A Point Creation Strategy for Mesh Generation Using Crystal Lattices as Templates

Ulisses T. Mello [1], Paulo Roma Cavalcanti[2]

[1]IBM T.J. Watson Research Center, Yorktown Heights, NY, U.S.A. (ulisses@watson.ibm.com)
[2] Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil (roma@cos.ufrj.br)

ABSTRACT

A strategy is presented to define background meshes based on crystal lattices. Crystal-based background meshes have much better initial distribution of minimum dihedral angles than regular background meshes, facilitating the optimization procedures that rely on previously defined tetrahedral meshes. In addition, these background meshes can reduce significantly, or even eliminate, the number of slivers generated during a Delaunay triangulation.

Keywords: FE mesh generation, background mesh, crystal lattices, triangulation, tetrahedralization, computational geometry.

1 INTRODUCTION

Discretization of geometric models representing domains of interest is an essential part of numerical solution of Partial Differential Equations. Delaunay triangulation algorithms are frequently used to generate unstructured meshes for Finite Element Analysis. Although Delaunay triangulation has a well-defined method to create the connectivity of a set of points, it does not provide a method for generating points within the domain. The creation of these points is in many cases a necessity to improve the quality of tetrahedral and triangular meshes. There are several methods described in the literature for field point creation. A comprehensive review can be found in [1]. One of the most common methods proposed in the literature is the use of auxiliary background meshes to define the position of the field points to be inserted within the domain. Using this approach, nodes from a regular grid, or an octree-based grid, are inserted within the domain during the triangulation. Distinct background grids can be used to control the density and distribution of points in the domain. In this paper, we suggest an alternative approach to define background meshes. Our background meshes are based on crystal lattices normally found in rock minerals. We show that crystal-based background meshes have much better initial distribution of minimum dihedral angles than regular grids, facilitating the optimization procedures that rely on previously defined tetrahedral meshes.

In addition, crystal-based background meshes can reduce significantly, or even eliminate, the number of slivers generated during a Delaunay triangulation. Crystal lattices of some minerals such as quartz (tectosilicate), or diamond (cubic-F) are very attractive to be used as background meshes because they form perfect tetrahedral lattices in the nature. In this paper, we investigate whether the quality of tetrahedral meshes generated by a Delaunay triangulation can be improved by using these crystal lattice templates for background meshes.

Despite the fact that crystal lattices are formed by the interaction of atomic forces, rather than a geometric criterion, the use of predefined background meshes is considerably faster and easier to implement than using physically based methods to define optimal posi-

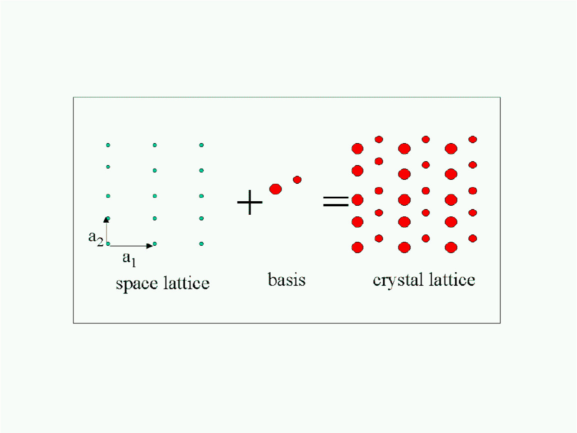

Figure 1: The crystal lattice is formed by the replication of the basis to every lattice point. The combination of a rectangular point lattice with a two-atom basis resulted in a two-dimensional hexagonal crystal lattice.

tion of field points. This approach seems natural when one considers that the dual of Delaunay triangulations, Voronoi tesselations, have been used extensively in crystallography to simulate crystal growth (e.g., [2])

2 CRYSTAL LATTICES AS TEMPLATES

Rock crystals consist of regular arrangement of atoms. In a crystal there must be a long-range order such that the pattern of the atoms repeats regularly throughout the crystal. Crystal packing depends on the type and size of the atoms, and also on the magnitude of interatomic forces. Many atoms, particularly metallic, can be considered as regular spheres, which pack together very closely. The structure of crystals can be described in terms of a lattice, with a group of atoms attached to every lattice point, as can be seen in Figure 1. The group of atoms is called basis, which when repeated in space forms the crystal structure. The point lattice is defined by three fundamental translation vectors (a_1, a_2, a_3). The set of points in the lattice is defined by a translation operator, T:

$$T = u_1 a_1 + u_2 a_2 + u_3 a_3, \qquad (1)$$

where, u_1, u_2, and u_3 are arbitrary integers [3]. Thus, a particular crystal lattice is defined by the choice of the basis and values for translation vectors. In this study, we make the analogy between crystal lattices and unstructured meshes, where the atoms in the lattice correspond to mesh nodes. In Figure 2, we show an example demonstrating how this concept can be used for the creation of field points in the mesh generation process. In this example, we use a two-dimensional lattice for the sake of simplicity, and in the next section we apply these concepts to three-dimensional cases.

