Journal of Computational Physics 229 (2010) 3523–3542

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Euler calculations with embedded Cartesian grids and small-perturbation boundary conditions

W. Liao [a,b,*], E.P.C. Koh [a], H.M. Tsai [a], F. Liu [c]

[a] Temasek Laboratories, National University of Singapore, Kent Ridge Crescent, Singapore 117411, Singapore
[b] Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, United States
[c] Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92697, United States

ARTICLE INFO

Article history:
Received 17 April 2009
Received in revised form 10 October 2009
Accepted 13 January 2010
Available online 18 January 2010

Keywords:
Cartesian grid
Embedded multigrid
Surface boundary
Least-squares approximation
Small-perturbation method

ABSTRACT

This study examines the use of stationary Cartesian mesh for steady and unsteady flow computations. The surface boundary conditions are imposed by reflected points. A cloud of nodes in the vicinity of the surface is used to get a weighted average of the flow properties via a gridless least-squares technique. If the displacement of the moving surface from the original position is typically small, a small-perturbation boundary condition method can be used. To ensure computational efficiency, multigrid solution is made via a framework of embedded grids for local grid refinement. Computations of airfoil wing and wing-body test cases show the practical usefulness of the embedded Cartesian grids with the small-perturbation boundary conditions approach.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

With complex configurations such as an aircraft with stores dealt with by structured, unstructured or hybrid grid methods, very significant effort is needed to create the initial grid and subsequently ensure the fidelity of the moving grid. The task of grid generation is to create just enough grid points to adequately resolve the significant flow features present. In the context of body-conformal curvilinear grid, generating good-quality grids can be challenging involving an iterative process where substantial human intervention is necessary. Inherently there are conflicting requirements in grid resolution and computational efficiency. Such requirements can give rise to grid skewness that negatively impacts the solvers accuracy and convergence. Though conceived to better handle complex geometries, the unstructured grid approach also needs considerable effort to ensure grid quality and careful prior surface grid preparation.

In contrast, geometric complexities have no particular effects on Cartesian grid quality. Cartesian grids devoid of any inherent skewness can be readily created. Non-body-conformal grids in the region near the surface are created by the solid boundary cutting through the Cartesian volume grids. The challenge is in the implementation of solid wall boundary conditions to achieve the necessary accuracy but the process can be automated using suitable algorithms. Modifying the equations along these cut cells is necessary. In addition, by using Cartesian grids one looses control of the grid resolution in the vicinity of the body. For instance, excessive mesh refinement is needed near curved boundaries such as the leading edges of wings. Nesting or embedment of Cartesian grids used in an overlapping manner for local refinement can alleviate this problem. When this embedded mesh is used in the form of a multigrid computation, rapid convergence rate for the solution can

* Corresponding author. Address: Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, United States. Tel.: +1 757 6836001x5035; fax: +1 757 6833885.
E-mail addresses: wliao@odu.edu (W. Liao), hm_tsai@science.edu.sg (H.M. Tsai), fliu@uci.edu (F. Liu).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.01.014

be achieved [1]. For these advantages, the CFD community has invested considerable attention to Cartesian methods [2–7]. The method is practical for inviscid flows, but is not really suitable for viscous flows unless it is at low Reynolds numbers.

Moreover, Cartesian grid method has the advantage of using significantly less computational resources. Compared to using body-conforming grids, terms involving grid transformations are not required in the computation. When compared to unstructured grids, grid connectivity information is obvious and thus incurs no additional storage requirements. The relative simplicity of structured Cartesian grid based solvers also means simpler and more efficient iterative schemes that exploit the structured nature of the problem, which can enhance the overall computational efficiencies. The computational efficiencies and the great ease in grid generation makes Cartesian grid a compelling approach.

