VNU Journal of Science, Mathematics - Physics 26 (2010) 93-106

Development of a software package for 3D structured mesh generation

Duong Ngoc Hai, Nguyen Tat Thang*
Institute of Mechanics, Vietnam Academy of Science and Technology (VAST)

Received 3 April 2010

Abstract. A 3D structured-mesh generation package has been developed using the transfinite interpolation (TFI) and/or the elliptic mesh generation methods to generate 3D structured meshes for the computational domain surrounding 3D regions (objects). A boundary stretching coefficient was used in the first method. And an attraction function was used in the second one. These stretching coefficient and attraction function are used to control the properties, i.e. the distribution density of the mesh points, of the meshes to be generated. The mesh generation function of the package is tested and shows its robustness. In addition, a friendly graphic user interface has also been designed and developed to assist in viewing and presenting the topographic data and the generated meshes. The package has been applied to the generation of 3D computational meshes used as the input of a computational fluid dynamics model for simulations of turbulent compressible atmospheric flows and air pollutant transport/dispersion in practical 3D domains.
Keywords: Mesh generation (Grid generation); Transfinite interpolation (TFI); Elliptic mesh (grid) generation; Graphic user interface.

1. Introduction

In computational fluid dynamics (CFD), mesh generation techniques for generating computational meshes used in numerical models have long been developed [1] (in scientific literature, the technical terms, computational mesh and grid, are used interchangably). There are two types of computational meshes, structured meshes and unstructured ones, used in the discretization of the governing equations. The structured meshes whose connectivity between meshes is regular and fixed are used widely in finite difference methods. On the other hand, unstructured meshes are usually exploited in the finite element method (FEM) or finite volume method (FVM) [2]. In this study, we have investigated two methods of 3D structured mesh generation, i.e. algebraic and elliptic mesh generation methods, and developed a software package for 3D structured mesh generation for computational domains surrounding 3D objects (topographies) rising from a plain surface (considered to be the bottom plain). The output of this package, i.e. 3D computational meshes, is directly used as the input of a computational fluid dynamics software developed in the Institute of Mechanics, VAST, for the simulations of atmospheric turbulent compressible flows coupled with the transport/dispersion of pollutants [3]. This package has been developed with a handy graphic user interface in the Windows working environment that allows users to view and present the input data (digitized contour lines), and the generated meshes in any 2D vertical or horizontal planes. This software package has already been in practical use in some research in the Institute of Mechanics, VAST [4].

* Corressponding author: Tel.: (+84) 983384692
Email: ntthang@imech.ac.vn

2. 3D structured mesh generation

2.1. The transfinite interpolation (TFI)

The transfinite interpolation [1] is a common algebraic method used in structured mesh generation. For a generalized cuboid in 3D real space (Fig. 1), it is written as below:

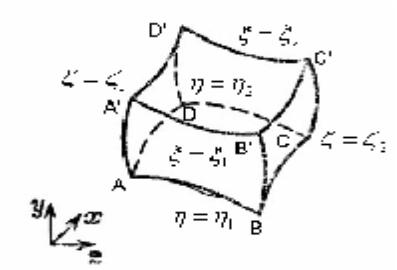

Fig. 1. A generalized cuboid in 3D real space.

$$
\begin{aligned}
&X(x,h,z) = SX(x,h,z) - \\
&\left[\begin{array}{l}(1-s_x)(1-s_h)X(x_1,h_1,z)+s_x(1-s_h)X(x_2,h_1,z)+\\ (1-s_x)s_hX(x_1,h_2,z)+s_xs_hX(x_2,h_2,z)\end{array}\right]- \\
&\left[\begin{array}{l}(1-s_h)(1-s_z)X(x,h_1,z_1)+s_h(1-s_z)X(x,h_2,z_1)+\\ (1-s_h)s_zX(x,h_1,z_2)+s_hs_zX(x,h_2,z_2)\end{array}\right]- \\
&\left[\begin{array}{l}(1-s_z)(1-s_x)X(x_1,h,z_1)+s_z(1-s_x)X(x_1,h,z_2)+\\ (1-s_z)s_xX(x_2,h,z_1)+s_zs_xX(x_2,h,z_2)\end{array}\right]+ \\
&\left[\begin{array}{l}(1-s_x)(1-s_h)(1-s_z)X(x_1,h_1,z_1)+\\ (1-s_x)s_h(1-s_z)X(x_1,h_2,z_1)+\\ (1-s_x)(1-s_h)s_zX(x_1,h_1,z_2)+\\ (1-s_x)s_hs_zX(x_1,h_2,z_2)+\\ s_x(1-s_h)(1-s_z)X(x_2,h_1,z_1)+\\ s_xs_h(1-s_z)X(x_2,h_2,z_1)+\\ s_x(1-s_h)s_zX(x_2,h_1,z_2)+s_xs_hs_zX(x_2,h_2,z_2)\end{array}\right]
\end{aligned}
\qquad (1)
$$

where x , h , z are the coordinate components in the parametric space; X is the coordinate on the Ox coordinate line of a point in the 3D real space. Here X is a function of the coordinates in the parametric; s_h is a function of the h and must satisfy the conditions $0 \le s_h \le 1$; $s_{h(ABCD)} = 0$; $s_{h(A'B'C'D')} = 1$; the functions s_x and s_z are similarly straightforward. The equation (1) is just an one to one inverse mapping between the 3D real computational domain and a rectangular domain in the parametric space [1, 2, 5, 6].