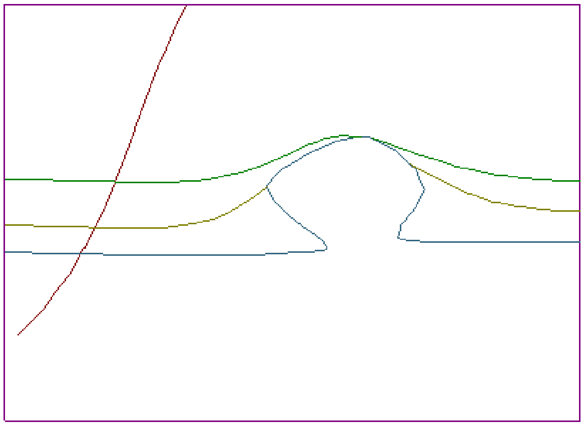

Figure 2: Cross-section of a geological model with complex geometry.

In Figure 2, a cross-section of a geological model is shown with 4 sub-regions. The resulting mesh after inserting field points using a regular grid as background mesh is shown in Figure 3. The meshing was done by using a constrained Delaunay triangulation, as described in [4]. Not all the points in the background mesh were used, because only points farther than a minimum specified distance from the lines defining the regions boundaries are inserted. This point classification is discussed in detail in section 3.2. Note that in Figure 3 most of the triangles in the mesh are rectangular isosceles, and thus they possess one $90°$ and two $45°$ internal angles. The predominance of these angles is evident in the bimodal distribution shown in the diagram of angle distribution for the mesh (Figure 5). On the other hand, we used an hexagonal crystal lattice as a template for a background mesh to generate the mesh displayed in Figure 4. Because the hexagonal lattice has only equilateral triangles the distribution of the angles is essentially unimodal, with an initial quality much better than the one generated using the regular grid as background.

Once an initial triangulated mesh is created, smoothing algorithms can be used to improve the quality of the mesh by moving the unconstrained nodes of the mesh.

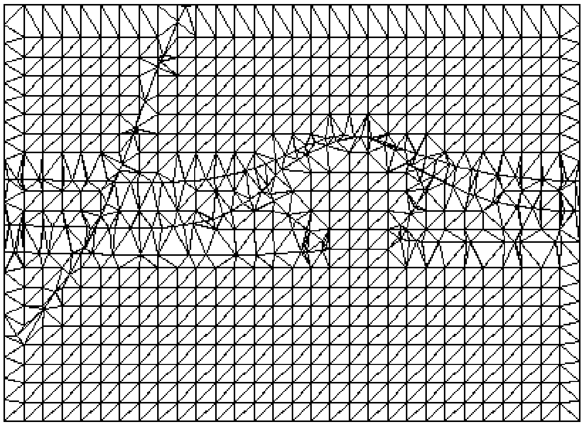

Figure 3: Triangular mesh using a regular grid as background mesh to create points within the regions of the model displayed in Figure 2. Note that most triangles are rectangular isosceles.

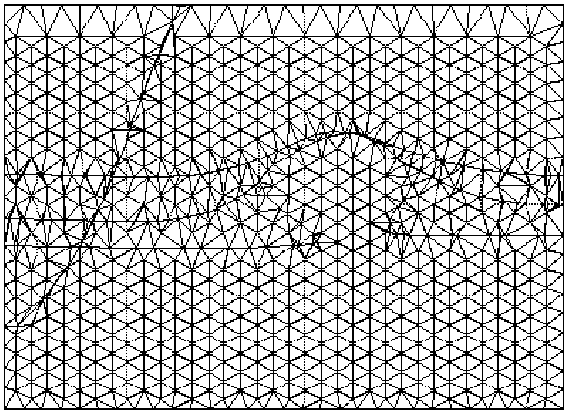

Figure 4: Triangular mesh using a hexagonal lattice as background mesh. Note that most triangles are equilateral.

