Computers & Fluids 110 (2015) 245–253

Contents lists available at ScienceDirect

Computers & Fluids

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

Reordering of hybrid unstructured grids for an implicit Navier-Stokes solver based on OpenMP parallelization

Meng Cheng, Gang Wang *, Haris Hameed Mian

National Key Laboratory of Science and Technology on Aerodynamic Design and Research, Northwestern Polytechnical University, Xi'an 710072, PR China

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form 1 April 2014
Accepted 4 May 2014
Available online 13 May 2014

Keywords:
Unstructured hybrid grids
Grid reordering
LU-SGS scheme
CFD
Parallel computation

ABSTRACT

Grid reordering is an efficient way to obtain better implicit convergence speed in viscous flow simulation based on unstructured grids. When performing parallel computation on the shared memory machines, the convergence performance for solving high-Reynolds number flow with LU-SGS implicit scheme is ruined by the interface between different sub-domains divided by OpenMP parallelization. In order to improve the compatibility between OpenMP parallel environment and the implicit LU-SGS time-stepping scheme, a grid reordering method for unstructured hybrid grids is proposed. In this method, the structured-grid cells in the viscous layer near-body surface are reordered along the normal direction (like columns) and the unstructured part is reordered layer by layer according to the neighboring relations. To investigate the performance of the current implementation, turbulent flows around the RAE2822 airfoil, the NHLP-2D L1T2 multi-element airfoil configuration, the DLR-F6 wing–body–nacelle–pylon configuration and an aerospace plane has been simulated on unstructured hybrid grids. The numerical results show that the grid reordering method is an efficient and practical strategy for improving the convergence rate and the overall efficiency in the parallel computation with unstructured flow solver.

1. Introduction

With the fast development in the field of computational fluid dynamics (CFD), research topics in this area have become more and more complex. As a result, the computational requirements have increased so rapidly that even the existing computing technology is far from being able to meet this requirement. Parallel computing [1–3] has become an inevitable choice to perform these extensive numerical computations. The open multi-processing (OpenMP) [4–6] which is based on the shared-memory platform and the message passing interfaces (MPI) [7–9] based on massage-passing platform are the two key standards usually adopted for parallelization. In calculations based on the MPI environment, the mesh is parted into several sub-domains and allocated to different processors. The data at the interface passes between neighboring sub-domains. However, for OpenMP, which is based on the shared-memory platform, there is no need to part the mesh geometrically or transfer information between different sub-domains, which ensures load balancing, saves the communication costs and is simpler to be programmed. For complex geometries, the demands of rational domain decomposition are always challenges for the MPI technology. In comparison, OpenMP does not possess this problem. By contrast, MPI is mostly used on distributed memory systems. The hybrid MPI/OpenMP approach [10,11] is widely applied in hierarchical machine model, in which MPI is used for communication across distributed memory nodes and OpenMP is used for fine-grained parallelization within a node. However, for shared memory systems, OpenMP is more convenient and efficient than MPI.

At the same time, the complexity of the geometry augmented the use and development of the hybrid unstructured grid technology [12–14]. For unstructured hybrid grids, structured or semi-structured grid cells are utilized to resolve viscous boundary layers and unstructured-grid cells are employed elsewhere [15]. The use of hybrid grids combines the geometric flexibility offered by unstructured grids and the numerical accuracy of the structured grids. This meshing technique offers the potential of attaining a balance between mesh quality, efficiency, and flexibility.

Although the unstructured grids are flexible in their use but a negative factor is also associated with them. The data storage of unstructured grids is random, which has negative impacts on the convergence behavior of the computation.

To achieve a better cell order, grid reordering has been employed by many researchers and has been proven to be effective. Löhner [16] discussed several reordering strategies leading

* Corresponding author. Tel.: +86 029 88460753.
E-mail address: wanggang@nwpu.edu.cn (G. Wang).

http://dx.doi.org/10.1016/j.compfluid.2014.05.003
0045-7930/

to a minimization of cache-misses and an optimal grouping of elements for different computer platforms. Martin and Loehner [17] used a linelet-preconditioner for an implicit finite element solver to propagate information in a fast way to the boundaries and obtains rapid convergence rates. The lower–upper symmetric Gauss–Seidel (LU-SGS) time-marching method [18] is a very efficient method for structured grids and has been used by some authors with unstructured grids. Sharov and Nakahashi [19] proposed a grid reordering method which improves the balance between lower and upper matrices. When performing large-scale parallel computation on the shared memory machines, the order of the cells (edges for edge-based solvers) affects the task assignment and has a great influence on the parallel efficiency. Aubry et al. [20] presented reordering methods which guarantee that nodes belonging to one thread are not accessed by other threads for vertex-centered discretizations. Löhner [21] described renumbering techniques based on shared-memory, cache-based parallel machines to avoid cache-misses and cache-line overwrite and gained good results. They found computational efficiency on the shared memory machines was greatly improved by grid reordering.

The convergence performance for solving high-Reynolds number flow with LU-SGS implicit scheme is ruined by the interface between different sub-domains divided by OpenMP parallel environment. As the parallel thread count increases, the interface is enlarged and its side effect on the convergence performance becomes more serious. For the cells in boundary layer, which have high aspect ratios, the neighboring cells along the normal direction contribute much to the residuals and implicit system. If neighboring cells along the normal direction are computed in different parallel processors, the convergence rate will decrease significantly. With regard to this situation, we present a grid reordering method to avoid the side effect from parallelization.

The paper is organized as follows. Section two introduces the numerical methods used for computations and presents the grid reordering method in detail. In section three, the technique is tested for various two and three dimensional aerodynamic configurations. The simulation results prove the feasibility of the proposed method.