For short, the formulae for Y and Z coordinates are the same and they are not written here.

In this method, to control the density of the distribution of the mesh points, an algebraic stretching formula is used as the following: $\Delta x_1 = \dfrac{(b-a)(1-l)}{(1-l^n)}$; $(\Delta x)_n = l^{n-1}\Delta x_1$ where l is the stretching coefficient; a and b are the coordinate values at the two ends; n is the index of the mesh segment on the Ox coordinate line [1, 2, 5, 6].

The solution of the equation system is straightforward without the need to solve any partial differential equations.

2.2. The elliptic method

The elliptic mesh generation method is based on the solution of a system of elliptic partial differential equations. The TFI method may sometimes generate meshes that mesh lines cross each other which causes wrong computational meshes. The elliptic method does not. That is why the elliptic method is used as an improvement of the algebraic method [1, 2, 5-8]. However it may be an expensive selection and comments will be given later on.

The elliptic mesh generation method usually exploits the Thomson-Thames-Martin (TTM) equation system which is shown below:

$$\begin{aligned} &g^{11}x_{xx} + g^{22}x_{hh} + g^{33}x_{zz} + \\ &2g^{12}x_{xh} + 2g^{13}x_{xz} + 2g^{23}x_{hz} + \\ &g^{11}P_1x_x + g^{22}P_2x_h + g^{33}P_3x_z = 0 \end{aligned} \tag{2}$$

where x is a function of the coordinates in the parametric space; P is a stretching function; g^{ij} is the components of the G matrix which $G = J^T J$; J is the Jacobian matrix.

The equations for y and z coordinates follow similarly.

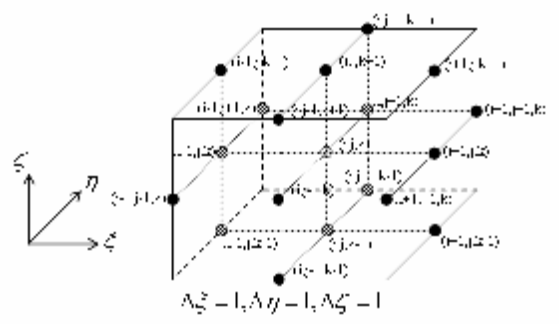

Fig. 2. Discretized mesh points in the discretization of the TTM equations.

Applying some kinds of basic approximations to the TTM partial differential equations (2) we get to an explicit formula in the form *x(i,j,k)=F[x(i',j',k')]* where *x(i',j',k')* and *x(i,j,k)* must not be the same points (Fig. 2). An iterative solution method in the form $x^{n+1}_{(i,j,k)} = F\left[x^{n}_{(i',j',k')} \right]$ is applied to solve the discretized equations (Poisson-type equations). In that method, a relaxation coefficient is exploited in the form $x^{n+1}_{i,j,k} = l\, x^{n}_{i,j,k} + (1-l)\, x^{n+1}_{i,j,k}$ where $0 < l < 2$, to speed up the convergence process of the solution procedure. The initial condition of the iterative method is a mesh obtained by the algebraic mesh generation method shown above [5, 6, 9].

Some remarks should be mentioned on the selection of the two mesh generation methods briefly presented above. First the algebraic mesh generation method is selected due to its simplicity in its nature. The implementation of the method is straightforward. There is no iterative process used, the CPU time needed for mesh generation using this method is therefore relatively small. In addition, the method is exploited to generate initial mesh for the elliptic method. However, in this method, it is difficult to control the quality of the generated meshes. The output meshes are sometimes wrong since

mesh lines may cross each other. Users need to check the output meshes carefully before using them in any further simulations. The meshes are also usually distorted at the areas close to the boundaries of the computational domain (low orthogonality at the boundaries). Moreover this distortion property propagates far into the computational domain.

In contrast, the elliptic mesh generation method is relatively more complicated. Moreover, the CPU time needed for iterative procedure for the solution of the Poisson-type equation is large. Sometimes, the convergence of the solution process can not be achieved. In those cases, the input parameters need to be adjusted. However, the quality of the generated meshes is much better than that of the meshes generated by the algebraic method. Due to the nature of the elliptic method, the better orthogonality of the generated meshes can be achieved and the distortion close to the boundary surfaces does not propagates very far.

The difference in mesh quality of the two types is clearly seen in the resulted meshes of three selected cases presented in Part 4 of this paper.