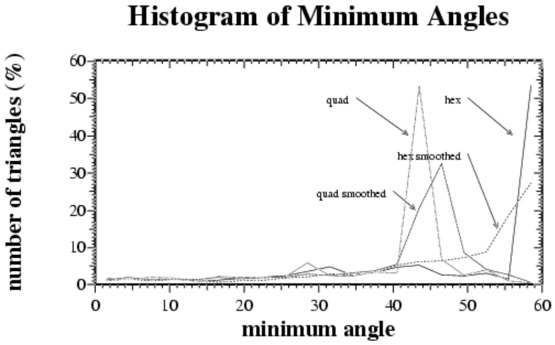

Figure 5: Angle distribution diagram for Figures 3 and 4. Note that the hexagonal lattice has a much better distribution of internal angles than the rectangular lattice.

In Figures 6 and 7, we show the resulting meshes corresponding to Figure 3 and Figure 4 after a Laplacian smoothing operator is applied. The respective diagrams of the angle distribution of these meshes are also shown in Figure 5. The effect of the initial distribution of the points is clear in these diagrams: the mesh that had a bimodal angle distribution is much improved after the smoothing. However, its quality is still in general inferior than the initial mesh that used the hexagonal background mesh. When we apply the smoothing operator to the mesh that used the hexagonal lattice, we obtain a very good quality mesh as shown in Figure 5.

3 THREE-DIMENSIONAL LATTICES

In this section we show the results of performing unconstrained Delaunay triangulations using the main types of background meshes based on Bravais lattice points [5]: cubic, hexagonal, tetragonal, and orthorhombic. The main characteristics of these lattices can be seen in Figure 8, which shows their basic unit cell. These cells are defined by the angle between their axes and the magnitude of each basis vector. Note that the cubic lattice corresponds to the regularly gridded background meshes normally described in the literature.

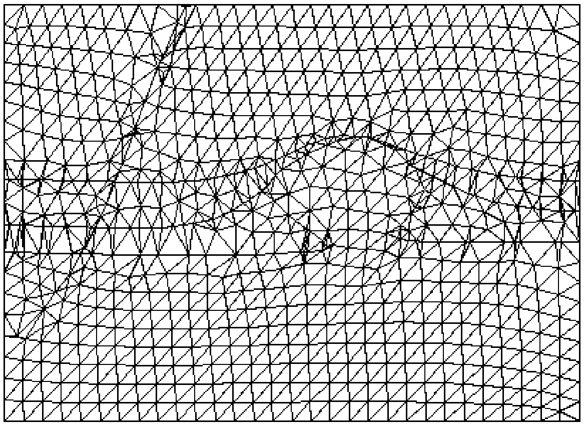

Figure 6: Triangular mesh corresponding to Figure 3 after the application of the Laplacian smoothing operator.

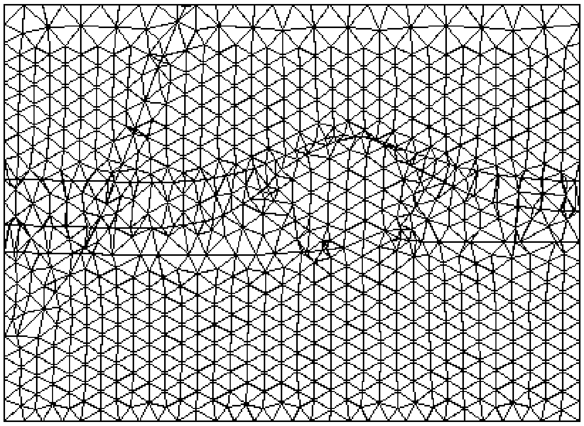

Figure 7: Triangular mesh corresponding to Figure 4 after the application of the Laplacian smoothing operator. Note that the quality of the triangles are superior than the ones shown in Figure 6.

