



# Parallel implementation of large-scale CFD data compression toward aeroacoustic analysis



Ryotaro Sakai <sup>\*</sup>, Daisuke Sasaki, Kazuhiro Nakahashi

Department of Aerospace Engineering, Tohoku University, 6-6-01 Aramaki-Aza-Aoba, Sendai 980-8579, Japan

## ARTICLE INFO

### Article history:

Received 13 September 2011

Received in revised form 15 April 2012

Accepted 17 April 2012

Available online 26 April 2012

### Keywords:

Data compression

Large-scale computation

Discrete wavelet transform

## ABSTRACT

A data compression method was developed for large-scale, unsteady flow simulation data. The method includes three steps of discrete wavelet transform, quantization and entropy encoding. The discrete wavelet transform extracts only important features in the flow field from the entire simulation data. In the quantization process, high quantization bit rate is assigned only to the important region extracted by the previous transform. Finally, entropy encoding reduces the quantized data size. This data compression method was implemented to unsteady incompressible flow simulation data obtained by a block-structured Cartesian mesh method named Building-Cube method. The framework of the Building-Cube method enables not only easy implementation of data compression but also highly efficient parallelization by a very simple scheme for load balance. Numerical test cases demonstrated that the present compression method gave a high compression ratio with good quality of compressed data in terms of velocity distributions, turbulent statistics, and aeroacoustic analysis, as well as high scalability in parallelization.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

Computational Fluid Dynamics (CFD) has been greatly progressed for the past several decades, and now it plays an important role as one of engineering design and analysis tools. Today, unstructured-mesh CFD [1] is widely used because of its flexibility to treat complex geometry, but it still has some challenges toward large-scale, high fidelity simulation. Efficient computation on a massively parallel computer is not straightforward because of the irregularity of the mesh, and the spatial accuracy is generally limited to second order at the most. To overcome these issues, a block-structured Cartesian mesh approach named Building-Cube Method (BCM) proposed [2,3]. In this method, an equally-spaced Cartesian mesh is employed to keep the simplicities of all the stages in CFD including the pre-processing, numerical computation, and post-processing. These simplicities reduce the computational operation and memory requirement per node, which will be more important for large-scale computation in the future. In the framework of BCM, the mesh generation in the pre-processing is fast and robust even for complex geometries [4], and numerical computation with 0.2 billion mesh points shows high vectorization and parallelization ratio [5].

Although large-scale computation is successfully implemented in BCM, we encountered other problems in the post-processing. In the computational case where 0.2 billion mesh points were used,

the data size per time step amounted to 0.8 GB for each flow variable, even in single precision format. Such huge data require more rendering time in visualization and more processing time for data transfer. They also require much amount of space in data storage. These problems would become much more serious especially in unsteady flow analysis which requires time-series data along multiple time steps. One of unsteady flow analysis cases using the framework of BCM is the aeroacoustic noise prediction of landing gear [6]. Sound generation is an unsteady phenomenon, therefore unsteady flow simulation is necessary for aeroacoustic analysis. In [6], the flow around a landing gear model with approximately 90 million mesh points was computed, and total 2816 instantaneous pressure data on the body surface were used for aeroacoustic analysis. In this case, assuming that the pressure data in the entire computational domain is stored, the total data size would amount to 1 TB in single precision format. The output data size and its handling will inevitably become a major issue in larger-scale, unsteady flow simulation in the near future. The data size problem is not specific to BCM but will appear in all numerical simulations.

As one of signal processing tools, wavelet has attracted much attention and has been used in many engineering fields. In the area of CFD, various wavelet applications have been proposed such as grid refinement [7], shock detection [8], vortex extraction [9], and data compression. In particular, Lee et al. proposed a CFD data compression method [10] using supercompact multiwavelets [11] and vector thresholding techniques. This compression method gives a high compression ratio and low error because of the use

<sup>\*</sup> Corresponding author. Tel.: +81 22 795 6981; fax: +81 22 795 6979.

E-mail address: [sakai@ad.mech.tohoku.ac.jp](mailto:sakai@ad.mech.tohoku.ac.jp) (R. Sakai).

of supercompact multiwavelets. However, the use of supercompact multiwavelets requires additional pre- and post-transforms of the original data and extra memory for the operation. In addition, the data size for the compression is up to  $O(10^6)$  mesh points. In compressing the huge amount of data obtained in the framework of BCM, the simplicity of the compression process should also be kept as the important advantage.

The objective of this study is to develop a data compression method for large-scale, time-series data for use in aeroacoustic analysis. Although there are several data reduction/compression methods proposed ever, using the flow field topology [12] and the singular value decomposition [13], we focus on the simplicity of the compression process for easy handling of huge data, and the visualization property of the compressed data. To achieve both properties, we follow the image encoding format JPEG 2000 [14], and employ discrete wavelet transform (DWT), quantization, and entropy encoding. In addition to the simplicity of the compression process, high parallel efficiency in a multicore environment is also required. By making use of the computational structure of BCM, we propose a simple but effective scheme for efficient parallelization. Through two cases of data compression with  $O(10^7)$  mesh points, the usefulness of the proposed method in large-scale data compression is demonstrated.