To be flexible, both methods are investigated in this study and are implemented in the software package to provide two mesh generation options. Based on their specific need, the users will decide which method is the most suitable one which is applicable for their applications.

3. The program flowcharts

The flowcharts below, which correspond to the two methods mentioned above, of the 3D structured mesh generation package are implemented using Visual Basic 6.0 programming language in the Windows working environment.

3.1. The flowchart of the algebraic mesh generation method

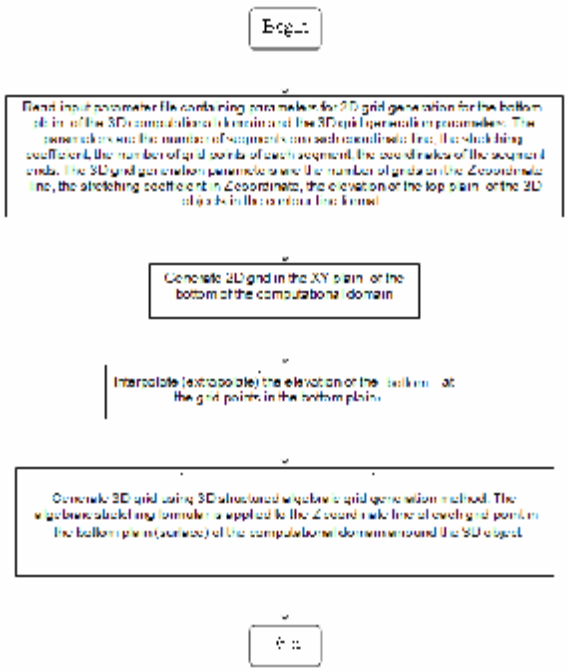

Fig. 3. The flowchart of the program for 3D structured algebraic mesh generation.

This flowchart is implemented in the program as one of the two options, i.e. algebraic mesh generation and elliptic mesh generation, in the package.

In addition, this flowchart is also integrated in the elliptic mesh generation method to be the step to initialize the initial condition of the elliptic method.

3.2. The flowchart of the elliptic mesh generation method

In the flowchart (Fig. 4), the error and maximum iteration step is pre-determined and input through the software package interface.

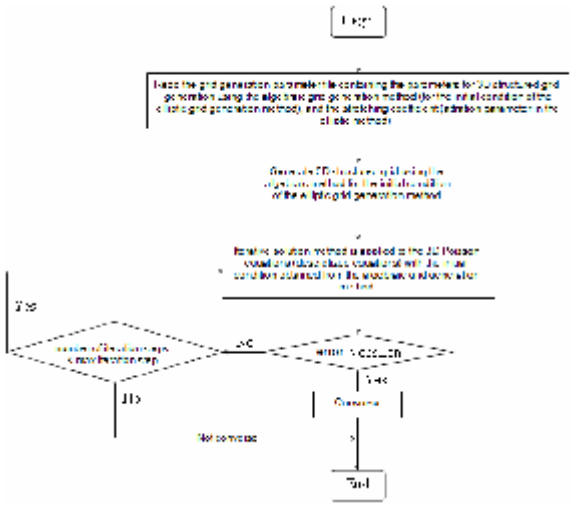

Fig. 4. The flowchart of the program for 3D structured elliptic mesh generation.

4. Results

The software package developed has been tested and applied to the mesh generation problems for computational domains surrounding different 3D objects. For each case, input parameters are varied to check the quality of the resulted meshes. It is obvious that the quality of the 3D structured computational meshes obtained is acceptable. These generated meshes have been used as the input of a 3D computational fluid dynamics software for turbulent compressible atmospheric flows and air quality simulations.

4.1. The graphic user interface of the package

The package is designed with a main interface that allows users to input topographical elevation data of the top (upper) surface of any 3D objects in the form of digitized contour lines. Once the input file has been read, the software will show the contour map in the main window. When the mouse pointer is moved in the map, the real 2D coordinates, in the XY coordinate plain, of the map are shown on the title bar of the window.

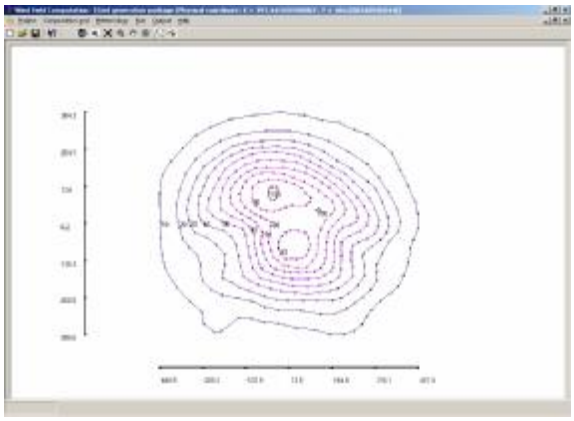

Fig. 5. The contour line view of the top (upper) surface of a real 3D topography.

The software also allows the users to zoom in/out or move the map freely.

