diagonal dominant (PDD) algorithm. Section 3 gives the strong scaling and weak scaling, as the computation performance of the code. Section 4 shows the numerical results produced by the present method and argue for high resolution in simulation of Rayleigh-Bénard convection.

2. NUMERICAL SCHEME

The incompressible Navier-Stokes equations coupled with a temperature equation are solved in a two-dimensional rectangular domain, with temperature being isothermal at two horizontal plates and adiabatic at sidewalls. All plates and sidewalls are subjected to no-slip condition. Under the Oberbeck-Boussinesq approximation, the non-dimensional Navier-Stokes equations for RBC read

$$\begin{cases} \nabla\cdot\boldsymbol{v}=0 \\ \dfrac{\partial v}{\partial t}+(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}=-\nabla p+\dfrac{1}{\sqrt{Ra/\mathrm{Pr}}}\nabla^2\boldsymbol{v}+\theta\vec{\kappa} \\ \dfrac{\partial\theta}{\partial t}+(\boldsymbol{v}\cdot\nabla)\theta=\dfrac{1}{\sqrt{Ra\cdot\mathrm{Pr}}}\nabla^2\theta \end{cases} \tag{1}$$

where $\boldsymbol{v}$, p and θ are the non-dimensional velocity, pressure and temperature, respectively, and t is the non-dimensional time. $\vec{\kappa}$ is the unitary vector parallel and opposite to gravity. Non-dimensionalization is done by choosing the temperature difference between two horizontal plates ΔT as the characteristic temperature, the height of cell H as the characteristic length and the free-fall velocity $U_f=\sqrt{g\beta\Delta TH}$ as the characteristic velocity, where g is gravity and β the isobaric expansion coefficient of the fluid. The Prandtl number $\mathrm{Pr}=\nu/\kappa$ indicates the property of specific fluid, where ν is the kinematic viscosity and κ the thermal diffusivity. The Rayleigh number is defined as $Ra=g\beta\Delta TH^3/(\nu\kappa)$.

The spatial discretization used is a conservative second-order central difference with velocities on a staggered grid, which is clustered near the upper and lower plates but equi-spaced in horizontal direction. The pressure and the temperature are calculated at the center of the cell. Velocities at cell center is interpolated when needed. The suitability of second-order discretization for turbulent flow simulations was demonstrated by Moin *et al*. [19]. It was checked and suggested [20] that second-order schemes can reliably produce first order and second order statistics, while higher accuracies (fourth or fifth order) are required for high-order statistics. It is worth to point out that low order scheme are better in dealing with sharp gradients, which ubiquitously exist for temperature field in RBC [17], because the advection/diffusion equation for temperature does not have an analogous counterpart of the pressure in the momentum equations.

Time marching is performed with first order explicit scheme. The accuracy of the method was confirmed by grid independent test and by good agreement of the Nusselt number with other simulation results, which will be discussed in the following sections.

Chorin's projection method [21, 22] is introduced to deal with velocity-pressure coupling. Three main steps are performed in sequence: First, an estimative velocity field (u^*, v^*) is calculated from the momentum equations but ignoring the pressure terms.

$$\begin{cases} \dfrac{u^*-u^n}{\Delta t}+\dfrac{\partial(uu)^n}{\partial x}+\dfrac{\partial(vu)^n}{\partial y}=\dfrac{1}{\sqrt{Ra/\mathrm{Pr}}}\left(\dfrac{\partial^2u^n}{\partial x^2}+\dfrac{\partial^2u^n}{\partial y^2}\right) \\ \dfrac{v^*-v^n}{\Delta t}+\dfrac{\partial(uv)^n}{\partial x}+\dfrac{\partial(vv)^n}{\partial y}=\dfrac{1}{\sqrt{Ra/\mathrm{Pr}}}\left(\dfrac{\partial^2v^n}{\partial x^2}+\dfrac{\partial^2v^n}{\partial y^2}\right)+\theta^n \end{cases} \tag{2}$$

Δt is the size of the time step. The superscript n denotes a quantity evaluated at the n step and $n+1$ identifies the next step.

The velocity of next time step is constructed using the estimative velocity and pressure field.

$$\begin{cases} \dfrac{u^{n+1}-u^*}{\Delta t}=-\dfrac{\partial p}{\partial x} \\ \dfrac{v^{n+1}-v^*}{\Delta t}=-\dfrac{\partial p}{\partial y} \end{cases} \tag{3}$$

Equations (2) and (3) add up to momentum equation in (1) with first order forward difference in time.

