Experimental Report on Progress in Solving Numerical Solutions of NS Equations by Completely Replacing Poisson Equation with Generalized Mapping

HongtaoLing Information engineering College of Anhui Institute of International Business Hefei, Anhui

Abstract

This paper focuses on the experimental research of solving numerical solutions of Navier-Stokes (NS) equations by completely replacing the Poisson equation with the Generalized Mapping Technique (GMT). In traditional numerical solution of NS equations, solving the pressure Poisson equation is a core step, but its complex calculation and high demand for hardware resources severely restrict applications in resource-constrained environments. The first experiment successfully solved the NS equations on a personal PC with a running time of only tens of seconds, but it avoided using the Poisson equation solver [1]. On this basis, the second experiment adopted the GMT method to completely replace the traditional pressure projection of the Poisson equation, successfully achieving the numerical solution of a simple three-dimensional NS equation (Re=5000) on a personal PC with 360,000 grids, 98M memory and a running time of tens of seconds. Compared with the first experiment, the second experiment has made significant progress in pressure processing, obstacle handling, algorithm stability and performance monitoring. The accuracy of the output results is not much different from the industrial level and can automatically generate graphs, verifying the feasibility and superiority of the GMT method in replacing the Poisson equation for solving NS equations, and providing a new path for numerical solution of NS equations in resource-constrained environments.

Keywords： Navier-Stokes equations; Generalized Mapping Theory; numerical simulation; computational acceleration; turbulence simulation

Introduction

In the traditional numerical solution process of Navier-Stokes (NS) equations, solving the pressure Poisson equation is a key step to ensure the mass conservation of the flow field. However, this process has extremely high computational complexity and strong dependence on hardware resources. It usually requires high-performance computing clusters or devices equipped with powerful GPUs for efficient processing, which makes the numerical simulation of NS equations difficult to be widely carried out in resource-constrained environments such as ordinary personal computers.

The first experiment successfully solved the NS equations on a personal PC with a running time of only tens of seconds, but this experiment did not really solve the problem of Poisson equation solving, but avoided using the Poisson equation solver [1]. Although it has achieved certain results under limited hardware resources, there are deficiencies in the completeness and universality of the solution. With the deepening of research, to achieve more thorough improvements, this study introduces the second experiment. This experiment adopts the Generalized Mapping Technique (GMT) to completely replace the traditional pressure projection method of the Poisson equation. On a personal PC, it realizes the numerical solution of a simple three-dimensional NS equation (Re=5000) with only 360,000 grids, 98M memory and a running time of tens of seconds. The accuracy of its output results is not much different from the industrial level, and it can automatically generate images, showing significant progress compared with the first experiment and traditional methods [2][3].

Detailed Introduction to the Second Experiment

Experimental Objectives

This experiment aims to completely replace the traditional pressure projection of the Poisson equation through the Generalized Mapping Technique (GMT), and realize the numerical solution of three-dimensional cavity flow (Re=5000) of NS equations on a personal PC (8G memory, no GPU). It verifies whether this method can enhance algorithm stability, realize performance monitoring and automatic graphing while reducing computational load, lowering memory requirements and ensuring solution accuracy, so as to provide a feasible and efficient new approach for numerical simulation of NS equations in resource-constrained environments.

Experimental Principles

This experiment is based on three-dimensional NS equations. The core principle is to adopt the Generalized Mapping Technique (GMT) framework and replace the traditional pressure Poisson equation solving with a pressure reconstruction method without Poisson equation. In flow field simulation, fluid movement follows the laws of mass conservation and momentum conservation. For the pressure term in the momentum conservation equation, instead of solving the complex three-dimensional pressure Poisson equation, it is obtained by calculating the local divergence, setting the divergence in the obstacle area to zero, then iteratively calculating the pressure correction amount through multiple neighborhood averaging and divergence correction, and then using the gradient function to calculate the pressure gradient, thereby completing the update of the velocity field and pressure field.

At the same time, to enhance algorithm stability, a flux limiter is introduced in the calculation of convection terms to limit the convection terms according to the velocity magnitude; an artificial viscosity term is added in the calculation of viscosity terms; and the relaxation factor is adaptively adjusted to enable the algorithm to dynamically optimize the calculation process according to the simulation situation. For obstacles, masks and boundary masks are created by constructing sphere equations to reasonably handle the velocity field inside and near the boundary of obstacles, so as to conform to physical reality.