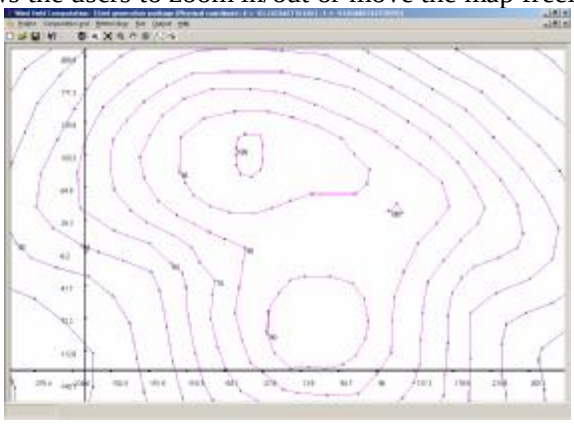

Fig. 6. A zoomed in window of the contour line map.

4.2. Mesh generation for the computational domain surrounding a conic

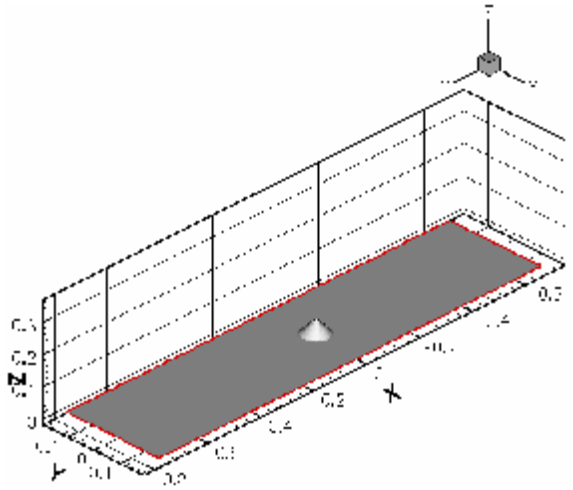

Fig. 7. The 3D computational domain surrounding a conic rising from the bottom plain.

A 3D object is assumed to have a conic shape. A 3D computational mesh will be generated for a specific domain surrounding the conic (Fig. 7). A common requirement for the generated mesh is that the density of the mesh points close to the conic surface has to be greater than that far from the conic.

The figures below show the resulted meshes (in 2D plains) by both methods.

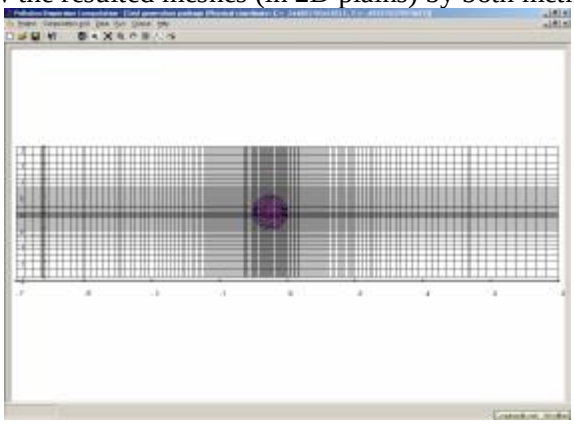

Fig. 8. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain surrounding the conic by the algebraic method.

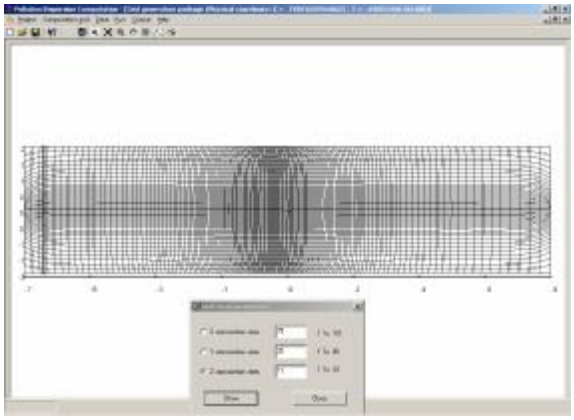

Fig. 9. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain by the elliptic method.

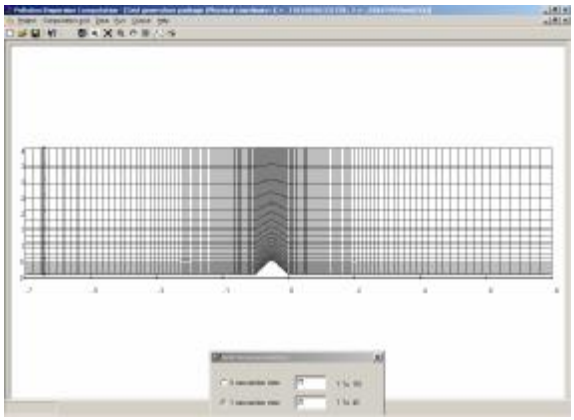

Fig. 10. The generated mesh in the ZX plain of the computational domain by the algebraic method.