Second, by taking partial differentiation on (3), the pressure must satisfy a Poisson equation with boundary conditions $\dfrac{\partial p}{\partial x}=0$ (at sidewalls) and $\dfrac{\partial p}{\partial y}=0$ (at horizontal plates), so that the velocity of next time step (u^{n+1}, v^{n+1}) is divergence-free.

$$\frac{\partial^2p}{\partial x^2}+\frac{\partial^2p}{\partial y^2}=\frac{1}{\Delta t}\left(\frac{\partial u^*}{\partial x}+\frac{\partial v^*}{\partial y}\right) \tag{4}$$

Finally, with the pressure obtained from (4), the velocity of next time step is derived from (3).

$$\begin{cases} u^{n+1}=-\Delta t\dfrac{\partial p}{\partial x}+u^* \\ v^{n+1}=-\Delta t\dfrac{\partial p}{\partial y}+v^* \end{cases} \tag{5}$$

Once the momentum equation is solved by the above three-step procedure, the temperature of next time step is updated from

$$\begin{aligned} &\frac{\theta^{n+1}-\theta^n}{\Delta t}+u^{n+1}\frac{\partial\theta^n}{\partial x}+v^{n+1}\frac{\partial\theta^n}{\partial y} \\ &=\frac{1}{\sqrt{Ra\cdot\mathrm{Pr}}}\left(\frac{\partial^2\theta^n}{\partial x^2}+\frac{\partial^2\theta^n}{\partial y^2}\right) \end{aligned} \tag{6}$$

The discrete form of the Poisson equation under second order central difference scheme read

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

$$\frac{p_{i-1,j}-2p_{i,j}+p_{i+1,j}}{\Delta x^2}+\frac{\dfrac{2(p_{i,j+1}-p_{i,j})}{\Delta y_j+\Delta y_{j+1}}-\dfrac{2(p_{i,j}-p_{i,j-1})}{\Delta y_j+\Delta y_{j-1}}}{\dfrac{1}{2}\Delta y_j+\dfrac{1}{4}(\Delta y_{j-1}+\Delta y_{j+1})}=d_{i,j} \tag{7}$$

where $d_{i,j}$ is the discrete form of the right hand side of Eq. (4). Let N_x and N_y denote the grid points in horizontal and vertical directions respectively, and $m = N_x$. A discrete Fourier cosine transform (first equation of (8)) is performed on the unknowns in Eq. (7) in x direction, assuming that the input array is periodically extended such that it is even around $i = -0.5$ and even around $i = m - 0.5$, thus automatically satisfying the zero pressure gradient boundary condition.

$$\begin{cases} DFT: & \hat{a}_{k,j}=2\sum_{i=0}^{m-1}a_{i,j}\cos[\pi(i+1/2)k/m] \\ IDFT: & a_{i,j}=\hat{a}_{0,j}+2\sum_{k=1}^{m-1}\hat{a}_{k,j}\cos[\pi i(k+1/2)/m] \end{cases} \tag{8}$$

Substituting the inverse transform (second equation of (8)) into (7) gives

$$\begin{cases} \dfrac{4}{\Delta x^2}\sum_{i=1}^{m-1}\hat{p}_{i,j}\cos[\pi i(k+1/2)/m]\{\cos[\pi i/m]-1\}+\{a_{0,j}\hat{p}_{0,j-1}+(-a_{0,j}-c_{0,j})\hat{p}_{0,j}+c_{0,j}\hat{p}_{0,j+1}\}+ \\ \qquad 2\sum_{i=1}^{m-1}\{a_{i,j}\hat{p}_{i,j-1}+(-a_{i,j}-c_{i,j})\hat{p}_{i,j}+c_{i,j}\hat{p}_{i,j+1}\}\cos[\pi i(k+1/2)/m]= \\ \qquad\qquad \hat{d}_{0,j}+2\sum_{i=1}^{m-1}\hat{d}_{i,j}\cos[\pi i(k+1/2)/m] \end{cases} \tag{9}$$

where $a_{i,j}=\dfrac{2}{(\Delta y_j+\Delta y_{j-1})\left(\frac{1}{2}\Delta y_j+\frac{1}{4}(\Delta y_{j-1}+\Delta y_{j+1})\right)}$ $c_{i,j}=\dfrac{2}{(\Delta y_j+\Delta y_{j+1})\left(\frac{1}{2}\Delta y_j+\frac{1}{4}(\Delta y_{j-1}+\Delta y_{j+1})\right)}$. Comparing the coefficients of both sides of equation (9) gives