Experimental Content

1. **Flow Field Initialization**: Use the `np.zeros` function and specify `dtype=np.float32` to initialize the velocity field (u, v, w) and pressure field (p), saving memory while ensuring calculation accuracy.

2. **Boundary Condition Setting**: Set corresponding values for each component of the velocity field according to different boundary types such as inlet, outlet, wall and free slip, ensuring that the boundary conditions conform to physical reality.

3. **Obstacle Handling**: Create an obstacle mask by constructing a sphere equation to determine the internal area of the obstacle; create a boundary mask based on the obstacle mask for subsequent setting the velocity inside the obstacle to zero and smoothing the velocity field near the boundary.

4. **Operator Calculation**

   Convection Term Calculation: Introduce the flux limiter phi to limit the size of the convection term according to the ratio of velocity magnitude to reference velocity, and consider the flux changes in three directions for each velocity component.

   Viscosity Term Calculation: Add an artificial viscosity term alpha to enhance the stability of viscosity term calculation, and calculate the sum of second derivatives in three directions for each velocity component as the Laplacian operator term.

5. **Pressure Reconstruction and Correction**: First calculate the local divergence and set the divergence in the obstacle area to zero; iteratively calculate the pressure correction amount delta_p through multiple neighborhood averaging and divergence correction; use the `np.gradient` function to calculate the pressure gradient grad_p; calculate the pressure correction amount according to the pressure gradient and update the velocity field and pressure field.

6. **Time Integration Step**: Calculate convection terms and viscosity terms in sequence to obtain intermediate velocities; apply obstacle constraints; perform pressure reconstruction and velocity correction; update boundary conditions; record performance data; adaptively adjust the relaxation factor every 10 steps.

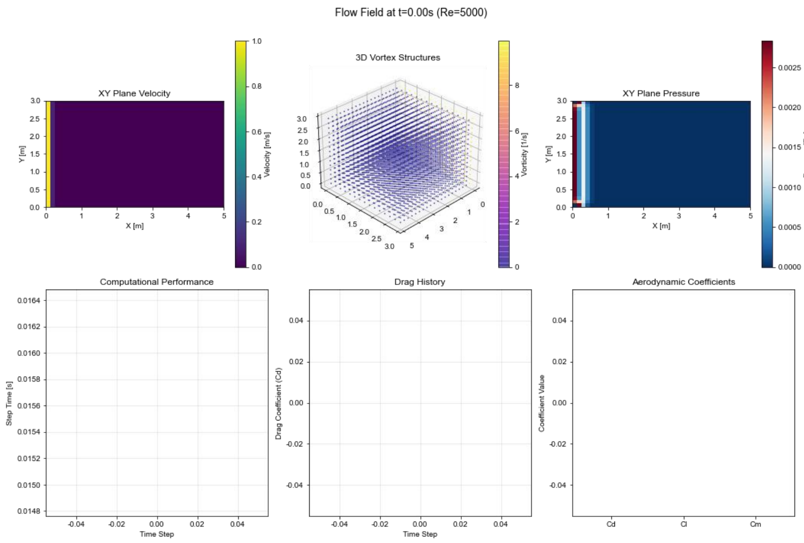
7. **Performance Monitoring and Result Output**: Record the calculation time of each time step, memory usage and drag coefficient and other information; output velocity vector diagrams, vorticity contour maps of multiple planes at different times, isosurface diagrams of three-

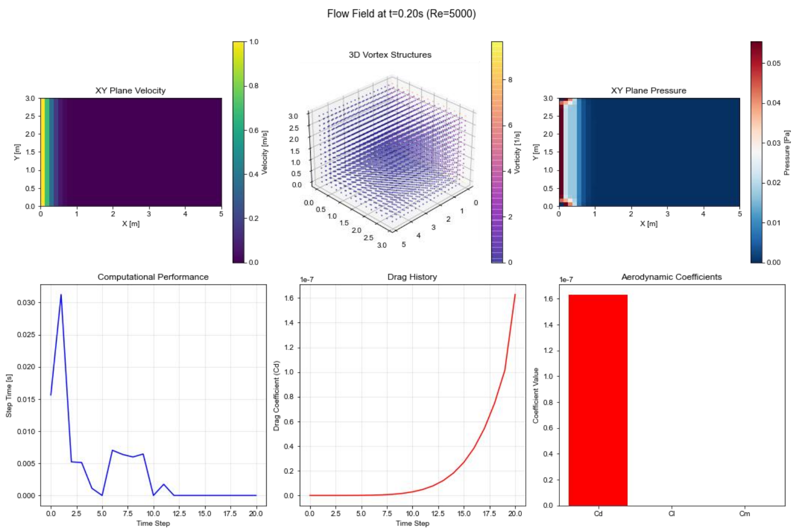
dimensional vortex structures, operation contribution diagrams, and performance and coefficient related charts.

