

# Fast Display Technology of 3D Cloud Image of Large-Scale Finite Element Model Based on Lightweight Processing

Xiao-Hui WANG<sup>1</sup>, Liang CHANG, Xiang-Yan XU, and Xiao-Hua NIE  
*National Key Laboratory of Strength and Structural Integrity, Aircraft Strength  
 Research Institute of China, Xi'an Shaanxi, 710065, China*

**Abstract.** Aiming at the visualization of large-scale finite element mesh models in finite element refined modeling and analysis, the rapid formation and efficient blanking of three-dimensional mesh models, large-scale finite element result data organization and rapid display, and other aspects of the system have overcome the rapid display optimization technology of the super-large-scale finite element mesh model, and realized the rapid display of the mesh model with tens of millions of degrees of freedom. The engineering application verification shows the feasibility and practicability of the method.

**Keywords.** 3D model display, minimum node correlation surface table method, directory index search, contour filling

## 1. Introduction

With the continuous improvement of equipment structure complexity and technical and tactical indicators, the scale of refined models has grown rapidly, and computer computing capabilities have continued to increase. The scale of single-machine finite element analysis is no longer limited, and refined analysis based on graphical visualization has become a conventional analysis method. This poses a huge challenge to the graphical visualization technology closely related to the solution.

At home and abroad, the research on graphic visualization technology of finite element model is mainly divided into system development and technical research, such as *ANSYS* [1], *ABAQUS* [2], *VCollab* [3] and other foreign software, which all have a consistent graphic interactive visualization system, but because its algorithm is not open, the scalability is limited; professional scientific visualization software, such as *VCollab*, *TecPlot* [4], and *EnSight* [5], has rich data interfaces but high learning costs. The technical research mainly focuses on universities and research institutes. For example, Xu Liangyin et al. [6] designed finite element model renderer and filter modules to support large-scale data visualization. Based on the basic requirements and processes of the finite element visualization task, Zou Jing et al. [7] of Tsinghua University combed the key algorithms for the display of millions of unit-level three-dimensional finite element models. Cao Yi et al. [8] proposed an interactive visual analysis engine to meet the needs of high-performance industrial simulation, which solved the bottleneck

---

<sup>1</sup> Corresponding author: Xiao-Hui WANG; Email: wxhd100@163.com

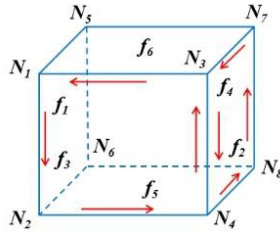
problem of interaction caused by large-scale simulation data. However, through investigation and analysis, it is found that in actual use, the visualization system that comes with the large-scale finite element calculation software has limited functions, poor scalability/openness, and basically does not provide development interfaces. In view of this, this paper breaks through the rapid display technology of large-scale finite element models, rapid rendering of 3D cloud image of finite element calculation results.

## 2. Fast display algorithm of 3d mesh model

**In the three-dimensional model grid, the internal element surface belongs to and only belongs to two elements; the external element surface belongs to only one element [9].** In view of this, this paper adopts the method of minimum node related surface table based on the geometric topological relationship of the finite element mesh, which can quickly obtain the set of element surfaces located on the structure surface.

### 2.1. Element geometry model

As shown in Figure.1, it shows the solid representation model of the three-dimensional unit. In the Figure 1,  $f_1, f_2 \dots f_6$  respectively represent the 6 element faces of the unit, which enclose a unit entity;  $N_1, N_2 \dots N_8$  are the 8 nodes of the unit respectively. The element surface is defined as an ordered node oriented according to the right-hand rule, and the node with the smallest node number in the element surface is regarded as the first node of the ordered node group of the element surface, and the outer normal vector of the element surface is the node The sequence is oriented according to the right-hand rule.