$$a_{i,j}\hat{p}_{i,j-1}+b_{i,j}\hat{p}_{i,j}+c_{i,j}\hat{p}_{i,j+1}=\hat{d}_{i,j} \tag{10}$$

where $b_{i,j}=2(\cos[\pi i/m]-1)-(a_{i,j}+c_{i,j})$ and $i = 0 \ldots N_x-1, j = 0 \ldots N_y-1$.

This operation results in N_x independent systems of tridiagonal equations, each of which has N_y unknowns in y direction. And each system is diagonal dominant, because obviously $|b_{i,j}| \geq |a_{i,j}| + |c_{i,j}|$.

With the consideration of parallelization in mind, one has to decide which kind of domain decomposition is used. The 1D "slab" decomposition is easy to implement, and it can be either x or y oriented. Since the calculation of velocities and temperature is local in space as they are advanced explicitly, the domain decomposition mainly affect the scheme for pressure equation. While the y oriented decomposition is convenient for solving the above-mentioned tridiagonal systems using traditional methods like Thomas' algorithm, it requires distributed FFT, which is generally less efficient and has a more complicated implementation interface.

If the x oriented decomposition is used instead (See Fig. 1), a parallel algorithm for tridiagonal system is required. In this case, because the tridiagonal system is partitioned in y direction, all-to-all communications are usually unavoidable, and become the main bottleneck of the overall computation. However, as the coefficients matrix of the tridiagonal systems is diagonal dominant, it can be solved using the parallel diagonal dominant (PDD) algorithm [23], which, when properly used, can obtain an approximate solution of machine accuracy with only two sequential neighbor communications. Therefore, we choose the x oriented decomposition over y to take the advantage of the elimination of all-to-all communications.

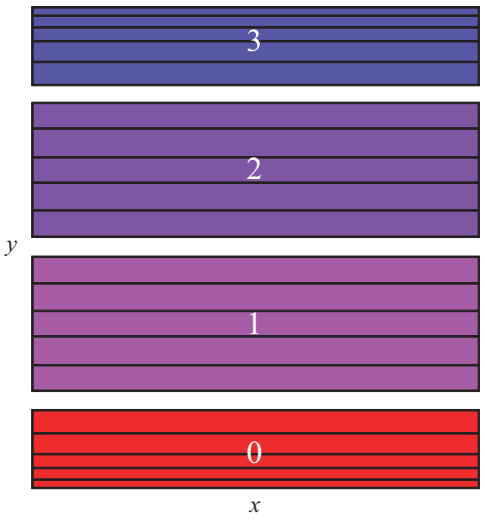


Fig. 1 A schematic of x oriented one-dimensional slap decomposition. Vertical grid lines are omitted for simplicity. The numbers denotes the rank of MPI processes.

Besides the coefficient matrix being diagonal dominant, to use the PDD algorithm one must satisfy another crucial condition $n \gg p$, where n is the order of the matrix and p the number of processes available. Based on

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

our tests on RBC, $n/p = q > 50$ is usually sufficient to obtain accurate solution, where q denotes the number of unknowns assigned to each process. The details and applications of the PDD algorithm is beyond the scope of this paper. We refer the readers to Ref. [24, 25] for more information. Here we briefly describe the procedure of employing the PDD algorithm to tridiagonal system, which is divided into p subsystems corresponding to p MPI processes:

Step 1. Each subsystem is reduced using Thomas's algorithm to two equations involving four unknowns (or three for the first and the last subsystems). Under the condition of $n \gg p$, the coefficients of the farthest unknowns become very small. Dropping these coefficients leads to two reduced and approximated equations, either one only related to an equation in neighbor subsystem.

Step 2. Information of the first (reduced and approximated) equation in a subsystem is sent to the last subsystem/process, as the first of two communications.

Step 3. Receivers of Step 2 solve two unknowns which are the last one of their own and the first one of the senders.

Step 4. The solution of Step 3 is sent back to the senders of Step 2, as the second of two communications.

Step 5. Other unknowns within a subsystems are solved based on the solution of Step 3.