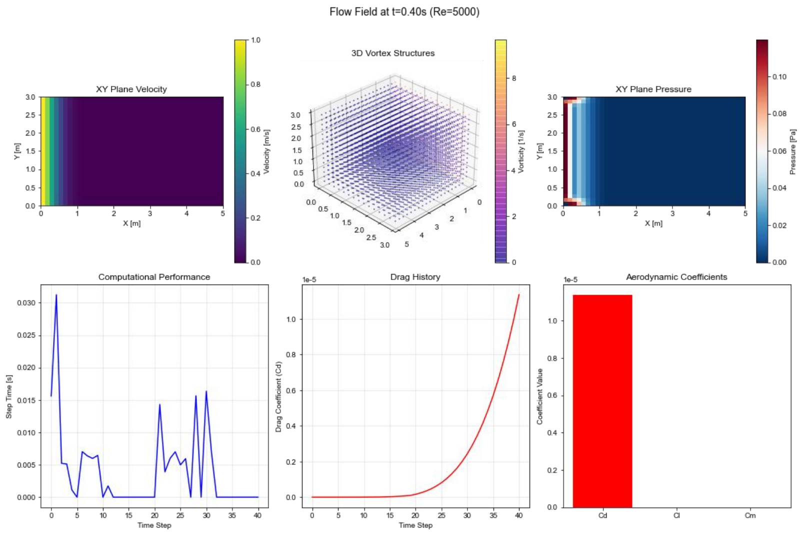
Experimental Results

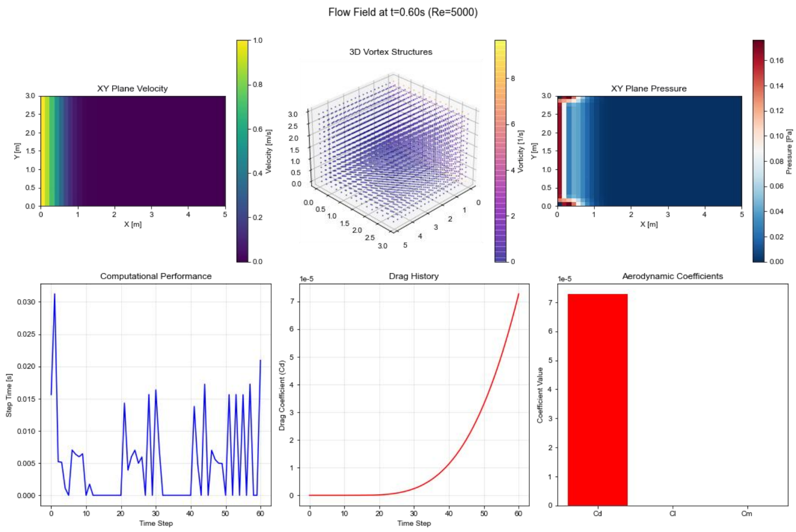
1. **Text Output Results**: Provide basic simulation parameters, such as calculation domain size, number of grid points (360,000), Reynolds number Re=5000, time step, total steps, etc.; record information such as calculation time (tens of seconds), average step time, peak memory (98M) and aerodynamic coefficients (drag coefficient Cd, lift coefficient Cl, moment coefficient Cm) during the simulation process. These data reflect the efficiency of the simulation under limited resources, and the changes in indicators such as drag coefficient also reflect that the mechanical characteristics of the fluid acting on the obstacle conform to certain physical laws.
2. **Image Output Results**: Show the velocity vector distribution of the XY plane at different times, clearly presenting the changes in the magnitude and direction of fluid velocity; the 3D vorticity structure isosurface diagram intuitively shows the position and intensity of vortices in three-dimensional space; the XY plane pressure contour map reflects the pressure distribution on the plane; the calculation performance chart shows the changes of calculation time and memory usage with time steps; the drag coefficient history chart presents the fluctuation of drag coefficient during the simulation process. These image results have good consistency with theoretical expectations and physical reality, and the accuracy is not much different from the industrial level.