**Figure 1.** Solid model of three-dimensional unit

For example, assuming that  $N_1$  to  $N_8$  are node numbers from small to large, then

$$\begin{cases} f_1 = (N_1, N_5, N_6, N_2) \\ f_2 = (N_3, N_4, N_8, N_7) \\ f_3 = (N_1, N_2, N_4, N_3) \\ f_4 = (N_5, N_7, N_8, N_6) \\ f_5 = (N_2, N_6, N_8, N_4) \\ f_6 = (N_1, N_3, N_7, N_5) \end{cases} \quad (1)$$

### 2.2. Based on the minimum node correlation surface method

The algorithm process is as follows:

Step1: First create a node-related surface table;

Step2: Perform the following processing on each unit:

According to the unit entity model, the unit faces of the unit are formed according to formula (1); each unit face is added to the node-related face table; the overall number of the first node of each face is its The node to which the node belongs in the related surface table. For the unit shown in Figure 1, the relevant surface tables of each node are respectively

$$\begin{aligned} H_{N_1} &= (f_1, f_3, f_6), H_{N_2} = (f_5) \\ H_{N_3} &= (f_2), H_{N_5} = (f_4) \\ H_{N_4} &= H_{N_6} = H_{N_7} = H_{N_8} = \emptyset \end{aligned} \tag{2}$$

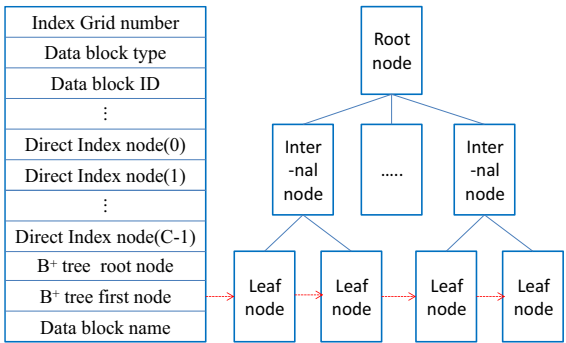
Step3: Eliminate the internal element surface to form the external element surface, that is, the structural boundary surface.

Compare the relevant surfaces in the non-empty  $H_i$ , and eliminate the unit surfaces with the same overall numbers of the nodes contained therein, and the rest is the set of boundary surfaces of the structure.

**3. Result Data Organization Method based on the combination of Hash table and B+ tree**

The calculation result uses a binary data block file format belong to a compression format. They are widely used due to their small data footprint, fast reading and writing speed, and high efficiency.

Store the data block content in the external memory, and only store the data block directory index table in the memory. First, a global Hash table is used to retrieve the data block name, and a separate B+ tree is created for each file to save the index node number of the data block in the file. The index node structure is shown in Figure 2. Second, use the index node number of the data block to replace the data block name in the data block directory file.



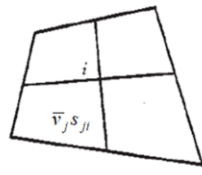
**Figure 2.** Index node structure

**4. 3D Cloud Map Drawing Method**

Based on the characteristics of continuous changes in physical field values and discrete changes in color parameters, this paper unifies the drawing of bar cloud graphs and smooth cloud graphs, and uses a unified algorithm to achieve the drawing of two different types of color cloud graphs.

#### 4.1. Generate the value of the node field

To generate a cloud image, the field value at the cell node must be known, so the cell value of the field value must be converted into the corresponding node value. Among them, the area weighted average method is simple in algorithm and has sufficient interpolation accuracy. Taking the equivalent strain field as an example, as shown in Figure 3, node  $i$  is surrounded by  $m$  adjacent elements, and the equivalent strain value at node  $i$  is



$$\text{and } \bar{v}_i = \frac{\sum_{j=1}^m \bar{v}_j s_{ji}}{\sum_{j=1}^m s_{ji}}$$

**Figure 3.** Equivalent strain interpolation method

Where  $\bar{v}_j$  is the new equivalent strain value around node  $i$ , and  $s_{ji}$  is the area contribution of the  $j$ -th element to node  $i$ .