The particularly exciting scalability of the present method, as evidenced in the next section, makes it distinguishable from others and we name it the parallel direct method of DNS (PDM-DNS) for Rayleigh-Bénard convection. 'Direct' in PDM indicates that we are not using any iterative solvers, especially for the Poisson equation.

3. COMPUTATIONAL PERFORMANCE

In this section we demonstrate the computational performance of PDM-DNS. In computing, two main factors determine the scalability of the program: load balancing and communication overhead. As described in previous section, the 1D slap decomposition divides the entire domain into equal size subdomains, and between every synchronization each core carries out the same amount of work. Throughout the program, as we do not have implicit terms, calculation of velocity and temperature is local in space, and only neighbor communications are made during synchronization. As for pressure solver, the PDD algorithm assures that no all-to-all communications are involved. Hence the number of neighbor communications is constant and independent of the number of processors. Therefore, not only the communication within a timestep is minimized, but also communication cost does not grow with the number of processes, making the parallelization scheme perfectly scalable.

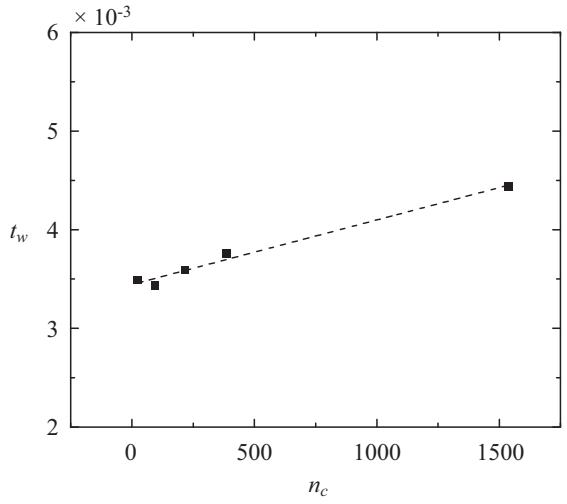


Fig. 2 Weak scaling for 1.18 Million per core. The dash line is a linear fit of all data points. t_w is the wall time per timestep and n_c is the number of cores being used.

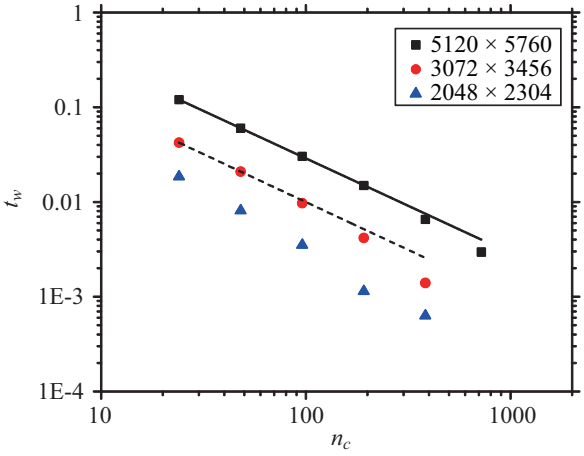


Fig. 3 Strong scaling of the code for 2048 × 2304 grid (triangles), 3072 × 3456 (circles) and 5120 × 5760 (squares) grids. The solid line is a linear fit of the first four squares, and the dash line is a linear fit of the first two circles. t_w is the wall time per timestep and n_c is the number of cores being used.

Due to the limit of the number of MPI processes introduced by the PDD algorithm, a hybrid MPI-OpenMP decomposition is not just feasible, but practically necessary. For a given amount of grid points, the ratio between the number of MPI processes N_p and the number of OpenMP threads N_{th} can be adjusted to optimize the performance on a specific computer platform. On Tianhe-2 supercomputer, we find that a ratio of $N_p/N_{th} = 2/12$ within each node is usually the optimal configuration. This ratio is adopted in all following tests. Figure 2 shows the weak scaling for grid size from 1024 × 1152 to 8192 × 9216 and n_c from 24 to 1536. Linear scaling can be seen in the full range, while the number of

