An environment for 3D geometric modeling applied to finite element simulations

Luiz Fernando Martha

Department of Civil Engineering and
Tecgraf – Computer Graphics Technology Group
Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rua Marquês de São Vicente, 225, Gávea, Cep 22453-900,
Rio de Janeiro – RJ, Brasil
lfm@tecgraf.puc-rio.br

This presentation is a summary of software development for engineering and geological applications, including some educational tools, that has taken place in Tecgraf/PUC-Rio in the last 18 years. It presents a computational environment for geometric modeling applied to finite difference and finite element simulations. This environment is based on the integration of computational techniques and methodologies developed in Tecgraf/PUC-Rio.

The main objective of this modeling environment is to provide interactive-graphics tools for the simulation of many engineering and geology processes. These tools rely not only on efficient graphics interface and visualization, but mostly on powerful underlying data representation schemes, robust geometric modeling, automatic mesh generation, and extensible management of simulation attributes.

The target applications include analysis of floating units and offshore structures, well stability analysis, structural modeling of subsurface geological units, simulation of fracture mechanics processes, simulation of geological restoration processes, simulation of geological sedimentation processes, and engineering educational tools.

One of the key aspects in many of these applications is the need for a multi-region heterogeneous underlying data representation. In addition, the degree of user interaction for model manipulation and visualization aimed poses severe performance requirements on the data representation scheme of the simulation software. To achieve this, the implemented modeling capability relies on powerful, topology-based, data structures that present the following characteristics:

- The data structures must provide a natural navigation across all phases of a simulation: pre-processing (model creation), numerical analysis, and post-processing (model results visualization).
- The data structures must take into account that the simulation may induce, at least temporarily during model creation, geometric objects (curves and surfaces) that are inconsistent with the target final model. This requires a non-manifold topology representation capability.
- The data structure should aid in key aspects of geometric modeling, such as surface intersection and automatic region recognition, as well as in surface and solid finite element mesh generation in arbitrary domains.
- The data structure must provide for efficient geometric operators, including automatic intersection detection and processing.

- The data structure must efficiently provide for any sort of adjacency relationship between model entities. That is, adjacency information such us retrieval of which geometric entities are adjacent to a given reference entity must be answered rapidly by the data structure query functions. This will help in several pre- and post-processing algorithms.

In addition, target applications require efficient and robust finite element mesh generation capability. Modeling methodology adopts the combination of both structured and unstructured mesh generation schemes. Structured mesh generation relies heavily on two aspects of the modeling methodology: efficient and friendly graphics user interface, and flexible domain decomposition capability, which depends on powerful underlying data representation and on automatic region recognition.

Unstructured mesh generation adopts an adaptive strategy that is based on a hierarchical refinement in which curves and surfaces (including internal interfaces) are discretized prior to the solid regions. Target sizes of finite elements obtained from numerical error analysis and from geometric restrictions are stored in a global background structure, an octree recursive spatial decomposition data structure. Based on this background structure, the model curves are refined using a binary-partition algorithm. The discretization of curves is used as input for the refinement of adjacent surfaces. Surface discretization also employs the background octree-based refinement coupled to an advancing front technique for the generation of an unstructured triangulation. Surface meshes are used as input for the refinement of adjacent volumetric domains. Volume discretization combines the background octree refinement with an advancing-front technique to generate an unstructured mesh of tetrahedral elements. In all stages of the adaptive strategy, curve and surface refinement takes into account curvature characteristics.

Finally, generic management of simulation attributes is another important aspect of this modeling environment. In one hand, target applications may share underlying data structures, geometric modeling capabilities, and mesh generation schemes. On the other hand, these applications differ significantly in the types of simulation attributes that they handle. Moreover, simulation attributes in general are not known a priori by the developer of the simulation software. These attributes are related to the specific needs of the target application and, usually, corresponds to the most important professional skills of the end user. The challenge is to build simulation software that provides for generic geometric modeling and mesh generation capabilities and, at the same time, allows for extensible and user-managed attribute handling. This is achieved using software engineering technology such as interpreted-embedded languages and object-oriented programming.