Computers and Structures 87 (2009) 27–38

Contents lists available at ScienceDirect

Computers and Structures

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

An efficient 3D mesh generator based on geometry decomposition

Konstantinos E. Kakosimos *, Marc J. Assael

Thermophysical Properties Laboratory, Chemical Engineering Department, Aristotle University, University Box 453, GR 54124 Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 5 February 2008
Accepted 22 September 2008
Available online 1 November 2008

Keywords:
3D Mesh
Urban
Decomposition
HeMUT
Hexahedral

ABSTRACT

A new, efficient 3D mesh generation algorithm, hexahedral mesh for urban terrains (HeMUT), is presented. HeMUT is developed under .NET and builds unstructured/structured hexahedral meshes. The algorithm focuses on urban terrains and on the mesh generation for the simulation of toxic gases dispersion (finite element). HeMUT is fully automated, multi-threading and takes advantage of terrain. In addition, it distributes the nodes on the domain by employing a method based on process design considerations. These features decrease the computational effort and at the same time differentiate this algorithm from all other similar ones. In comparison with a commercial software HeMUT performed well.

1. Introduction

The proposed mesh generation algorithm (HeMUT) has been developed by Kakosimos et al. [1] to be used in conjunction with a new computer software [2] (TGD – toxic gas dispersion), which was created for the calculation of toxic gases dispersion in urban terrains in local scale. TGD implements the finite element method and solves the Navier–Stokes equations for the wind flow and the advection/diffusion equations for the dispersion of toxic gases. TGD is developed as a risk management and assessment tool for toxic releases in chemical warfare. The extraordinary lethality of modern chemical weapons requires immediate decisions and reactions. This means that TGD has to favour speed against accuracy, without decreasing the quality of the calculations.

It is generally known that the FEM using a quadrilateral mesh is more accurate than that of a triangular one. Masud et al. [3] confirmed the usefulness of the quadrilateral meshes by comparing the numerical errors of the advection–diffusion equation with one of triangular meshes. Their work verified that errors of convergence rate in the case of quadrilaterals were smaller than with triangles. However, the generation of quadrilaterals of a complicated system is more demanding [4]. The most important drawback of FEM analysis is the need to discretize an arbitrary geometry into a valid finite element mesh; usually requires large CPU times and extensive user interaction. Focusing on a specific geometry, as TGD focus on gases dispersion in urban terrains, allows a more flexible approach regarding the mesh generation. Despite a large number of meshing methods developed in the last decades, algorithms still tend to be complex and consume large CPU time even for simple geometries. This is the reason why a new mesh generator for urban terrains was developed, instead of using an existing one that is applicable in an arbitrary domain. The proposed mesh generator forms hexahedral elements, in a hybrid structured/unstructured grid.

In general there are two types of computational grids structured and unstructured [5]. Structured grids are produced over a predefined block structure and the nodes are numbered in a pre-ordered sequence. For a structured mesh, all interior nodes have an equal number of adjacent elements. Unstructured meshes on the other hand, are generated over the entire domain and/or predefined blocks and their nodes are numbered arbitrarily. An unstructured mesh allows any number of elements to meet at a single node. While both structured and unstructured meshes can take regular and irregular shapes, the latter often produce element shapes that are more versatile in better filling the computational domains. There are many surveys [6] that list the available mesh generators and classify them according to their approach [7] into direct and indirect methods:

- In a direct scheme, quadrilateral elements are generated either individually or in patches. One approach is the advancing front technique, paving [8] or an enhanced version of paving [9]. Another approach is the geometry decomposition, which is based on decomposing a domain into simpler regions. The decomposition has been proposed many times in the past [10–15] and it is the method employed in the present work.
- In indirect approaches, the quadrilateral mesh is obtained through combining and splitting triangles. A usual method

* Corresponding author. Tel.: +30 2310996170.
E-mail address: kkakosim@cheng.auth.gr (K.E. Kakosimos).

0045-7949/$ - see front matter
doi:10.1016/j.compstruc.2008.09.002

is the combination of triangles to form a quadrilateral [16,17]. These, triangles are formed following the placement of nodes in the domain. Such an example is the Q-morph algorithm, which was introduced by Owen et al. [18].

It should however be pointed out, that there are methods that use elements of both approaches, like the one proposed by Park [19], which generates quadrilateral elements in a domain with line constraints.