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

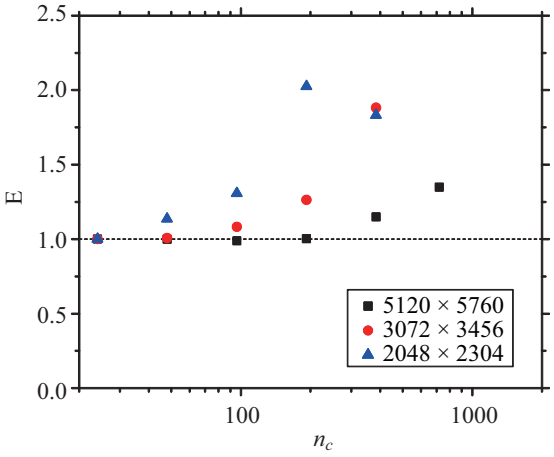


Fig. 4 Efficiency of strong scaling for the same data points in Fig. 3. The dash line denotes 100% for reference. See the text for the definition of efficiency E.

grid points assigned to each core remains approximately 1.18 Million. The linearly increase of the wall time is mainly due to the computational complexity of the FFT algorithm, which is of order $O(n \log n)$.

Figure 3 shows the strong scaling of the code for 2048 × 2304, 3072 × 3456 and 5120 × 5760 grids. For each grid, linear scaling can be seen in the beginning, followed by a surprisingly drop of wall time per timestep. These drops occur when the grid points assigned to each core is approximately less than 100 thousand. In Fig. 4, this scaling behavior is further shown by the parallel efficiency, defined as $E = W_1/(t_w \times n_c) \times 100\%$ where $W_1 = t_w \times n_c$ for the least $n_c = 24$, corresponding to one node. The increased efficiency at higher n_c can be explained by that the memory size coincides with the processors' cache, as known as superlinear scaling due to cache effect, which deserved a little more discussion.

It is well known that cache effect usually occur when the problem size of each core is sufficiently small, compared with the cache size. In fact, the problem size of each core is smaller for larger number of cores in strong scaling, where the global problem size remains constant. In weak scaling, the problem size of each core remains unchanged, although the shape of arrays may change. Therefore, assuming that the communication complexity remains constant, which is indeed the case in our tests, one would expect to see (1) the cache effect for large number of cores in strong scaling, (2) the cache effect for any number of cores in weak scaling only if the problem size of each core is sufficiently small. These two scalings should not be confused with each other.

However, in many parallel schemes, communication complexity grows with the number of cores, slowly or rapidly. Hence, cache effect, if observed otherwise, would be compromised for large number of cores, both in strong scaling and weak scaling. In that case, superlinear behavior due to cache effect would likely be observed only for small number of cores in weak scaling or not observed at all.

To summarize, constant communication complexity, as the key feature of the present method, makes it possible to fully expose the cache effect and consequently leads to the observed superlinear behavior. Nevertheless, user or adapter of the present method should not rely on this superlinear behavior in production, because it heavily depends on the optimization capability of compiler and the architecture of CPU.

The relative narrow range of parallel scaling, both strong and weak scaling, is due to the double restriction introduced by 1D slap decomposition and the PDD algorithm. However, from a scientific viewpoint, the major obstacle to massive scale problems, such as simulations of "ultimate regime" in RBC, is the parallel overhead. And this obstacle is overcome by the presented method, as it eliminates all-to-all communication. Therefore, in a sense, the restriction here encourages us to deal with larger size problems, in order to utilize the perfect parallel scaling.

4. NUMERICAL EXPERIMENTS

In this section, we present the numerical results of RBC and argue that high resolution both in space and time are required for very high Ra numbers.

As Ra number increases, the smallest length scale of the flow (Kolmogorov scale η_K and Batchelor scale η_B) decreases. The resolution of DNS must increase accordingly to fully resolve the flow structures. In our numerical experiments, the resolution is increased from 1024 × 1152 for $Ra = 10^9$ to 3072 × 3456 for $Ra = 10^{12}$. We note that the number of grid points inside the thermal boundary layers (BLs) exceeds the requirement studied by Shishkina *et al.* [26]. We have 21 points and 37 points inside the BLs for $Ra = 10^9$ and $Ra = 10^{12}$ respectively, while the requirements are 5 and 13. We also check the Courant–Friedrichs–Lewy (CFL) conditions by calculating the CFL number, which is less than 0.02 in all simulations.