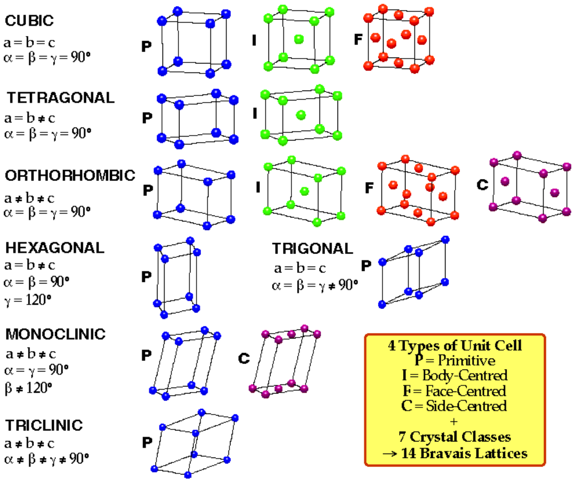

Figure 8: 14 Bravais lattices [6].

3.1 Point creation in 3D

Since there are various possible crystal structures, they have been divided into groups according to the configuration of unit cells or atomic arrangements. One of the most widely used classification scheme is the Bravais lattices (Figure 8). This scheme is based on the unit cell geometry of a parallelepiped, regardless of the atomic positions in the cells. In this scheme, a coordinate system is defined with its origin at one of the unit cell corners and its axes coincide with one of the three parallelepiped edges, which extend from this corner, as shown in Figure 8. The unit cell geometry is completely defined in terms of six parameters: the three edge lengths, and the three inter-axial angles.

It has been found that crystals can be classified based on seven possible combinations of these parameters. These seven crystal lattice systems are:

- cubic,
- tetragonal,
- hexagonal,
- orthorhombic,
- rhombohedral,
- monoclinic, and
- triclinic.

The lattice parameter relationships and unit cell for each system is illustrated in Figure 8. The general lattice is triclinic and all the remaining 13 are special cases of it. Note that some systems are subdivided into four packing types:

- primitive (P),
- body-centered (I),
- face-centered (F), and
- side-centered (C).

These lattices have remarkable symmetry properties that have been discussed in detail by Lovett [7].

In this study, we used five lattices as templates for background meshes:

- cubic-P (halite-NaCl),
- cubic-F (diamond),
- orthorhombic-P,
- tetragonal-P, and
- hexagonal-P (quartz).

In Figure 9, we show the minimum dihedral angle distribution obtained by performing an unconstrained Delaunay triangulation to the points generated at the positions corresponding to each type of lattice. Note, in this figure, that the histograms have a common interval for the cubic, orthorhombic, and tetragonal lattices.

The crystal structure of the diamond and quartz are especially interesting as templates for background meshes because in the nature they have perfect tetrahedral bound arrangements, where all dihedral angles are equal to $70.53°$. The space lattice for a diamond is cubic-F. The basis has two identical carbon atoms at $(0,0,0)$; $(1/4,1/4,1/4)$ associated with each point of the cubic-F lattice. The tetrahedral bounding characteristic is such that each atom has 4 nearest neighbors and 12 next nearest neighbors. Despite the fact that the Delaunay triangulation is purely based on geometric criteria, it still generates perfect tetrahedra during the triangulation of the diamond lattice, although only in a small percentage (about 10%). The hexagonal lattice performs better than the diamond, where 20% of all generated tetrahedra are perfect.

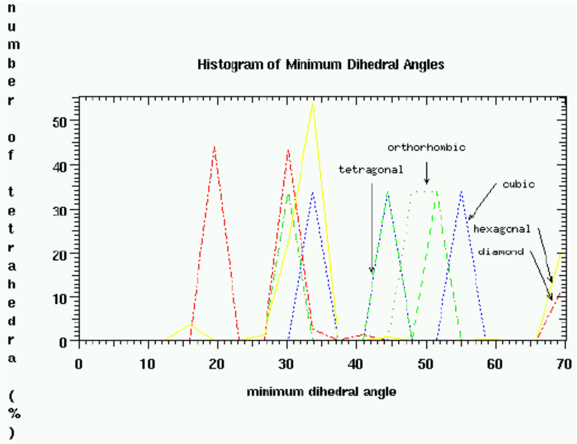

Figure 9: Diagrams of dihedral angles for: Cubic-P (blue), Orthorhombic-P (brown), Tetragonal-P (green), Cubic-F (red), Hexagonal-P (yellow) lattices.

The selection of the arbitrary integers in equation (1) has an important effect on the occurrence of slivers. The cubic lattice is much more likely to present problems because of the presence of more than 4 co-spherical nodes. By changing the spacing of the nodes from cubic (rectangular grid) $(u_1 = u_2 = u_3)$ to tetragonal or orthorhombic $(u_1 \neq u_2 \neq u_3)$ lattice greatly reduces, or even eliminates, the occurrence of co-spherical nodes.

