Available online at www.sciencedirect.com

ScienceDirect

Aerospace Science and Technology 12 (2008) 418–426

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Adaptive mesh refinement computation of acoustic radiation from an engine intake

Xun Huang [a], Xin Zhang [a,*], Simon Keith Richards [b]

[a] *Aeronautics and Astronautics, School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, UK*

[b] *GE Global Research Centre, Niskayuna, New York, NY 12309, USA*

Received 21 February 2007; received in revised form 19 September 2007; accepted 20 September 2007

Abstract

A block-structured adaptive mesh refinement (AMR) method was applied to the computational problem of acoustic radiation from an aeroengine intake. The aim is to improve the computational and storage efficiency in aeroengine noise prediction through reduction of computational cells. A parallel implementation of the adaptive mesh refinement algorithm was achieved using message passing interface. It combined a range of 2nd- and 4th-order spatial stencils, a 4th-order low-dissipation and low-dispersion Runge–Kutta scheme for time integration and several different interpolation methods. Both the parallel AMR algorithms and numerical issues were introduced briefly in this work. To solve the problem of acoustic radiation from an aeroengine intake, the code was extended to support body-fitted grid structures. The problem of acoustic radiation was solved with linearised Euler equations. The AMR results were compared with the previous results computed on a uniformly fine mesh to demonstrate the accuracy and the efficiency of the current AMR strategy. As the computational load of the whole adaptively refined mesh has to be balanced between nodes on-line, the parallel performance of the existing code deteriorates along with the increase of processors due to the expensive inter-nodes memory communication costs. The potential solution was suggested in the end.

Keywords: Adaptive mesh refinement; Computational aeroacoustics; Parallel computation; Aeroengine

---

1. Introduction

Stringent noise regulation requirements for modern aircraft have promoted research into efficient and accurate numerical methods capable of predicting aircraft noise, e.g. engine intake noise radiation. The physical process of noise generation and radiation is governed by the Navier–Stokes equations. At present, a full numerical solution of noise generation, propagation and radiation process using the Navier–Stokes equations is prohibitively expensive. However, certain aspects of the noise propagation and radiation process can be modelled by linearised equations. For example, in the duct upstream of the rotor-stator region of an aeroengine, where nonlinear and thermal noise generation effects are minimal, the propagation of the rotor-stator noise can be studied using the linearised equations about the mean flow. A significant amount of research has been undertaken to develop theoretical and computational methods to predict engine tonal noise propagation and radiation. However, the development of a cheap and quick computational method is still a challenging job. Of the three main numerical approaches for engine duct noise propagation and radiation problems, boundary element (BE) methods [12] are confined to problems of acoustics through uniform mean flows; finite/infinite element (FE/IE) methods [1] are generally restricted to acoustic propagation through irrotational mean flows; and computational aeroacoustic (CAA) methods based upon the Euler or linearised Euler equations (LEE) are much general in terms of governing physics [19]. However, CAA methods are more expensive. Realistic engineering applications of CAA methods call for continuous research into efficient computational schemes/methods.

AMR is efficient and effective in treating problems with multiple spatial and temporal scales [3]. It represents compu-

---

[*] Corresponding author. Tel.: ++44 (0)2380 592321.
E-mail address: x.zhang1@soton.ac.uk (X. Zhang).

1270-9638/$ – see front matter © 2007 Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.ast.2007.09.004

tational domain as hierarchal refinement levels and increases points per wavelength only in areas of interest. A given spatial error tolerance is achieved by recursively refining meshes. Subsequently a localised mesh of high grid resolution is distributed within an otherwise coarse mesh. The computational efficiency is improved by reducing the required number of computational cells. The operation of refinement could be operated either for each single cell [3], i.e. cell-structured AMR, or for each single block [13,25] and called block-structured AMR.

For the block-structured AMR method, a computational domain consists of blocks with a predefined number of cells, e.g. 4×4 cells in each block. If any cell in one block requires refinement, the whole block is refined [13]. As a result the data structure is only maintained for blocks. It is well accepted that block-structured AMR requires less programming efforts and is computationally more effective than cell-structured AMR with respect to communication costs and memory requirements. In this work we extend our earlier effort [10] where a block-structured AMR code was constructed and tested against benchmark problems on rectangular meshes. In order to solve aeroacoustic problems of practical significance, e.g. acoustic radiation from a general aeroengine intake, the current code is extended to support body-fitted meshes and works on parallel machines using message passing interface (MPI) library.