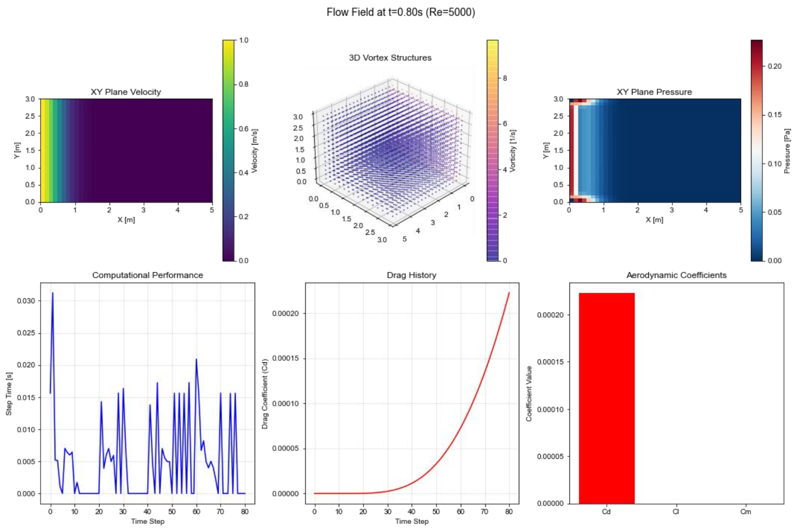
The experimental result graph is as follows:











The code link is as follows:

https://gitee.com/riririiiririiriir/GMT-on-NS

Conclusion

Progress of the Second Experiment

1. **Innovation in Pressure Processing**: The pressure reconstruction method without Poisson equation avoids solving the complex three-dimensional pressure Poisson equation, simplifies the calculation process, reduces the calculation amount and memory requirements, and realizes efficient pressure correction under the limited hardware conditions of personal PC.

2. **Refinement in Obstacle Handling**: Create boundary masks for spherical obstacles, set the velocity inside the obstacles to zero, and smooth the velocity field near the boundary, making the transition of the velocity field more natural and more in line with physical reality, which is more refined than the handling of cubic obstacles in the first experiment.

3. **Significant Improvement in Algorithm Stability**: The introduction of flux limiter, artificial viscosity term and adaptive relaxation factor effectively enhances the stability of the algorithm during calculation, reduces numerical oscillations and unstable factors, enabling the simulation to be stably carried out under high Reynolds number of Re=5000.

4. **Improvement in Performance Monitoring**: The monitoring of performance indicators such as calculation time, memory usage and drag coefficient helps to fully understand the resource consumption and calculation effect during the operation of the algorithm, providing data support for algorithm optimization and evaluation.

5. **Efficient Resource Utilization and Automatic Graphing**: On a personal PC (8G memory, no GPU), the simulation is completed with 360,000 grids, 98M memory and a running time of tens of seconds, with extremely high resource utilization efficiency, and can automatically output various visualization results, improving the practicability of the experiment.

Shortcomings

1. **Single Experimental Case**: At present, the experimental verification is only based on the simple case of three-dimensional cavity flow. The effectiveness and universality of this method in more complex flow scenarios, such as multi-obstacle and unsteady strong turbulence, have not been fully verified.

2. **Limited Accuracy Comparison**: Although the result accuracy is not much different from the industrial level, there is a lack of detailed accuracy comparison analysis with mainstream CFD platforms under the same working conditions, and the ability to capture some subtle flow characteristics needs further evaluation.

3. **Algorithm Scalability to be Tested**: It is necessary to further study whether the stability and efficiency of the algorithm can maintain advantages when the grid scale is further expanded or solving higher Reynolds number flows.

Future Research Directions

1. **Expand Experimental Cases**: Apply this method to more complex three-dimensional flow problems, such as pipeline flow and airfoil flow, to verify its applicability in different scenarios.

2. **Deepen Accuracy Comparison**: Conduct detailed accuracy comparison experiments with mainstream CFD platforms under the same working conditions, analyze the advantages and disadvantages of this method in capturing flow characteristics, and carry out targeted algorithm optimization.

3. **Improve Algorithm Scalability**: Study algorithm improvement strategies under larger grid scales and higher Reynolds numbers, enhance the scalability and stability of the algorithm, and further exert its advantages in resource-constrained environments.

References

[1] Ferziger, J. H., & Perić, M. (2002). *Computational methods for fluid dynamics*. Springer Science & Business Media.

[2] Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: The finite volume method*. Pearson Education.

[3] Patankar, S. V. (1980). *Numerical heat transfer and fluid flow*. Hemisphere Publishing Corporation.