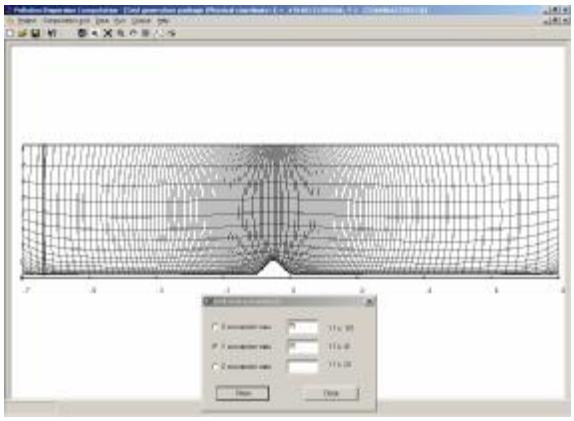

Fig. 11. The generated mesh in the ZX plain by the elliptic method.

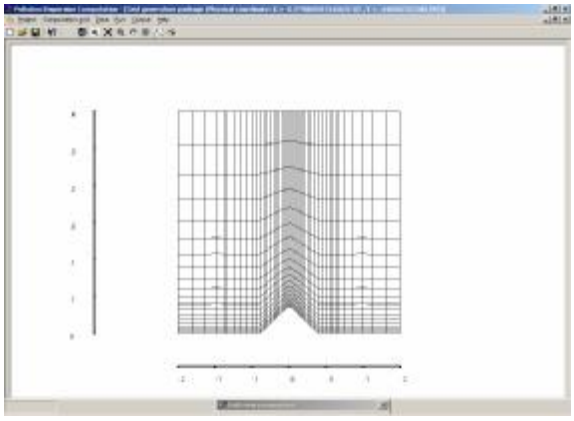

Fig. 12. The generated mesh in the YZ plain of the computational domain by the algebraic method.

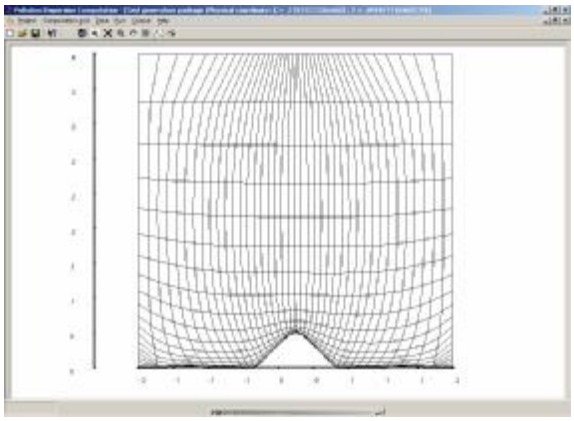

Fig. 13. The generated mesh in the YZ plain by the elliptic method.

4.3. Mesh generation for the computational domain surrounding a 3D sinusoidal object

Another 3D object is assumed to have a sinusoidal shape. A 3D computational mesh will be generated for a specific domain surrounding the object (Fig. 14). The common requirement for the

generated mesh is the same as that of the mesh surrounding the conic (as mentioned in the section above) that the density of the mesh points close to the sinusoidal object has to be greater than that far from the object.

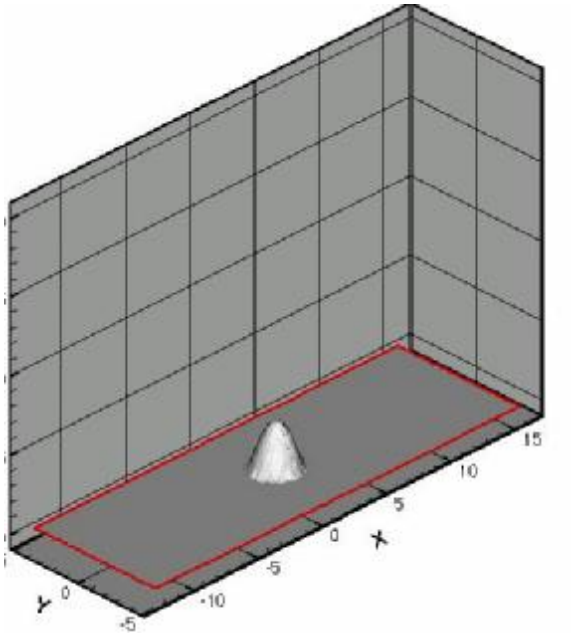

Fig. 14. The 3D computational domain surrounding a sinusoidal object rising from the bottom plain.

The figures below show the resulted meshes (in 2D plains) by both methods.

