수학계산학부(계산과학공학)
School of Mathematics and Computing
(Computational Science and Engineering)

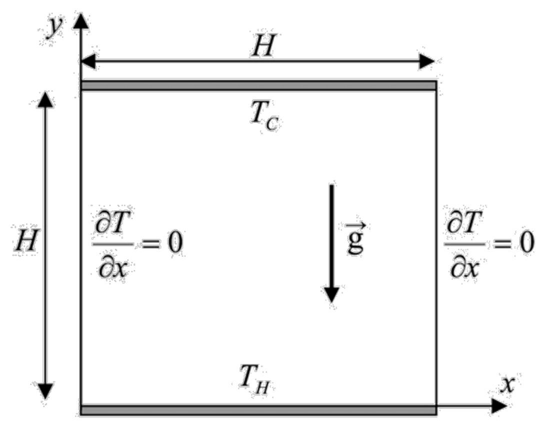
Optimized GPU-Based Simulation of two-dimensional Rayleigh-Bénard Convection in an Enclosure

Jin-Hyung Bae | Dept. of Mechanical Engineering, Turbulence Lab, Yonsei University

Undergraduate Research Program, School of Mathematics and Computing (Computational Science and Engineering), Yonsei University

Abstract

- RBC is a buoyancy-driven flow in a fluid layer with a heated bottom and a cooled top. When the temperature difference is large enough, convection cells develop and circulate within the fluid layer.
- RBC is crucial in heat transfer, atmospheric dynamics, and industrial cooling. At high Rayleigh numbers, flow transitions to turbulence, requiring high-resolution simulations for accurate analysis.



- High Rayleigh number simulations demand fine grids, making CPU-based solvers inefficient. GPU parallel computing accelerates key calculations, enabling faster and more efficient high-resolution simulations.

Key Concepts

• Governing equations

- Continuity Equation

$$\nabla \cdot u = 0$$

- Momentum Equation (Navier-Stokes Equation)

$$\frac{\partial u}{\partial t} + (u \cdot \nabla) u = -\nabla p + \sqrt{\frac{Pr}{Ra}} \nabla^2 u + \theta$$

- Energy Equation

$$\frac{\partial \theta}{\partial t} + (u \cdot \nabla)\theta = \sqrt{\frac{1}{Ra \cdot Pr}} \nabla^2 \theta$$

• Numerical Methods & Code Optimization

Used *QUICK Scheme* for advection term discretization, *2nd order Adams-Bashforth method* for time integration, *Explicit method* for solving the energy equation.
GPU acceleration is implemented on Google Colab with CUDA, optimizing key numerical operations, achieving high-resolution simulations within limited resources:

- *Pressure-Poisson equation*: Parallelized, vectorized iterative solver (SOR method)
- *Intermediate velocity calculation*: Tridiagonal matrix operations (TDMA)
- *Advection term calculation*: Vectorized operations
- *Stream function computation*: Parallelized, vectorized iterative solver (SOR method)