#### 4.2. Determination of the number of isolines in a unit

In order to make full use of the continuous distribution of the physical field, first calculate the number of isolines required to divide the element into small areas with the same color parameters based on the numerical value of the physical field of the element nodes, and then calculate the physical equivalent of each isoline field value. If the color corresponding to the physical quantity on the entire area is taken as the entire visible light and arranged in the order from red to purple, then  $L=256 \times 5=1280$  colors can be defined and the calculation formula for the number of equivalent lines of physical quantities in any triangle unit is:

$$N^e = \frac{(V_{\max}^e - V_{\min}^e) \times L}{V_{\max} - V_{\min}} \quad (3)$$

In the formula (3), the number of contours in the triangular unit, and, respectively represent the maximum and minimum values of the physical quantities in the unit and the entire solution domain. In any unit, the value of the physical quantities of the contours arranged from low to high is:

$$V_i^e = V_{\min}^e + \frac{(V_{\max}^e - V_{\min}^e)}{N^e} \times i, \quad i = 1, 2, \dots, N^e \quad (4)$$

#### 4.3. Filling between contours in triangular elements

According to the value of the physical field in the area, the color parameter to be filled is calculated, and then filling the area with this color is the main content of drawing a cloud image. Arrange the physical field values of the 3 nodes in a triangular element from large to small, and the order of arrangement is. In this way, when the area is filled between the contours in the triangle unit, according to the different physical field value

relationship between the contours and the triangle nodes, the filling area between the contours is different, which can be expressed as shown in the Figure 4 shows several types of situations.

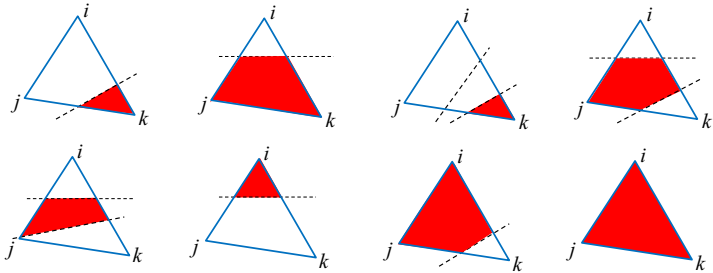


Figure 4. Scan line method of quadrilateral

According to this, the compilation software can complete the filling of different colors in the triangular unit to obtain the required cloud image.

5. Feasibility Verification

In order to verify the actual performance of the technology, software development and integration were carried out based on the self-developed CAE software pre-processing software named *SABRE.visual*[10].

The test results of loading the finite element mesh model are shown in Figure 5 below. It can be seen that the optimized *SABRE.visual* can handle millions of scale models, and the efficiency is significantly improved compared with that before optimization. Millions of models can be loaded in 20s, and it is comparable to Patran in terms of model display.

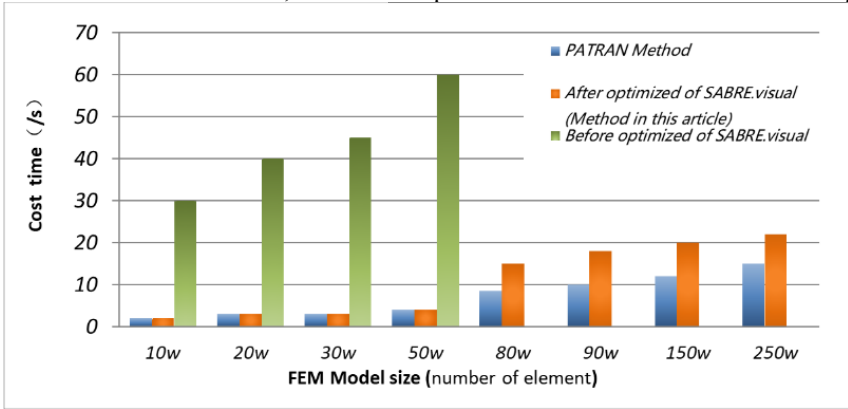
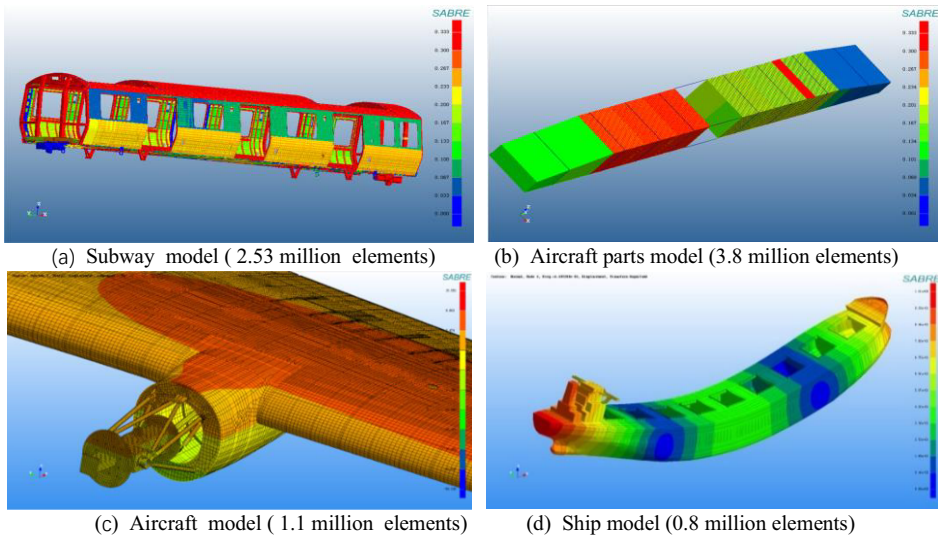