Integral quantities, such as the Nusselt numbers $Nu = \sqrt{Ra \cdot \mathrm{Pr}} < v\theta >_{A,t} - \left\langle \frac{\partial \theta}{\partial y} \right\rangle_{A,t}$ (<…>$_{A,t}$ denotes averaging in time and at a fixed height), can be used to verify our results. Given enough physical time, the integral quantities are expected to converge statistically. A grid independent test is conducted for $Ra = 10^{10}$, Pr = 0.7 on three grids 1024 × 1152, 1280 × 1440 and 1536 × 1728. The relative error of Nu from three grids is within 1.1%. Figure 5 shows the compensated Nu plotted against Ra, where the dimensionless time of averaging is at least 80 for every Ra. Data points extracted from van der Poel *et al.* [27] show good agreement with the present calculation. The relative error of Nu at $Ra = 10^9$, $Ra = 10^{10}$ and $Ra = 10^{11}$ are within 2.7%, 0.1% and 4.7% respectively. It is worth to mention that, long live transients in RBC [28] make averaging difficult, as the system may retain a flow state with a particular Nu number for a long time until it switches to other flow states. In order to explore every possible transients, we need to simulate a sufficiently long period of physical time. The scalability of the present method assures that

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

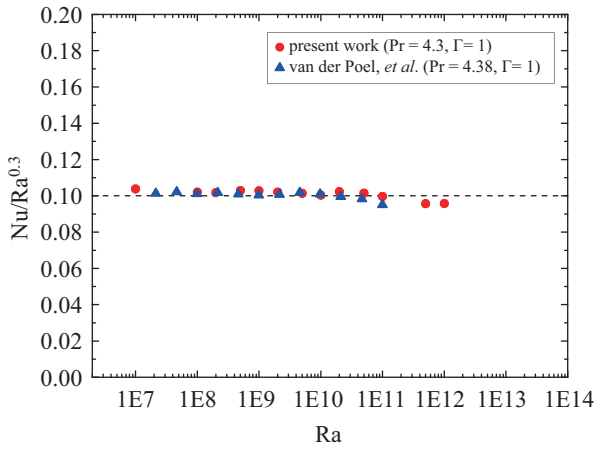


Fig. 5 Nusselt number compensated by $Ra^{0.3}$ for Pr = 4.3 and aspect-ratio $\Gamma = 1$. The simulation result by the present method is indicated by red circles, while blue triangles indicate result extracted from van der Poel *et al*. [27]

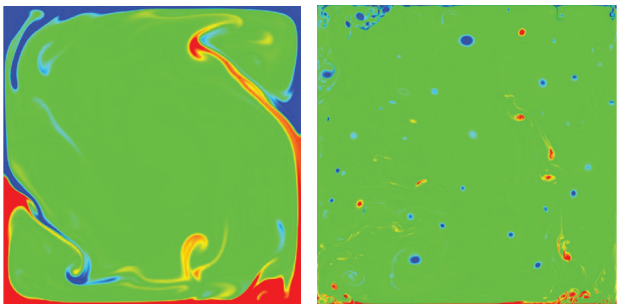


Fig. 6 Instantaneous temperature fields for well resolved cases of (a) $Ra = 10^9$, Pr = 4.3, timestep $\Delta t = 5.0 \times 10^{-5}$ and (b) $Ra = 10^{12}$, Pr = 4.3, timestep $\Delta t = 2.0 \times 10^{-6}$.

we are able to achieve this at reasonable cost for very high Ra numbers.

Well resolved cases are given for $Ra = 10^9$ and $Ra = 10^{12}$ in Fig. 6. The entire domain is full of structures from the largest scale to the smallest scale. By comparing the instantaneous fields in Fig. 6, we can see that not only the plume size of high Ra number is smaller, but also the plumes can remain in the bulk area, traveling along large eddies for a long time until they disappear. Stable corner rolls are no longer exist for high Ra number, and replaced by plume emission or impact areas. For $Ra = 10^{12}$, we can see that thermal plumes formed adjacent to boundary layers roll into round shaped "temperature islands". These round shaped plumes usually associate with high rotation speed, which may stretch other plumes around them and together they develop very fine and complex structures.

Simulation which is under-resolved in time may give non-physics solutions. Figure 7 shows a zoom-in part of an instantaneous temperature field for Ra = 10^{12}, Pr = 4.3, performed on the same 3072×3456 grid as Fig. 6(b), but with a larger timestep $\Delta t = 4.0 \times 10^{-6}$. Even though the solution is stable, the internal structure of a thermal plume is destroyed. In RBC, the magnitude of the fluctuation increases with Ra, eventually higher than the magnitude of the mean flow [29], which is intuitively observed as the vortices or thermal plumes rotate, merge and strike against each other. Thus, in addition to high resolution in space to capture these fine structures, we need high resolution in time to capture rapid vortex motion.