2. Numerical methods

The proposed method has been implemented in an in-house flow solver HUNS3D [22], developed for viscous flows based on hybrid unstructured meshes.

2.1. Governing equation

The integral form of non-dimensionalized three-dimensional unsteady Reynolds averaged Navier-Stokes (RANS) equations can be written as:

$$\frac{\partial}{\partial t}\iiint_{\Omega}\mathbf{Q}dV+\iint_{\partial\Omega}\mathbf{F}(\mathbf{Q})\cdot\mathbf{n}dS=\iint_{\partial\Omega}G(\mathbf{Q})\cdot\mathbf{n}dS \tag{1}$$

where Ω is the control volume; $\partial\Omega$ is the boundary of the control volume; $\mathbf{Q}$ is the conservative variable; $\mathbf{F}(\mathbf{Q})$ is the inviscid flux; and the right side is the viscous term. Using the cell-centered finite volume method, the semi-discretization form of Eq. (1) can be expressed as:

$$\Omega_i\frac{d\mathbf{Q}_i}{dt}=-\mathbf{R}_i(\mathbf{Q}) \tag{2}$$

where Ω_i represents volume of cell i, residual term $\mathbf{R}_i(\mathbf{Q})$ is the summation of inviscid and viscous flux terms on all faces of cell i.

2.2. LU-SGS scheme

Eq. (2) is a system of coupled ordinary differential equations in time. By using the backward Euler scheme for the implicit time integration, we obtain

$$\Omega_i\frac{\Delta\mathbf{Q}^{n+1}}{\Delta t_i}=\Omega_i\frac{\mathbf{Q}_i^{n+1}-\mathbf{Q}_i^{n}}{\Delta t_i}=-\mathbf{R}_i(\mathbf{Q}^{n+1}) \tag{3}$$

where the superscript n represents the number of time level. Since $\mathbf{Q}^{n+1}$ is unknown on current time level, the residual $\mathbf{R}_i(\mathbf{Q}^{n+1})$ cannot be evaluated directly. However, it can be linearized by using first order Taylor expansion in the following way:

$$\mathbf{R}_i(\mathbf{Q}^{n+1})\approx\mathbf{R}_i(\mathbf{Q}^{n})+\sum_{j\in C(i)}\frac{\partial\mathbf{R}_i(\mathbf{Q}^{n})}{\partial\mathbf{Q}_j^{n}}\Delta\mathbf{Q}_j^{n+1} \tag{4}$$

where $C(i)$ is the set of cell i and its neighbor cells. After $\mathbf{R}_i(\mathbf{Q}^{n+1})$ in Eq. (3) is substituted by the linearization term in Eq. (4), then the following implicit system is obtained:

$$\Omega_i\frac{\Delta\mathbf{Q}^{n+1}}{\Delta t_i}=-\mathbf{R}_i(\mathbf{Q}^{n})-\sum_{j\in C(i)}\frac{\partial\mathbf{R}_i(\mathbf{Q}^{n})}{\partial\mathbf{Q}_j^{n}}\Delta\mathbf{Q}_j^{n+1} \tag{5}$$

The HUNS3D flow solver uses an improved LU-SGS scheme for solving the above equation system, details of which could be found in Ref. [23]. By using the LU-SGS scheme, the expression becomes

$$\Delta\mathbf{Q}_i^{*}=\mathbf{D}^{-1}\left[\mathbf{R}_i^{n}-\sum_{j;j<i}\frac{1}{2}\left(\frac{\partial\mathbf{F}(\mathbf{Q}_j)}{\partial\mathbf{Q}_j}\cdot n-|\lambda_{ij}|\mathbf{I}\right)\Delta S\cdot\Delta\mathbf{Q}_j^{*}\right],\quad i=1,2,\cdots,N \tag{6}$$

$$\Delta\mathbf{Q}_i^{n}=\Delta\mathbf{Q}_i^{*}-\mathbf{D}^{-1}\sum_{j;j>i}\frac{1}{2}\left[\frac{\partial\mathbf{F}(\mathbf{Q}_j)}{\partial\mathbf{Q}_j}\cdot n-|\lambda_{ij}|\mathbf{I}\right]\Delta S\cdot\Delta\mathbf{Q}_j^{n},\ i=N,N-1,\cdots,1 \tag{7}$$

where j is the cell adjacent to cell i, λ_{ij} is the maximum eigenvalue of Jacobi matrix on the cell face and D is the diagonal matrix expressed as:

$$\mathbf{D}=\left(\frac{\mathbf{V}_i}{\Delta t}+\sum_{all\ face}\frac{1}{2}|\lambda_{ij}|\Delta S\right)\mathbf{I} \tag{8}$$

2.3. Grid reordering method

As seen from Eqs. (6) and (7), the LU-SGS scheme requires two sweeps: forward sweep through cell numbers from 1 to N and backward sweep in a reverse loop. In case of forward sweep (lower), summation for cell i is over all surrounding cells whose number is less than i. Backward sweep (upper) is summation over surrounding cells whose number exceed the current cell number. If some cells are surrounded by only those cells whose numbers are greater (lesser) than current cell number, the local iterations will degenerate from Gauss–Seidel iterations to Jacobi iterations [19]. In other words, the lower/upper balance of the method highly depends on grid numbering.

In this work, we employ static scheduling in the parallel computation for its advantages of lower scheduling overhead, less data race [24,25] and less cells at the interfaces between different sub-domains compared to dynamic scheduling and guided scheduling. By the static scheduling, the grids are divided averagely into M sub-domains in accordance with the cell indexes and allocated to M processors. According to the assignment characteristics of the static scheduling, we design the grid reordering method making the cells in the same processor more centralized in spatial position to reduce the number of cells at the interfaces. The reduction on