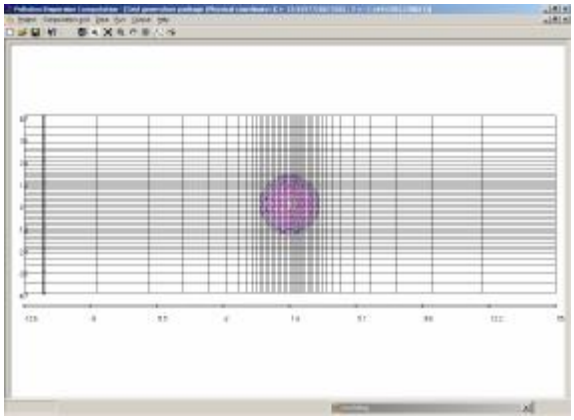

Fig. 15. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain surrounding the sinusoidal object by the algebraic method.

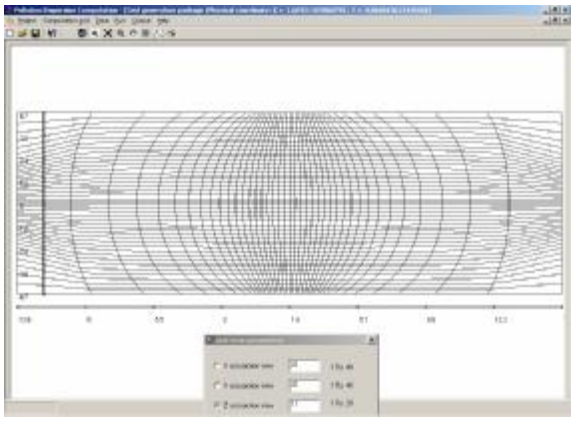

Fig. 16. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain by the elliptic method.

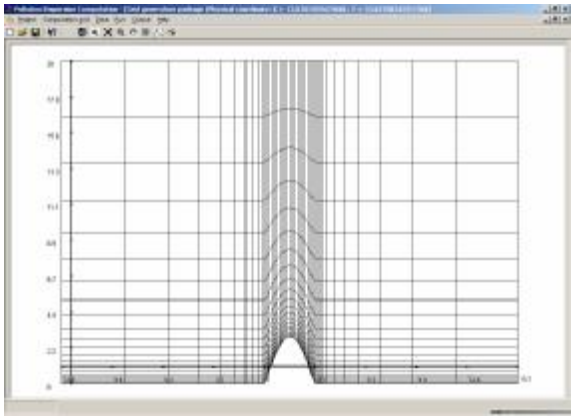

Fig. 17. The generated mesh in the ZX plain of the computational domain by the algebraic method.

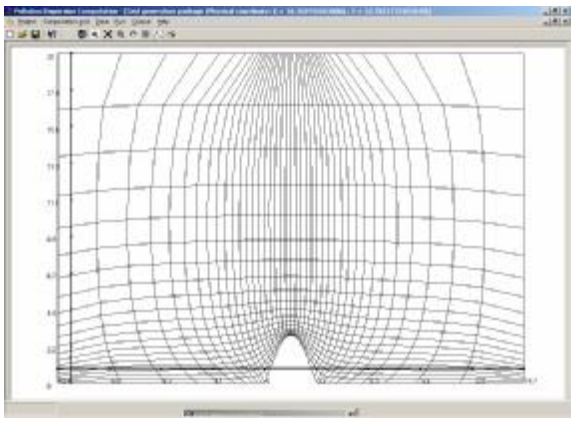

Fig. 18. The generated mesh in the ZX plain by the elliptic method.

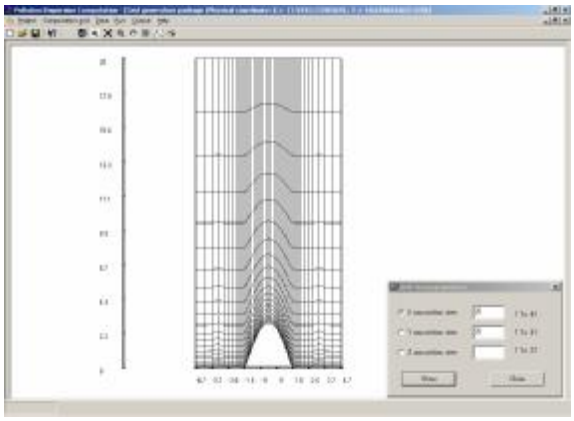

Fig. 19. The generated mesh in the YZ plain of the computational domain by the algebraic method.

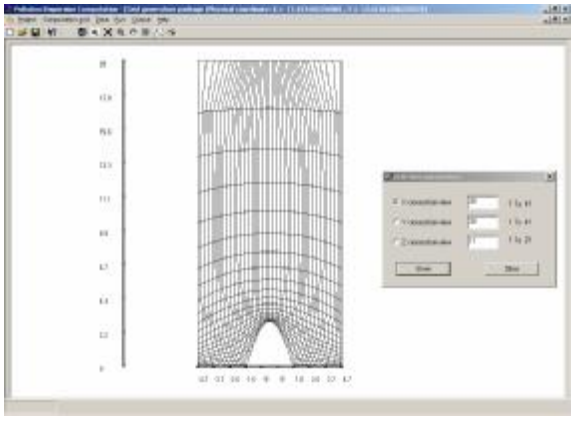

Fig. 20. The generated mesh in the YZ plain by the elliptic method.