Figure 5. Performance comparison between PATRAN and *SABRE.visual* before and after optimized

Among them, this method achieves very good results in refined model analysis. Such as full-vehicle model (subway model (2.53 million elements, as shown in Figure 6(a)) and aircraft parts model (3.8 million elements, as shown in Figure 6(b)), this method realizes finite element within one minute quick model display and result data visualization.



**Figure 6.** Visualization of million-level finite element model

## 6. Conclusion

Aiming at the problem of visualization of large-scale finite element models in refined analysis, based on the research status and existing problems at home and abroad, the system overcomes the rapid display optimization algorithm of large-scale finite element models, and implements software in the *SABRE.Visual* system. The engineering application verification shows the feasibility and practicability of the method.

## References

- [1] Thompson M K, Thompson J M. ANSYS mechanical APDL for finite element analysis[M]. Oxford: Butter-worth-Heinemann, 2017.
- [2] Ciesielska-Wrobel I. Finite element modeling of textiles in Abaqus™ CAE[M]. Boca Raton: CRC Press, 2019.
- [3] Jing T. Aras Partners with VCollab on Simulation Pro-cess Technology[J]. 2019.
- [4] Bellevue W. Tecplot user's manual[J]. Amtec Engineering Inc, 2003.
- [5] LLNL. VisIt user's manual[OL]. [2021-07-06]. [https:// visit-sphinx-github-usermanual. readthedocs.io/en/ develop/](https://visit-sphinx-github-usermanual.readthedocs.io/en/develop/).
- [6] Xu Liangyin, Li Yunpeng, Chen Paosong. Design and implementation of a general visualization system SIPEsc.POST for very large-scale finite element calculation [J], Chinese Journal of Computational Mechanics, 2015,32 ( 5 ) : 220-224.
- [7] Zou Jing, Ji Hongguang. Fast visualization algorithm for super-large three-dimensional finite element model[J]. Journal of Graphics. 2012, 33(2): 13-19.
- [8] Cao, Wang, Xia et al. Interactive visual analysis engine for high performance industrial simulation [ J ].Journal of Computer Aided Design and Graphics, 2021, 33 (12) : 1803-1810.
- [9] Lin Genghao, Ma Tianbao, Ning Jianguo. Large-scale grid generation and display technology in three-dimensional finite difference calculations[J]. Computer Aided Engineering, 2012, 21(4): 1-5.
- [10] Wang Binwen, Duan Shihui, Nie Xiaohua and so on. Development status and future challenges of CAE software for aeronautical structure analysis [ J ].Aeronautical Journal, 2022 ( 06 ) : 28-51.