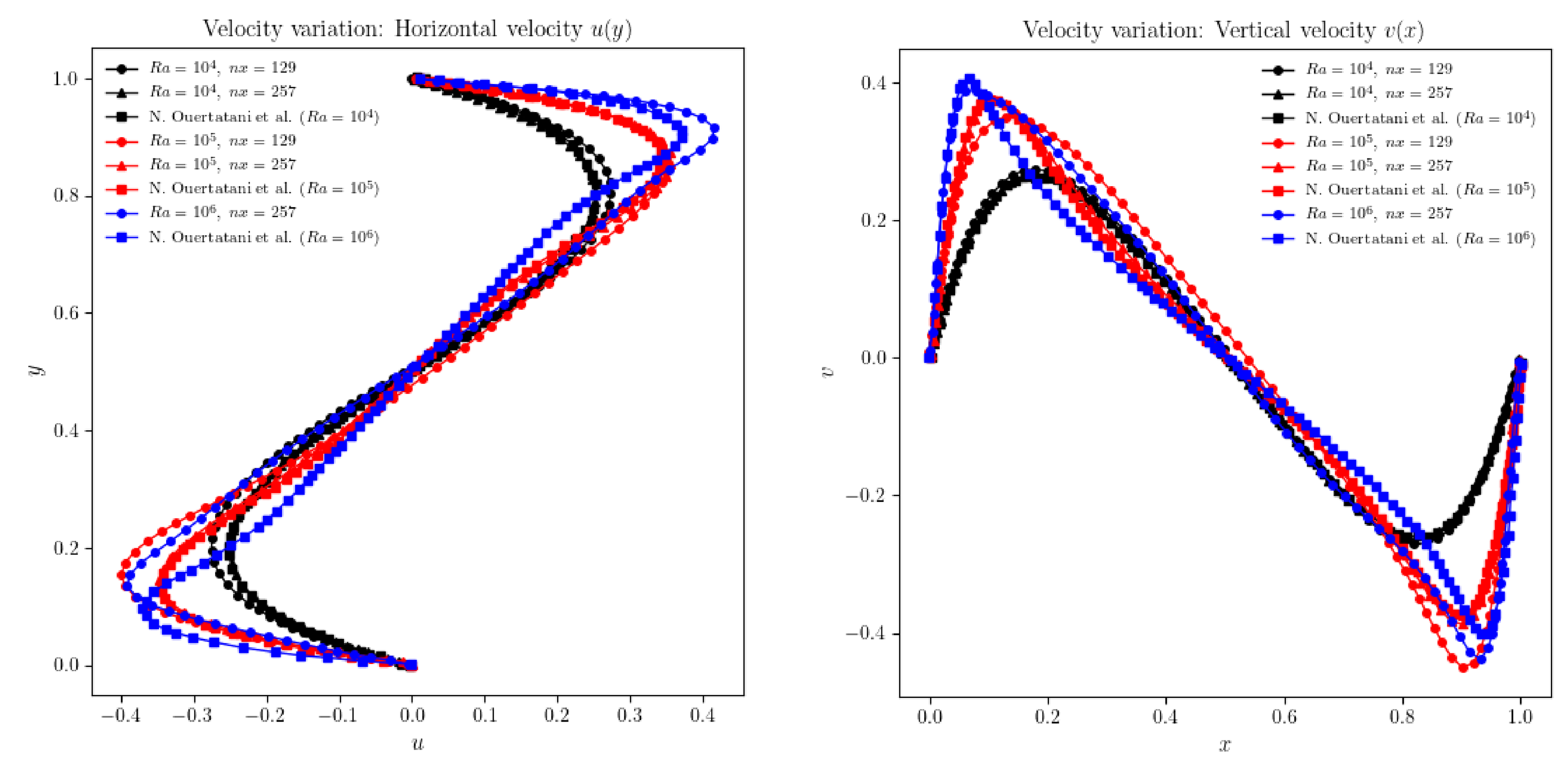
• Performance Comparison

To evaluate the impact of numerical optimizations and GPU acceleration, I compared the performance of a non-optimized CPU-based simulation and a GPU-accelerated, optimized simulation at a fixed Rayleigh number $Ra = 10^5$. The table below shows the improvements achieved through key optimizations:

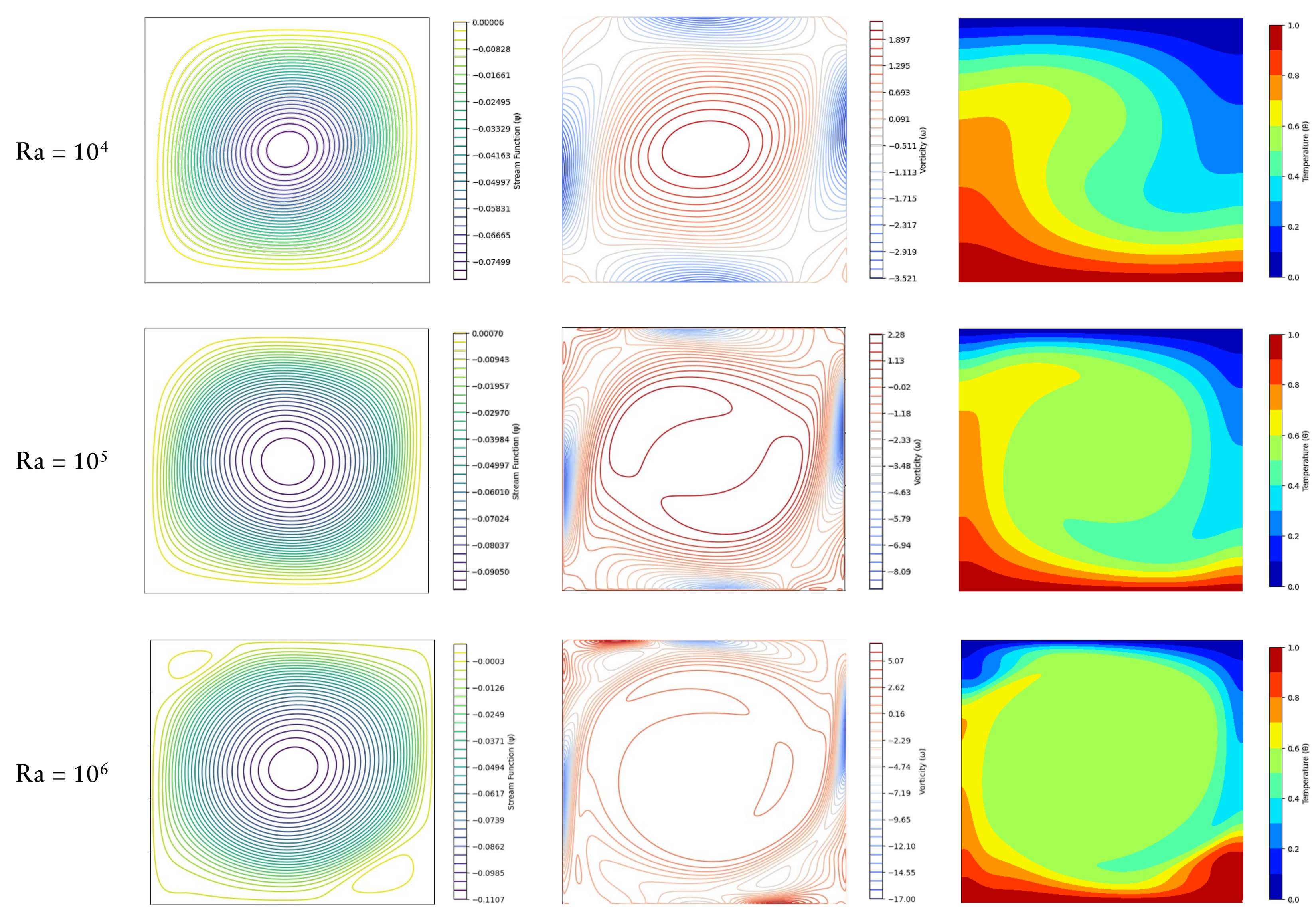
Method	Grid Size	Time/step [s]	Improvement Factor
Non-Optimized (CPU, NumPy)	129 × 129	2.72	1.0× (Baseline)
Optimized (GPU, CuPy)	129 × 129	0.283	**9.61×**
Non-Optimized (CPU, NumPy)	257 × 257	41.88	1.0× (Baseline)
Optimized (GPU, CuPy)	257 × 257	1.114	**37.59×**

Results

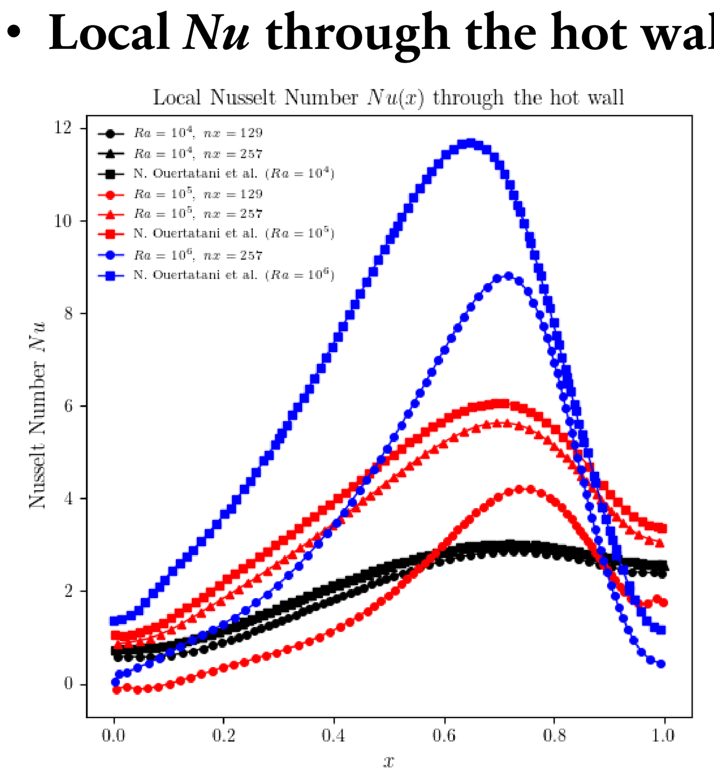
• Velocity Variation: Horizontal velocity $u(y)$ and vertical velocity $v(x)$