4.4. Mesh generation for the computational domain surrounding a real 3D topography

A real 3D topography is considered. A 3D computational mesh will be generated for a specific domain surrounding the topography (Fig. 21). The common requirement for the generated mesh is the same as that of the mesh surrounding the conic and the sinusoidal objects (as mentioned in the sections above) that the density of the mesh points close to the real area has to be greater than that afar.

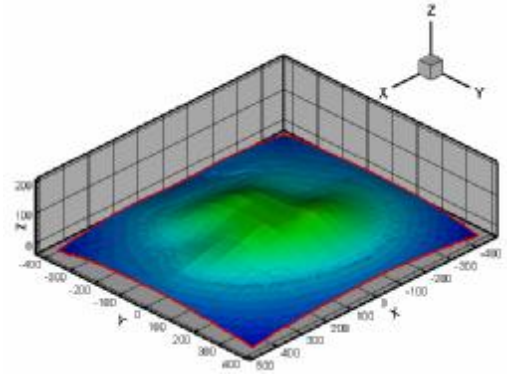

Fig. 21. The 3D computational domain surrounding a real 3D topography.

In this practical case, it should be noted that it is usually desirable that the mesh in the area where the contour lines are close to each other, i.e. big slope, is finer than the mesh in the area where the contour lines are far from each other, i.e. small slope (as shown in the generated meshes below).

The figures below show the resulted meshes (in 2D plains) by both methods.

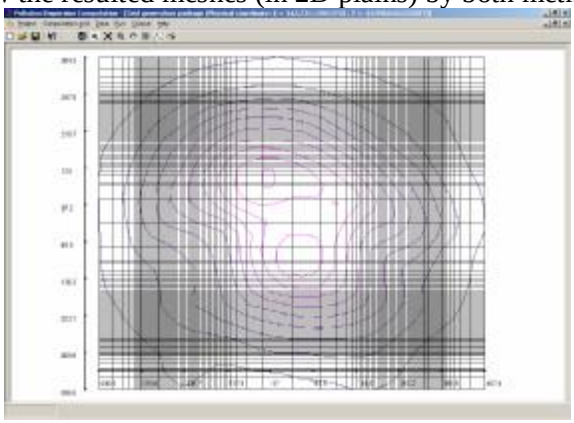

Fig. 22. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain surrounding the topography by the algebraic method.

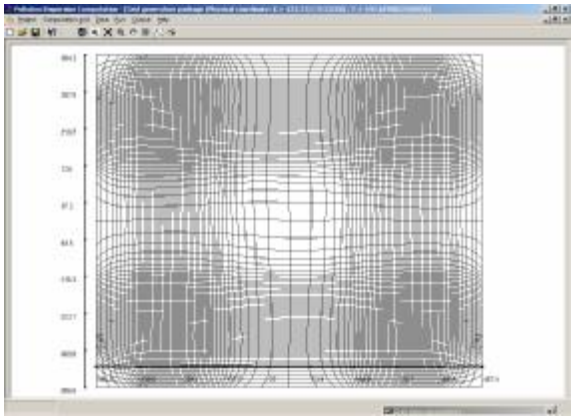

Fig. 23. The generated mesh (projected on the XY coordinate plain) of the bottom surface of the computational domain by the elliptic method.

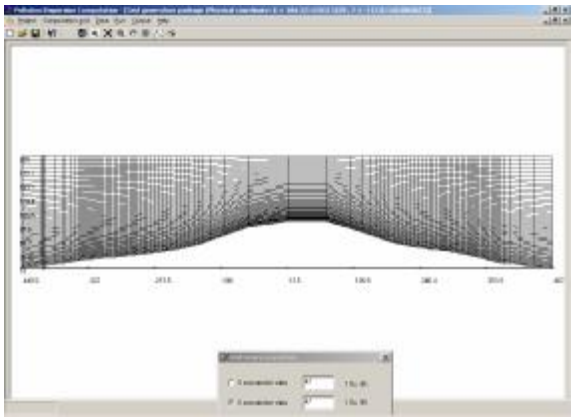

Fig. 24. The generated mesh in the ZX plain of the computational domain by the algebraic method.

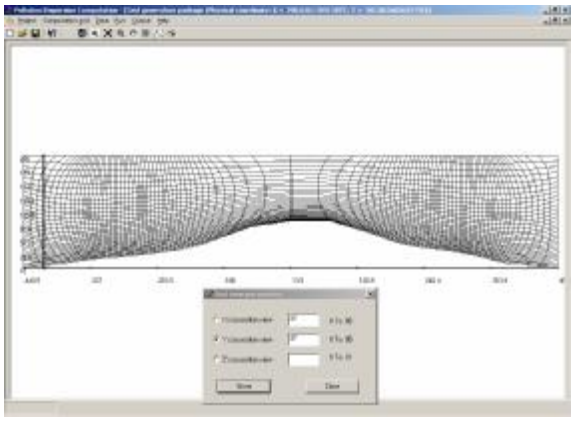

Fig. 25. The generated mesh in the ZX plain by the elliptic method.