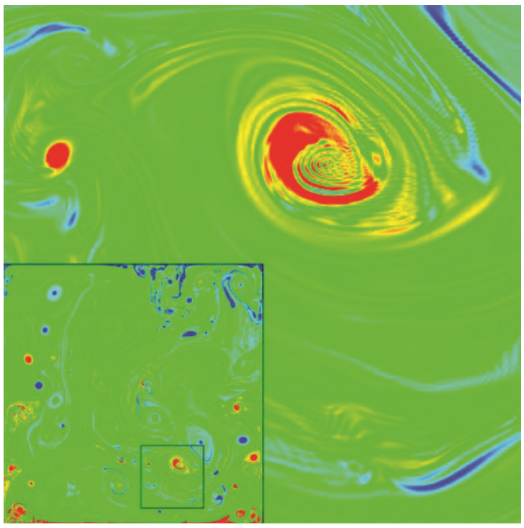


Fig. 7 A zoom-in part of an instantaneous temperature field for $Ra = 10^{12}$, Pr = 4.3 and timestep $\Delta t = 4.0 \times 10^{-6}$. The location of the structure is marked by a black square in the inset.

5. CONCLUSIONS

A new highly efficient parallelization scheme of DNS for 2D Rayleigh-Bénard convection, PDM-DNS, is established in the present paper. The PDD algorithm is introduced to a second order accurate finite difference scheme developing into a perfectly scalable method for solving Rayleigh-Bénard convection in two-dimensional rectangular box. In this method, all-to-all communication is eliminated, preventing the parallel overhead from growing with the number of computation cores. The ideal strong scaling and weak scaling are achieved at the expanse of limiting the number of MPI processes for a given amount of grid points in y direction. Numerical results show that the convection flow with very high Ra numbers contain very fine structures which locally associated with complex vortex motion. The scalability of the present method renders it preferable for high resolution both in space and in time to resolve such flow structures. In particular, the "ultimate regime" at very high Ra numbers, one of the important research aims for the Rayleigh-Bénard convection, may be attained in simulations at reasonable cost.

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

ACKNOWLEDGEMENTS

The work reported here is funded by the National Natural Science Foundation of China (11372362, 11452002).