In Figure 10, the triangulation of a model using the cubic lattice is shown. The highlighted tetrahedra are slivers, totalling 42. By just using an hexagonal lattice on the faces bounding the model, the number of slivers is reduced to 13. When an hexagonal lattice is used for both the faces and the interior volume of the model no sliver is produced at all.

3.2 Point Classification

The point classification step is necessary to determine to which region of a model a node in the background mesh belongs. Generally, the points of the background mesh are generated within the bounding box of a model. A typical model may contain a set of regions, which are 3D enclosed volumes, bounded by a set of faces. Only points classified as being into a region, or possessing a neighborhood of a predefined radius that does not intersect any boundary face, should be used in the triangulation.

To check whether a given point is into a region, it suf-

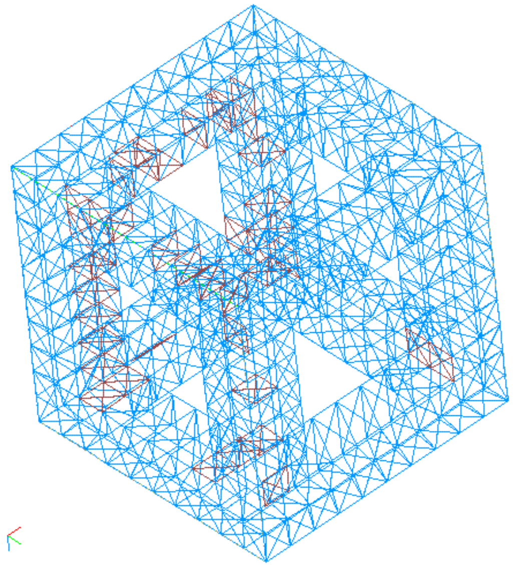

Figure 10: Triangulation of a model using the cubic lattice. 42 slivers are generated and they are displayed in dark gray.

fices to count the number of intersections that a ray, starting at the point, generates on the region boundary. If this number is even the point is outside. Otherwise, the number of intersections is odd and the point is inside [8]. An alternative method, based on spherical polygons, can be found in [9]. Although straightforward, both methods are very slow in 3D, since the number of nodal points may be quite large. The performance is further compromised if the boundary is defined by a large number of 3D polygonal faces, or if there are a large number of regions in the model.

We use an alternative method for speeding up the classification process. Rays starting outside of the bounding box are shot towards the box and we use the adjacency information of the 3D model to classify each sample generated along a ray. This method has also been used to create regular and rectilinear meshes for arbitrarily shaped geological models [10]. In this case, rays are shot and sampled at specific intervals, and thus generating rectilinear meshes along a ray path and reducing the classification to a one-dimensional problem. This process is analogous to the scan-line algorithm used in computer graphics for visible surface determination [11]. The intersection points between a ray and the boundary faces of the model are sorted along the ray path. Then, by using adjacency information, the region that contains all samples between two consecutive intersections is determined. For efficiency, we have used a search data structure, R*tree [12, 13, 14], to locate quickly the intersecting faces along the shooting ray.

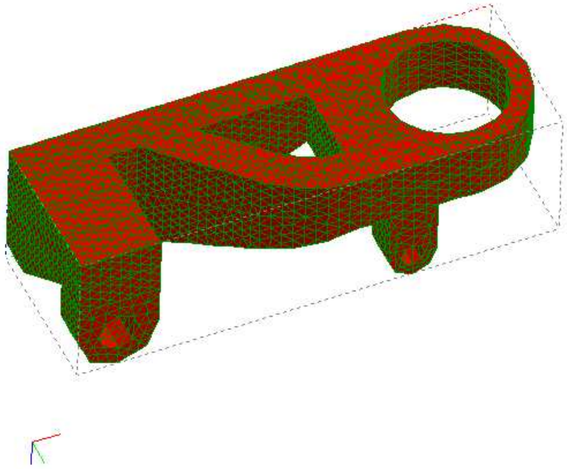

Figure 11: Triangulation of a mechanical model using an hexagonal lattice.