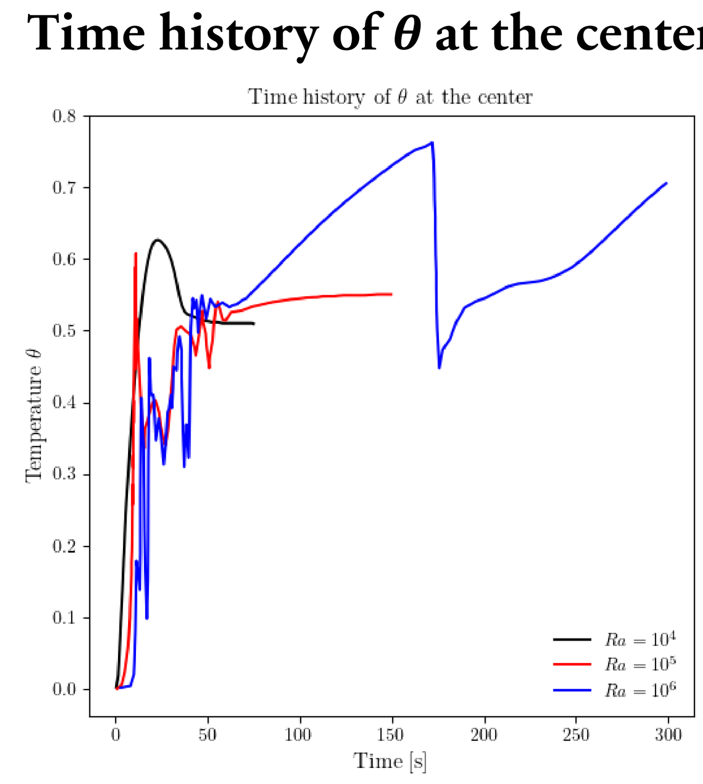
• Flow Visualization: Streamline, Vorticity, and Isotherm contours



• Local *Nu* through the hot wall



• Time history of θ at the center



Discussion & Conclusions

• Flow Regimes & Computational Complexity

- Ra = 10^4 (Laminar flow): Stable and symmetric circulation cells with smooth gradients.
- Ra = 10^5 (Transition flow): Onset of flow instabilities, increasing convective heat transfer.
- Ra = 10^6 (Turbulent flow): Highly chaotic, non-symmetric structures with enhanced mixing and heat transfer.
- As flow becomes more turbulent, computations become increasingly complex, requiring higher grid resolution, larger time steps, and longer computational time.
- For Ra = 10^6, a 129 × 129 grid led to numerical divergence (temperature exceeding 1).
  → Higher grid resolution (257 × 257) stabilized solutions and aligned in some degree with published results.

• Time History Analysis & Grid Resolution

- To determine quasi-steady-state time, central u-velocity and temperature time history profiles were analyzed.
- Simulations were performed at each 75s, 150s, 300s, for Ra = 10^4, 10^5, 10^6, showing better convergence at larger time steps.
- Increasing grid resolution led to solutions that closely matched existing literature.

• *Nu* Accuracy & High-Order Differencing

- *Nu* calculation: Used first-order forward differencing, but results can be further refined using higher-order methods.
- Even with first-order differencing, a 257 × 257 grid provided sufficiently accurate results for Ra = 10^4, 10^5.

• Comparison with Previous Studies

- Unlike studies using multigrid FVM, this research applies FDM with projection method and GPU acceleration to achieve precise solutions within limited resources.
- Even at 129 × 129, plotting results illustrates the importance of high-resolution grids for complex flow patterns.

• Limitations & Future Work

- Constraint of runtime made full steady-state resolution for Ra = 10^6 challenging.
- Higher computational power (e.g., research lab GPU servers) could improve accuracy for large Ra cases.
- Potential future extensions to three-dimensional, cylindrical, and rotating RBC simulations with better resources.

References

- [1] N. Ouertatani et al., Numerical simulation of two-dimensional Rayleigh–Bénard convection in an enclosure, C. R. Mecanique 336 (2008) 464–470
- [2] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Computer Methods in Applied Mechanics and Engineering, Volume 19, Issue 1, 1979, Pages 59-98, ISSN 0045-7825