REFERENCES

1. Ahlers, G., Grossmann, S. and Lohse, D., "Heat Transfer and Large Scale Dynamics in Turbulent Rayleigh-Bénard Convection," *Reviews of Modern Physics*, **81**, pp. 503-537 (2009).
2. Heslot, F., Castaing, B. and Libchaber, A., "Transitions to Turbulence in Helium Gas," *Physical Review A*, **36**, pp. 5870-5873 (1987).
3. Grossmann, S. and Lohse, D., "Scaling in Thermal Convection: A Unifying Theory," *Journal of Fluid Mechanics*, **407**, pp. 27-56 (2000).
4. Grossmann, S. and Lohse, D., "Thermal Convection for Large Prandtl Numbers," *Physical Review Letters*, **86**, pp. 3316-3319 (2001).
5. Grossmann, S. and Lohse, D., "Prandtl and Rayleigh Number Dependence of the Reynolds Number in Turbulent Thermal Convection," *Physical Review E*, **66**, pp. 1-6 (2002).
6. Grossmann, S. and Lohse, D., "Fluctuations in Turbulent Rayleigh–Bénard Convection: The Role of Plumes," *Physics of Fluids*, **16**, pp. 4462-4472 (2004).
7. Kraichnan, R., "Turbulent Thermal Convection at Arbitrary Prandtl Number," *Physics of Fluids*, **5**, pp. 1374-1389 (1962).
8. Stevens, R., van der Poel, E., Grossmann, S. and Lohse, D., "The Unifying Theory of Scaling in Thermal Convection: the Updated Prefactors," *Journal of Fluid Mechanics*, **730**, pp. 295-308 (2013).
9. Verzicco, R. and Orlandi, P., "A Finite-Difference Scheme for Three-Dimensional Incompressible Flows in Cylindrical Coordinates," *Journal of Computational Physics*, **123**, pp. 402-414 (1996).
10. Shishkina, O. and Wagner, C., "A Fourth Order Accurate Finite Volume Scheme for Numerical Simulations of Turbulent Rayleigh–Bénard Convection in Cylindrical Containers," *Comptes Rendus Mécanique*, **333**, pp. 17-28 (2005).
11. Xia, K.-Q., "Current Trends and Future Directions in Turbulent Thermal Convection," *Theoretical and Applied Mechanics Letters*, **3**, 52001 (2013).
12. Kaczorowski, M., Shishkin, A., Shishkina, O. and Wagner, C., "Developement of a Numerical Procedure for Direct Simulations of Turbulent Convection in a Closed Rectangular Cell," *New Results in Numerical and Experimental Fluid Mechanics VI*, **96**, pp. 381-388 (2007).
13. Daya, Z. and Ecke, R., "Does Turbulent Convection Feel the Shape of the Container?," *Physics Review Letters*, **87**, 184501 (2001).
14. Chong, K.-L., Huang, S.-D., Kaczorowski, M. and Xia, K.-Q., "Condensation of Coherent Structures in Turbulent Flows," *Physics Review Letters*, **115**, 264503 (2015).
15. Huang, S.-D., Kaczorowski, M., Ni, R. and Xia, K.-Q., "Confinement-Induced Heat-Transport Enhancement in Turbulent Thermal Convection," *Physics Review Letters*, **111**, 104501 (2013).
16. Chong, K.-L. and Xia, K.-Q., "Exploring the Severely Confined Regime in Rayleigh–Bénard Convection," *Journal of Fluid Mechanics*, **805**, R4 (2016).
17. Ostilla-Monico, R., Yang, Y., van der Poel, E., Lohse, D. and Verzicco, R., "A Multiple-Resolution Strategy for Direct Numerical Simulation of Scalar Turbulence," *Journal of Computational Physics*, **301**, pp. 308-321 (2015).
18. van der Poel, E., Ostilla-Mónico, R., Donners, J. and Verzicco, R., "A Pencil Distributed Finite Difference Code for Strongly Turbulent Wall-Bounded Flows," *Computers & Fluids*, **116**, pp. 10-16 (2015).
19. Moin P. and Verzicco, R., "On the Suitability of Second-Order Accurate Discretizations for Turbulent Flow Simulations," *European Journal of Mechanics - B/Fluids*, **55**, pp. 242-245 (2016).
20. Mohan Rai, M. and Moin, P., "Direct Simulations of Turbulent Flow Using Finite-Difference Schemes," *Journal of Computational Physics*, **96**, pp. 15-53 (1991).
21. Chorin, A., "On the Convergence of Discrete Approximations to the Navier-Stokes Equations," *Mathematics of Computation*, **23**, pp. 341-353 (1969).
22. Chorin, A., "Numerical Solution of the Navier-Stokes Equations," *Mathematics of Computation*, **22**, pp. 745-762 (1968).
23. Sun, X.-H., Sun, H.-Z. and Ni, L. M., "Parallel Algorithms for Solution of Tridiagonal Systems on Multicomputers," *Proceedings of the 3rd International Conference on Supercomputing - ICS '89*, pp. 303-312 (1989).
24. Zhang, H., "On the Accuracy of the Parallel Diagonal Dominant Algorithm," *Parallel Computing*, **17**, pp. 265-272 (1991).
25. Sun, X.-H., "Application and Accuracy of the Parallel Diagonal Dominant Algorithm," *Parallel Computing*, **21**, pp. 1241-1267 (1995).
26. Shishkina, O., Stevens, R., Grossmann, S. and Lohse, D., "Boundary Layer Structure in Turbulent Thermal Convection and its Consequences for the Required Numerical Resolution," *New Journal of Physics*, **12**, 75022 (2010).
27. He, X., Gils, D., Bodenschatz, E. and Ahlers, G., "Reynolds Numbers and the Elliptic Approximation Near the Ultimate State of Turbulent Rayleigh–

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press

Bénard Convection," *New Journal of Physics*, **17**, 63028 (2015).

28. van der Poel, E., Stevens, R. and Lohse, D., "Comparison Between Two- and Three-Dimensional Rayleigh–Bénard Convection," *Journal of Fluid Mechanics*, **736**, pp. 177-194 (2013).
29. van der Poel, E., Stevens, R., Sugiyama, K. and Lohse, D., "Flow States in Two-Dimensional Rayleigh-Bénard Convection as a Function of Aspect-Ratio and Rayleigh Number," *Physics of Fluids*, **24**, 85104 (2012).

(Manuscript received December 29, 2016,
accepted for publication June 8, 2017.)

https://doi.org/10.1017/jmech.2017.54 Published online by Cambridge University Press