International Communications in Heat and Mass Transfer 75 (2016) 52–58

Contents lists available at ScienceDirect

International Communications in Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ichmt

Framework for simulation of natural convection in practical applications☆

Chung-Gang Li [a,b,*], Makoto Tsubokura [a,b], Rahul Bale [a]

[a] Complex Phenomena Unified Simulation Research Team, RIKEN Advanced Institute for Computational Science, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan
[b] Department of Computational Science, Graduate School of System Informatics, Kobe University, 1-1 Rokkodai, Nada-ku, Kobe, 657-8501, Japan

ARTICLE INFO

Available online 09 April 2016

Keywords:
Natural convection
Turnaround time
Preconditioning
Building cube method
Immersed boundary method
Massive parallelization

ABSTRACT

Natural convection is commonly used as a means of heat transfer in many practical products because it is highly cost-effective. The development of simulation tools for this type of application is generally accompanied by several critical issues, including high-temperature differences, rapid turnaround demand, and complex geometries. Under conditions of natural convection with high-temperature differences, the density of the medium is variable but the flow speed is low. Therefore, a compressible solver, i.e., Roe scheme developed by P.L. Roe in 1981, must be combined with a preconditioning method that can make the Roe scheme available at low speeds to allow the above situation to be addressed. The building cube method is adopted to make our method suitable for massive parallelization systems, which can reduce the calculation and turnaround times immensely. An immersed boundary method for compressible flows combined with a fast, easy to implement, and robust interpolation method is developed to handle flows with complex immersed geometries. The results show that the program described here is suitable for application to product design and analysis because of its wide applicability to natural convection with high-temperature differences, its capacity to handle complex geometries, and its feasibility for use in massive parallelization systems.

1. Introduction

The cost-effectiveness of natural convection has caused it to be widely adopted as the main means for heat transfer in a variety of electronic devices, such as light-emitting diodes. To enable the manufacture of better products, the use of computational fluid dynamics (CFD) to gain an understanding of natural convection for design and analysis applications appears to be a promising option. In these situations, new CFD programs with high-performance computing capabilities for rapid turnaround, dealing with complex geometries such as those of heat sinks and heat pipes, and obtaining accurate results, are in high demand.

To promote these computational capabilities, the use of exaflop-type supercomputers for product simulations to produce vast reductions in turnaround times is expected to occur within the next decade [1]. While CFD using unstructured grids is suitable for all three-dimensional complex geometries on such huge-scale systems, it is impractical because of the lengthy computation times required when adopting higher-order schemes and the inherent memory overheads. To solve these issues, Nakahashi [2] proposed an idea that involved the generation of a mesh based on structured grids, which he called the building cube method (BCM). A typical BCM configuration is shown in Fig. 1. The basic idea is to fill the entire computational domain with cubes and then feed each cube with the same number of grids. In this way, by assigning the same number of cubes to each CPU, the democratization of the computational cost for massively parallel computations is easily achieved and is accompanied by an obvious concomitant improvement in computational efficiency. In addition, in complex geometries, smaller cubes can be located near objects to increase the resolution near the wall and obtain more accurate results.

While using structured grids can reduce the computational overheads, accuracy, particularly that near the boundary, is an important consideration. In structured grids, a body boundary with complex geometry must be represented using a staircase-shaped boundary (Fig. 2), where it is clearly difficult to obtain accurate results. To solve this issue, the immersed boundary method (IBM) has been developed over the past few decades. Fadlun et al. [3] pioneered the use of the IBM on complex geometric structures. In their study, the finite difference method with second-order interpolation was used to mimic the boundary of an IC piston/cylinder; the results obtained were consistent with the experimental data. In an incompressible solver, Tullio et al. [4] combined the IBM, in which the interpolation is basically the same as that by Fadlun et al. [3], with a flexible local refinement technique from a compressible solver with preconditioning for a wide range of Mach numbers. The results showed that this method can achieve second-order accuracy. To address heat transfer problems using the IBM, Kim and Choi [5] followed their earlier work [6] by introducing a

☆ Communicated by W.J. Minkowycz.

* Corresponding author at: Complex Phenomena Unified Simulation Research Team, RIKEN Advanced Institute for Computational Science, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan.
E-mail address: chung-gang.li@riken.jp (C.-G. Li).