Current computations related with moving boundaries typically take a direct approach involving deformable meshes. While efficient moving grid methodology is available (see Tsai et al. [8]), they may not be sufficiently robust to avoid degenerate grids when it comes to modeling complex geometries. The use of non-deforming Cartesian grid can by-pass the problems associated with generating a new grid and the need to project the solution onto this new grid at each time step. The direct approach would be to work out the new boundary conditions arising from the moving surface relative to the stationary Cartesian grid. However, the procedures to search and determine new cut cells at each time step make this computationally unattractive. An alternative approach for handling moving boundary conditions is the small-perturbation techniques [9–12]. Gao et al. [9] reported a small-perturbation boundary condition method for the Euler equations on non-moving Cartesian grids. The essential idea is to use the classical small-disturbance potential flow method, to approximate the solid wall boundary conditions for an airfoil by a first-order expansion on the airfoil mean line. Computations of both steady and unsteady problems involving even relatively thick airfoils and moderately large angles of attack show excellent accuracy for an unsteady problem. Strictly the method is limited to thin airfoil assumption and treatment of stores and fuselage is neither straightforward nor realistic with the method. Yang et al. [10,11] used stationary body-conforming grids, which did not require the assumption of thin geometry. Small-perturbation boundary conditions were not used to mimic the geometry. Instead they were used only to track the movements of two and three-dimensional surfaces such as the airfoil or wings. Kirshman et al. [12] also developed similar approach but on a Cartesian grids in their studies of two-dimensional airfoils. As applied to flutter simulations where the unsteady motion or deformation of the flow boundary is small with respect to the mean position, the small-perturbation approximation proves to be general and accurate. In Kirshman and Liu's two-dimensional Cartesian grid method [12], they made use of a set of shape functions over a cloud of gridless nodal points and incorporated the unsteady perturbation of the boundary conditions by a perturbation of the shape functions.

The aim of the present work is to address the problems in using Cartesian grids for computing complex static or moving geometries in three dimensions. A different approach is adopted in the present three-dimensional studies. The approach presented here makes use of a Cartesian grid for handling complex geometries and small-perturbation assumption to avoid the need for moving grid algorithm. Though possible, the direct extension of the Cartesian grid solvers with the gridless boundary condition of Kirshman et al. [12] and Koh et al. [5] for three-dimensional problems is not efficient. The present method of boundary implementation does not require the solution of the flow equation for boundary condition implementation and is direct and considerably simpler especially when applied in three dimensions. The approach makes use of reflected points or ghost cells. A gridless cloud of points is needed to obtain the variables for the reflected node via a least-squares approximation. It is found to be more efficient especially when used for three-dimensional problems. Euler fluxes for the neighbors of cut cells are directly computed using the reflected points which are set to satisfy the moving and non-moving wall boundary conditions. A finite-volume formulation with central differencing for the Euler equations is used throughout the rest of the flow field [13,14]. As will be discussed below to implement the small-perturbation method, the flow variables at the reflected points are selected such that the perturbation in the surface normal of the moving surface is taken into account. The inherent simplicity, accuracy, robustness, and computational cost of the proposed solution procedure, makes this an attractive approach for the envisioned application.

2. Numerical algorithms

2.1. Governing equations

The governing equation for the three-dimensional unsteady Euler equations in integral form is given as follows,

$$\frac{\partial}{\partial t}\int_V \vec{U}\,dv + \int_S \vec{F}\cdot n\,d\vec{S} = 0, \tag{1}$$

where V denotes a control volume with closed boundary surface S, and $\vec{n}$ is the outward normal vector on S. The dependent variable $\vec{U}$ and the flux vector $\vec{F} = f_x\vec{e}_x + f_y\vec{e}_y + f_z\vec{e}_z$ are given by,

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix}, \quad f_x = \begin{Bmatrix} \rho(u-u_b) \\ \rho u(u-u_b)+P \\ \rho v(u-u_b) \\ \rho w(u-u_b) \\ \rho H(u-u_b) \end{Bmatrix}, \quad f_y = \begin{Bmatrix} \rho(v-v_b) \\ \rho u(v-v_b) \\ \rho v(v-v_b)+P \\ \rho w(v-v_b) \\ \rho H(v-v_b) \end{Bmatrix}, \quad f_z = \begin{Bmatrix} \rho(w-w_b) \\ \rho u(w-w_b) \\ \rho v(w-w_b) \\ \rho w(w-w_b)+P \\ \rho H(w-w_b) \end{Bmatrix}, \tag{2}$$

Download English Version:

https://daneshyari.com/en/article/519502

Download Persian Version:

https://daneshyari.com/article/519502

Daneshyari.com