4 THREE-DIMENSIONAL EXAMPLE

In Figure 11, we show the triangulation of a typical mechanical part, and in Figure 12, the dihedral angle distribution using an hexagonal and a cubic lattice. As expected, the minimum dihedral angle distribution using the cubic lattice has three peaks centered at $35.3°$, $45.0°$, and $54.7°$, corresponding to the minimal internal angles of the triangulation of a cube with 6 tetrahedra. Using the hexagonal lattice, these peaks are shifted to $30°, 38°$, and $70°$, respectively. After using the smart smoothing described by Freitag [15] on both lattices, the smoothed versions do not show any change in the position of the peaks. They olny have their magnitude reduced and the distribution slightly spread, like in a diffusion process. However, the hexagonal lattice produced the most smooth distribution of all of them. Also, the hexagonal lattice increased significantly (by 20%) the number of perfect tetrahedra.

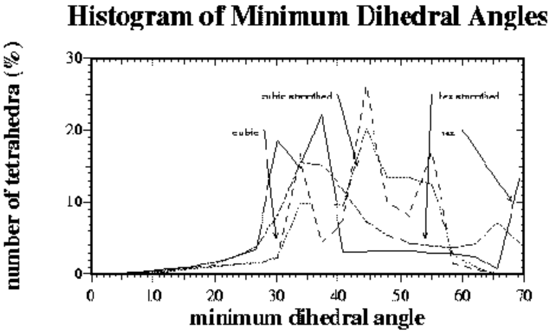

Figure 12: Angle distribution diagram for the model shown in Figure 11.

5 CONCLUSIONS

Among the 14 Bravais lattices, we have investigated the use of five of them as background mesh templates for the creation of field points in the mesh generation process. In 3D, the hexagonal lattice produces the best results. It generates perfect tetrahedra, and in general avoids the generation of slivers. Our results indicate that cubic lattices should be avoided due the potential of generating a large number of slivers. This is specially true, if one uses cubic lattices in the interior of the model and on the surfaces delimiting the regions of the model. In 2D, hexagonal lattices are particularly superior than rectangular lattices. Our results suggest that when hexagonal lattices are used more than 80% of the triangles generated have dihedral angles larger than 50°.

Acknowledgement

We thank Tom Jackman for revising this manuscript and providing a set of routines to create crystal lattices.

References

[1] P-L. George and H. Borouchaki. *Delaunay Triangulation and Meshing: Application to Finite Elements*. Editions Hermes, Paris, 1998.

[2] B. F. Schaudt and R. L. Drysdale. Multiplicative weighted crystal growth voronoi diagrams. In *Procedings of the 7th Annu. ACM Symposium on Computional Geometry*, pages 214–223, 1991.

[3] C. Kittel. *Introduction to solid state physics*. John Wiley and Sons, seventh edition, 1996.

[4] P. Roma Cavalcanti and Ulisses T. Mello. Three-dimensional constrained Delaunay triangulation: a minimalist approach. In *Proceedings of the 8th International Meshing Roundtable*, pages 119–129, Lake Tahoe, CA, October 1999. Sandia National Laboratories.

[5] J.D. Dana, R.V. Gaines, H.C.W. Skinner, E.G. Foord, B. Mason, and A. Rosenzweig. *Dana's New Mineralogy Manual*. John Wiley and Sons, eigth edition, 1997.

[6] P. C. Gasson. *Geometry of spatial forms: Analysis, synthesis, concept formulation and space vision for CAD*. John Wiley and Sons, 1983.

[7] D. R. Lovett. *Tensor properties of crystals*. Adam Hilger, Philadelphia, 1989.

[8] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, New York, 1994.

[9] Paulo Cezar Pinto Carvalho and Paulo Roma Cavalcanti. Point in polyhedron testing using spherical polygons. In Alan Paeth, editor, *Graphics Gems V*, pages 42–49. Academic Press, Boston, 1995.

[10] Ulisses T. Mello and Paulo Roma Cavalcanti. A topologically-based framework for simulating complex geological processes. In *press, AAPG Hedberg Conference Special Volume*, 2000.

[11] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics, Principles and Practice*. Addison-Wesley, second edition, 1992.

[12] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 47–57, Boston, June 1984.

[13] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-tree: a dynamic index for multi-dimensional objects. Technical report, College Park, University of Maryland, February 1987.

[14] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger. The r-tree: an efficient and robust access method for points and rectangles. In *Proceedings of the SIGMOD Conference*, pages 322–331, Atlantic City, June 1990.

[15] Lori Freitag. On combining Laplacian and optimization-based mesh smoothing techniques. *Trends in Unstructured Mesh Generation, ASME Applied Mechanical Division*, 220:37–44, 1997.