http://dx.doi.org/10.1016/j.icheatmasstransfer.2016.03.022
0735-1933/

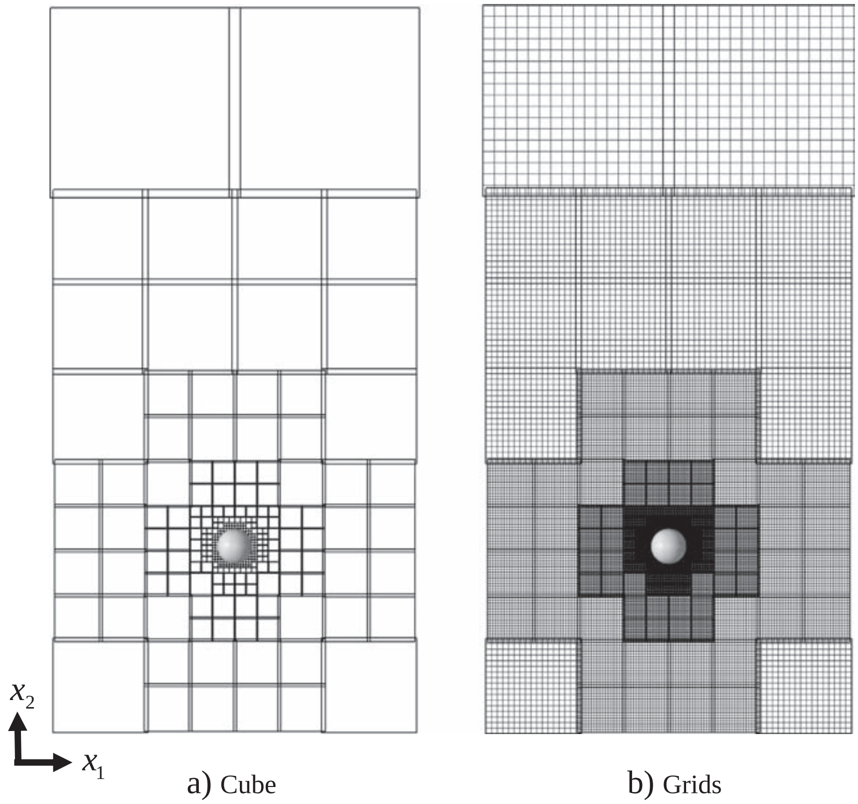

Fig. 1. BCM configuration. (a) Cube; (b) grids.

heat source/sink in the body to satisfy the iso-thermal and iso-heat-flux conditions. In their method, the value of the forcing term is calculated based on the surrounding points rather than simply being based on the points in the axial direction alone, as in the calculations performed by Fadlun et al. [3] and Tullio et al. [4].

Natural convection in practical applications generally is accompanied by large temperature differences. In such situations, the traditional method of using an incompressible solver with the Boussinesq approximation is not appropriate because the approximation is limited to temperature differences of less than 30 K [7]. Therefore, a compressible solver that can also be applied to the low flows induced by buoyancy is required. Weiss and Smith [8] adopted a preconditioning method to simulate natural convection through a two-dimensional annulus while involving a large temperature difference of 1000 K at $\mathrm{Ra} = 4.7 \times 10^4$. The results showed that this preconditioning method could reduce the

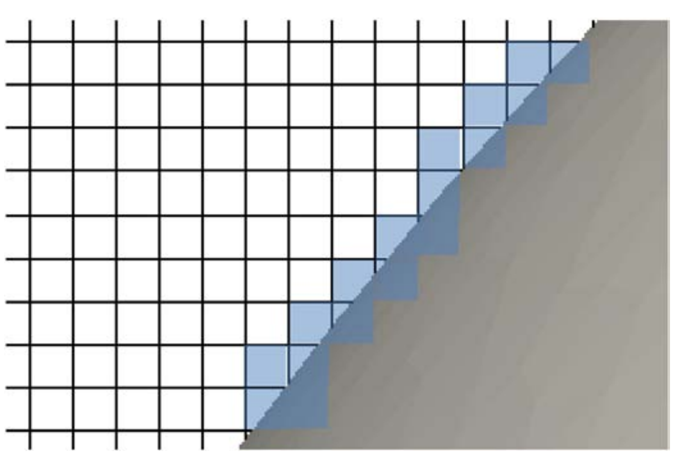

Fig. 2. Geometry represented by stair shape.

computational times required by a factor of 60. Fu et al. [9–11] used the preconditioning method to study the same problem in a channel, demonstrating the reliability and wide applicability of the method.

The aim of this study is to develop a sophisticated and robust program for natural convection analysis in practical products. The BCM [2] is used to make the program suitable for massive parallelization and high-performance computing implementation to reduce turnaround times. In addition, in practical applications, the temperature difference required for natural convection should be recognized as always being higher than the 30 K permitted by the Boussinesq approximation. Therefore, we use the Roe-scheme compressible solver [12], a preconditioning method, and dual time stepping [8], which enable the program to be applicable to the intended situation. In general, practical applications always come with complex geometry issues. A simple and robust interpolation technique for the IBM has been developed that is suitable for use with compressible flows and enables very thin structures such as fins to be treated.

2. Governing equations

To calculate the effects of the buoyancy force, the governing equations used here are the original Navier–Stokes equations with a source term,

$$\frac{\partial U}{\partial t} + \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3} = S. \tag{1}$$

The quantities U and F_i are

$$U = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e \end{pmatrix}, \tag{2}$$