8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

MESH DOMAIN DECOMPOSITION SUPERVISED BY CONTROL SPACE

*** Tomasz Jurczyk[1] and Barbara Głut[1]**

[1] AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
{jurczyk,glut}@agh.edu.pl

Key Words: *Geometry Decomposition, Tetrahedral Mesh Generation, Load Balancing.*

ABSTRACT

The discretization of computational domain is a problem commonly encountered in computational mechanics. In cases where the computations are performed using meshes with number of elements surpassing 10^7, the sequential approach to the mesh generation task may become problematic not only due to the large computational cost, but also with respect to the memory requirements. As a consequence, in recent years much work has been devoted to the problem of parallel mesh generation [1-5].

In the presentation there will be described two methods of decomposition of the discretization domain into a number of subdomains, which can be then discretized separately. The first method is based on a geometrical decomposition of the domain, based on the surface mesh only [6,7]. The surface mesh is partitioned into subdomains by cutting it with separators. Next step is the closing of the subdomains by generation of proper surface mesh on the separators. Then, the volume mesh can be created independently for each closed subdomain. The second method utilizes the coarse volume mesh to decompose the domain. The coarse mesh is created by triangulation of boundary nodes. In this approach the separators are only virtual, their task is to identify where the mesh should be refined. The decomposition of the subdomains is performed by separating the coarse mesh along the refined areas, which also creates the boundaries for split subdomains. Then, as in the first method, all closed subdomains can be meshed independently. In the latter method the sequential part of algorithm increases, but more information about the domain can be gathered. Utilization of this additional information helps to properly establish the splitting surfaces.

The proposed procedure (in both approaches) has a number of advantages. The communication between computational nodes is limited. The volume mesh is not stored in the memory of a single computational node. The sequential mesh generator can be utilized without any modifications. Also, this technique allows to preserve the original surface mesh of the domain.

However, in both methods the essential problem is the appropriate positioning of the separator (both the real and virtual one). The selection of a separator (with respect to its shape and placement) significantly influences the final decomposition of the domain as well as the performance of the subsequent steps of the decomposition procedure. The selection of separator should ensure:

- minimum *cut size* – measured with the number of elements on the separator closure,
- *load balancing* – basically balancing the time required for mesh construction in each subdomain,
- minimum number of adjacent interfaces – the separators should be selected in subsequent steps in a way, which minimizes the number of subdomains having a common contact point,
- quality of volume elements – the placement of separator may significantly influence (in some methods) the quality of created volumetric elements (if the cutting angle between the separator and the surface mesh is too small, bad quality elements may be created).

In the presentation the Authors intend to concentrate on the load balancing problem. At the beginning of the discretization process only the geometric description of the model is known. With this information only, it is hard to properly assess the time required to create meshes in subdomains resulting from the partitioning. Such assessment may be based on the number of volume elements created in each of the subdomains, which in turn can be approximated during the selection of separator basing on the number of surface mesh elements. However, such measure is rarely sufficient. It may fail to properly recognize the shape of subdomains, mesh anisotropy or the method used to generate the mesh. The generator developed by Authors constructs the meshes using the data gathered in a control space structure. The essential information, stored in this control space, is the metric supervising the shape and size of elements. In the presented work the information stored in control space is used in order to obtain better prediction of the number of volume elements, which will be created in the given subdomains.

REFERENCES

[1] H. de Cougny and M. Shephard, "Parallel refinement and coarsening of tetrahedral meshes" *Int. J. Numer. Meth. Engng* Vol. **46**, 1101-1125, 1999.

[2] N.P. Chrisochoides. "A survey of parallel mesh generation methods", *http://www.andrew.cmu.edu/user/sowen/pmesh_survey.pdf*, 2005.

[3] E. Ivanov, H. Andrä and A.N.Kudryavtsev. "Domain decomposition approach for automatic parallel generation of 3D unstructured grids", *Proc. of ECCOMAS CFD 2006 European Conference on Computational Fluid Dynamics*, Egmond aan Zee, The Netherlands, 2006.

[4] T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, D.R. O'Hallaron, "From Mesh Generation to Scientific Visualization: An End-to-End Approach to Parallel Supercomputing", *Proc. of SC'06*, Tampa, FL, 2006.

[5] Y. Ito, A.M. Shih, A.K. Erukala, B.K. Soni, A. Chernikov, N.P. Chrisochoides, and K. Nakahashi. "Parallel unstructured mesh generation by an advancing front method", *Mathematics and Computers in Simulation*, Vol. **75**, 200–209, 2007.

[6] T. Jurczyk, B. Głut and P. Breitkopf. "Parallel 3D Mesh Generation using Geometry Decomposition". *AIP Conf. Proc. Volume 908 NUMIFORM 2007, Materials Processing and Design: Modeling, Simulation and Applications*, 1579–1584, June 17-21, Porto, Portugal, 2007.

[7] B. Głut, T. Jurczyk, P. Breitkopf, A. Rassineux and P. Villon. "Geometry Decomposition Strategies for Parallel 3D Mesh Generation". *Proc. CMS'05 Conference on Computer Methods and Systems*, Vol. **1**, 443–450, Kraków, 14-16 November 2005.