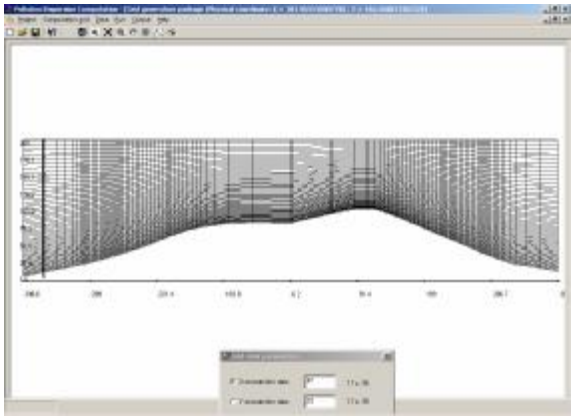

Fig. 26. The generated mesh in the YZ plain of the computational domain by the algebraic method.

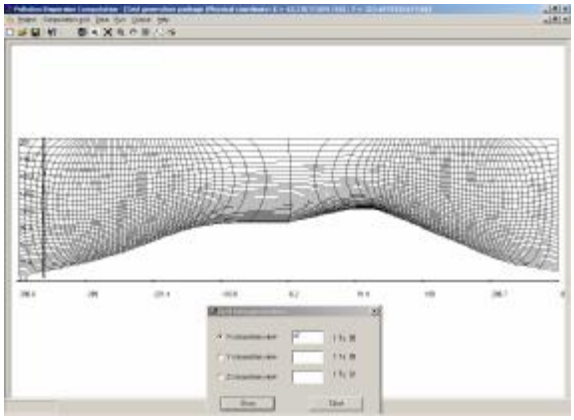

Fig. 27. The generated mesh in the YZ plain by the elliptic method.

4. Conclusions and remarks

The software package developed in this study has been successfully applied to practical computational mesh generation problems. The numerical methods implemented in the software works

well with various sets of input parameters. The generated 3D structured meshes have already been tested and directly used as the input of a CFD code for simulations of 3D turbulent compressible atmospheric flows and/or coupled with pollutant transport/dispersion. The generated mesh quality has been qualified to work well with the CFD code.

The two mesh generation options, which correspond to the algebraic and elliptic methods, implemented in the software package have some features that should be mentioned:

- The 3D algebraic structured mesh generation is more time efficient than the elliptic method. However, with this method, the distortion characteristics in the areas close to the boundary surfaces of the mesh go far into the computational domain. In simulation problems, this feature is not desired. The output meshes need to be checked very carefully to avoid mesh line crossing each other.

- The elliptic method is more time consuming since there is an iterative solution procedure implemented to solve the partial differential equations. It is relatively more complicated than the first one since there must be a pre-generated mesh to be used for the initial condition. The initial mesh in this case is any basic mesh generated by the algebraic mesh generation method.

The graphic user interface is designed and tested to work well. Features coming with this interface are handy and useful in viewing, checking and presenting input data and the output meshes.

The software package has been initially used as a tool for 3D structured mesh generation for simulations of compressible turbulent atmospheric flows and air quality in the Institute of Mechanics.

References

[1] J.F. Thomson, Z.U.A. Warsi, C.W. Mastin, *Numerical Grid Generation: Foundations and Applications*, North-Holland, Amsterdam, 1985, 483.

[2] Patrick Knupp, Stanly Steinberg, *Fundamentals of GRID GENERATION*, CRC Press, Inc. (1994).

[3] Nguyen The Duc, Duong Ngoc Hai, Nguyen Van Thang, Nguyen Tat Thang, Development of a software for the simulation of atmospheric flows over complex topography, *Proceedings of the 2005 Annual National Conference on Fluid Mechanics*, (2005) 55 (in Vietnamese).

[4] Duong Ngoc Hai (project manager), *Development of a software for the simulation of atmospheric flows over complex topography*, A Research Project of the Institute of Mechanics, (2004-2005) (in Vietnamese).

[5] C.A.J. Fletcher, *Computational techniques for fluid dynamics 1. Specific techniques for different flow categories*, Springer-Verlag (1991).

[6] C.A.J. Fletcher, *Computational techniques for fluid dynamics 2. Specific techniques for different flow categories*, Springer-Verlag (1991).

[7] Karstein Sorli, *Generation of structured and adaptive grids by solving elliptic partial differential equation*, SINTEF, N-7034 Trondheim, Norway; www.sintef.no (1996).

[8] Dzhmova Olga, *Generation of Structured grids by solving elliptic PDEs*, ww10.informatik.uni-erlangen.de.

[9] Joe. D. Hoffman, Numerical Methods for Engineers and Scientists, McGraw-Hill, Inc. (1993).