The major objective of this work is to apply the AMR strategy to study acoustic radiation from a generic aeroengine intake. The case is governed by LEE and computed on a mesh that is adaptively refined according to the magnitude of perturbation pressure gradients that reflects the sound propagation procedure. The accuracy of the prediction is compared with the earlier efforts [17–19]. Costs and the parallel speedup performance are given at the end of the paper.

2. The parallel AMR algorithm

On parallel machines, the existing AMR applications [4,13,21] generally employ a block-structured AMR algorithm. It involves a) representing the two-dimensional (2D)/three-dimensional (3D) hierarchical computational domain in blocks, b) connecting the generated blocks in a quadtree/octree data structure, c) estimating local truncation errors at all grid points and identifying blocks with excessive errors, d) regridding the identified blocks by superimposing or removing blocks to accommodate changes in flow physics, and e) redistributing computational load between processors to maintain dynamic load balancing. This procedure is operated recursively until either a given refinement/coarsening level is reached or a predefined local truncation error level has been met. After regridding, the initial conditions of the newly generated blocks are inherited from their base blocks. This operation is referred to as the AMR prolongation operation. Conversely, after each computing step, the solutions on the finer blocks should be used to update the solutions of the corresponding base blocks to maintain the desired accuracy. This is known as the AMR restriction operation. To solve partial differences of cells located near a block boundary, an extra area surrounding each block is required. This operation is referred to as the ghost construction in the following description.

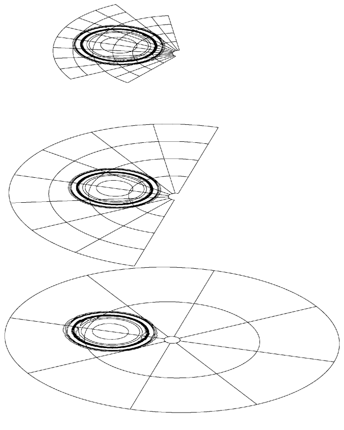

Fig. 1. Block-based AMR for an acoustic scattering problem. Solid lines denote blocks boundaries. Each block contains 21×21 cells.

In Fig. 1 a schematic of the block-structured AMR method employed in this work is given. The example is a benchmark problem of acoustic scattering [22] solved with body-fitted multi-block AMR. The display style is chosen to illustrate the hierarchical structure of the adaptively refined mesh more clearly. In reality meshes on the fine levels are superimposed on the coarse meshes. In this example, a nested mesh consisting of three refinement levels is created at the start of the computation. The refinement ratio between the consecutive coarse and fine levels is two. The AMR regridding operation defines the relationships between the blocks as parents/children or sibling according to the type of their connection, stores the hierarchy information in the data structure of quadtree, and either refines or coarsens the hierarchical meshes based on the gradients of perturbation pressure. As the simulation progresses, the meshes are dynamically updated to reflect the evolving solution. The prolongation operation provides the initial solutions on the newly generated blocks, and the restriction operation updates the solutions on the coarse blocks.

The essential algorithm of the regridding operation is roughly the same in the existing programs [4,13], mainly consists of traversing in blocks to generate new blocks or to delete excessive blocks, and maintaining the corresponding data structure to reflect the volatile connection relationships between blocks. Nevertheless, the parallel strategies of the other AMR operations, i.e. prolongation, restriction and ghost construction are generally different. Fig. 2(a) illustrates the potential situations, where unidirectional communications are operated to transfer the whole solutions on the related blocks for prolongation and restriction, whereas bidirectional communications are used to transport a part of solutions on the corresponding blocks for ghost construction. To achieve high efficiency, separate communication subroutines were designed specifically for these AMR operations in the existing AMR programs [4,13], which consequently tend to be complex and inaccessible.

A simplified AMR library is developed as a starting point for our work in the area of CAA. To realise the parallel AMR operations as simply as possible, the essential decision is to combine the parallel communication of the AMR operations together