## 2. Building-Cube method

BCM is based on block-structured Cartesian mesh. The flow field is divided into many blocks of cuboids, named “Cube”, as shown in Fig. 1a. Each cube is a sub-domain of the original computational domain, and has an equally-spaced Cartesian mesh named “Cell” in itself, as shown in Fig. 1b. Although the geometrical size of cube becomes large as the cube gets away from the object, all the cubes have the same number of cells regardless of their cube size. This means the computational resolution is determined by the size of each cube, and the computational cost is ideally the same among all the cubes. This approach enables (1) quick and robust mesh generation around complex geometries, (2) easy introduction of a higher-order scheme in the numerical computation, (3) easy introduction of adaptive mesh refinement by changing the size of the cube, (4) efficient parallelization of the numerical procedure based on cube, and (5) easy treatment of output data in the post-processing.

Here size level is assigned to each cube according to its size, which means a factor of scale-up. Level 0 is assigned to the smallest cube in the computational domain, and level  $n$  is assigned to the cubes whose size is  $2^n$  times larger than that of the smallest cube. The size level becomes large in the same way as the cube size does, as shown in Fig. 1c. These levels are used below to determine the degree of data compression.

## 3. Data compression method

### 3.1. Discrete wavelet transform

Discrete wavelet transform is applied to a series of discrete signal for the purpose of analysis or compression of its signal. The transform divides the input signal into two parts: the approximation part and the detail part. The approximation part contains the low frequency component of the original data, and the detail part contains high frequency component. The high frequency component is not as important for visualization as the low frequency one, thus discarding the high frequency component leads to lossy data compression.

As an example of DWT, Fig. 2 shows the original and transformed two-dimensional flow field data. Here original data

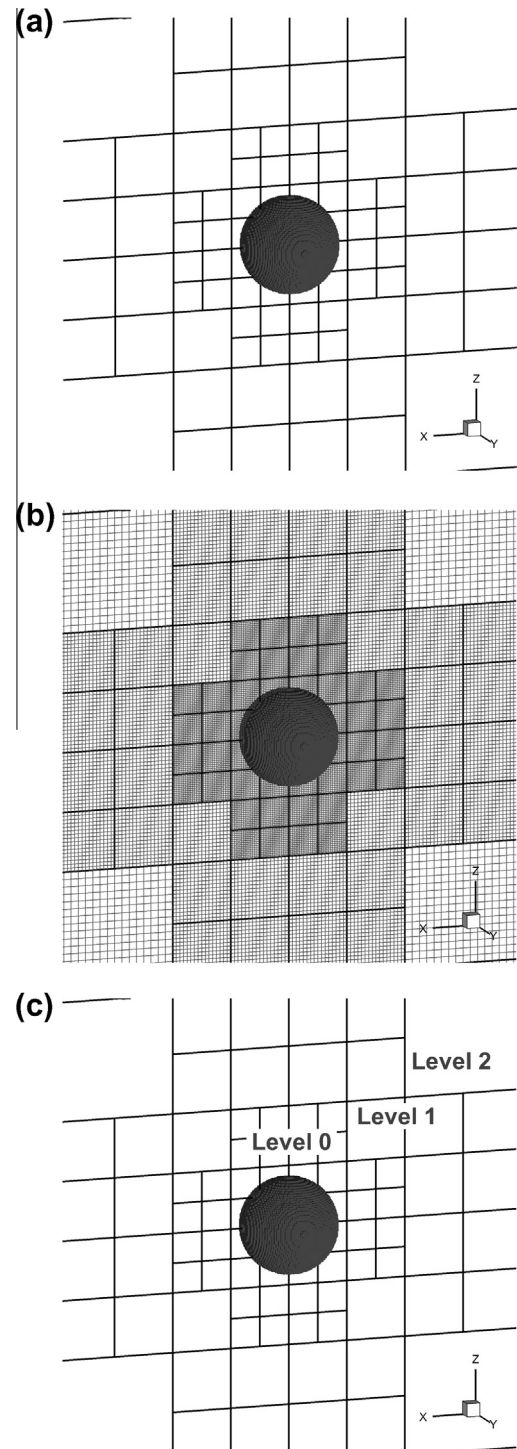


Fig. 1. Dividing scheme of the Building-Cube method; (a) cube, (b) cell, (c) cube size level.

(Fig. 2a) shows the streamwise velocity distributions around a circular cylinder. In the transformed data, flow field is divided into one approximation part (Fig. 2b) and three detail parts (Fig. 2c–e). These four parts have a quarter of mesh points of the original data, thus they can be treated together as one flow field that has the same number of mesh points as the original data (Fig. 2f). The approximation part shows velocity distributions similar to those of Fig. 2a. In contrast, the detail parts represent distributions that have nearly zero value, indicating that their impacts on the

Download English Version:

<https://daneshyari.com/en/article/756591>

Download Persian Version:

<https://daneshyari.com/article/756591>

[Daneshyari.com](https://daneshyari.com)