1.1. The new algorithm

This paper presents a mesh generation algorithm for hexahedral finite elements for urban terrains (HeMUT) that improves upon the basic concept of 2D recursive domain decomposition as proposed by Nowottny [11], Sarrate et al. [14] and as used by Bastian [15]. The algorithm is implemented using parallel programming, an efficient algorithm for nodes distribution and an appropriate extrusion method to build the 3D mesh. Direct implementation of Sarrate and Huerta's algorithm frequently results in poor meshes [15] that in many cases require a make-up stage to repair. Later, Bastian [15] composed an improved version by building an object-oriented algorithm and by adding a set of closure techniques (similar to that of Nowottny [11]) to deal with misshaped sub-domains. To decide the next step of the decomposition process and ensure its applicability various tests are performed for all the possible combinations of pairs of nodes. It is obvious that the necessary computational time increases drastically for large number of nodes and complicates its adoption to parallel applications. These drawbacks can basically render the algorithm impractical for the purpose of large terrains meshing and demanding adaptive numerical computations. To overcome these difficulties, the procedures of geometry decomposition and of mesh generation were separated. Then an efficient technique distributes the final mesh nodes on the boundaries (internal and external) of each sub-domain at the final decomposed domain. In more detail, the employed technique calculates the number of nodes on each boundary by taking in consideration the predefined density of nodes on key boundaries (e.g. walls) and an even (not odd) total number of nodes per sub-domain. Finally, the mesh generation is based on the new grid of nodes. These additional critical improvements, combined with the free-from-human-interference original scheme, allow the mesh generation algorithm to become more efficient and applicable to larger geometries.

Compared with the existing algorithm, the present scheme represents a new and different concept for quadrilateral mesh generation. Instead of repeatedly validating a large number of conditions per sub-domain, this present scheme decomposes the geometry with a coarse resolution (adequate to describe the details of the geometry) and then calculates the desired density and place of the nodes. It should be mentioned that the mesh generation is fully automatic and completely free from human interaction, which is essential for adaptive finite element applications, parallel execution and practical computations (e.g. operational study of atmospheric dispersion of toxic gases). Furthermore, the new algorithm takes in consideration the sensitivity problems close to walls [20] that numerical simulations face when dealing with turbulent flows around obstacles. Around each obstacle (e.g. building) a fully structured mesh is formed and the size of the mesh can be determined by the flow characteristics when the mesh generator is used in conjunction with a CFD software.

Also the algorithm has been extended to the three dimensions. The urban terrain is divided into horizontal layers by grouping the existing buildings by height. Using as base the 2D mesh that was formed for the lower layer (ground level) the hexahedral elements are formed bottom-up and layer by layer. This approach boosts the performance of the algorithm (both speed and mesh quality) despite the fact that limits its application to the specific type of geometries.

Another improvement comes from the use of multi-threaded programming, which provides substantial computing and programming advantages. The new algorithm is also object-oriented (like C++ of Bastian [15]), but was developed under .NET, which takes advantage of the "multi-core" modern home computers. The discussion of .NET implementation has not yet been made, to our best knowledge, on mesh generation for finite element applications. So far, most mesh generation code has been developed using the traditional FORTRAN language or C. The former is, even today, a natural choice from the point view of continuity. Unfortunately, such a natural choice may not necessarily be the optimal choice. In fact, the use of a modern, object-oriented and multi-threaded language is more desirable for the improved meshing scheme and for the adoption by young researchers. Because this language allows the different parts of the mesh to be described naturally [15] it allows the parallel execution of the mesh generation routines in a common personal computer.

To sum up, the aspects where HeMUT makes the difference, compared to other mesh generation algorithms are the following:

(i) HeMUT focuses on urban terrains. This means that the algorithm takes advantage of the characteristics of the urban terrain, that is, the low geometrical complexity on the vertical dimension. For instance Fig. 1a illustrates the top view of an arbitrary urban terrain where the complexity of the geometry can be noticed, while in Fig. 1b, a side view of a city, the buildings can easily be grouped according to their height and form a distinct number of individual layers.
(ii) The geometry decomposition and the mesh generation take place in different stages to minimize the computational effort. In between, an integrated method calculates the number of nodes on internal and external boundaries. These boundaries are formed after the decomposition of the domain. The method originates from the process design industry and it is used to obtain a solution algorithm for large systems of equations.
(iii) HeMUT imposes a mixed structured (close to boundaries) – unstructured (interior areas) 3D mesh, and employs a fully automatic algorithm. Furthermore, the algorithm operates in parallel threads to decrease the computational time.

Finally, it should also be noticed that HeMUT uses a custom data format for input and output, which is closer to geographical information system (GIS) format. This format was preferred because geometrical data for cities are stored in GIS format, that it employs polygonal areas with specific properties e.g. height, in contrast to computer aided design (CAD) format like .iges, which are employed for geometrical data for fluid dynamics problems. Furthermore, to deal with not flat terrains the mesh generator deforms the constructed mesh, by moving the nodes vertically in order to adopt the hills and valleys of the terrain. However this module, is currently under testing and thus no more details are presented here.

2. Description

The proposed algorithm is developed in four main modules (Fig. 2):

(i) the decomposition of the topology and of the geometry,
(ii) the distribution of nodes on edges (internal and external boundaries),