SIMPLE AND FAST MESHING IN THE ELECTROMAGNETIC FIELD SOLVER GDFIDL

W. Bruns, R. Merte, Technische Universität, 10587 Berlin

Abstract

The paper presents some possibilities and applications with the mesh-generator of GdfidL. GdfidL uses pyramidal cells to discretise the material fillings. The wanted geometry can be built from primitives such as cylinders and bodies of revolution, or the geometry can be described by a set of CAD-files. The primitives 'cylinder' and 'body of revolution' are allowed to be extremely general, i.e. the axis of the 'cylinder' and the 'body of revolution' may point in arbitrary direction. The footprint of the cylinders and the r(z) function of the 'body of revolution' may be arbitrary closed polygons. The meshing of geometries which are specified via CAD files is reasonably fast.

1 SIMPLE AND FAST MESHING DECREASES THE TOTAL ERROR IN THE COMPUTED FIELD

GdfidL computes fields, which are approximations of electromagnetic fields. We normally distinguish two kinds of error: The error due to the discretisation of the differential equation in homogeneous regions, and the error due to the approximation of the material filling. Previous papers already dealt with the generalised diagonal fillings used in GdfidL to minimise the error due to the approximation of the material filling.

There is another source of error: The user. The user is a human being, as such, he is lazy. If the field-solver only offers a clumsy way to define the geometry that the user is interested in, the user is likely to simplify the geometry such, that the definition is easier. But then the field solver can no longer compute the real fields, since the real geometry with all its details is not known. In order to reduce the error that stems from the user, we have to give the user convenient tools to describe his geometry.

GdfidL has only a limited set of geometric primitives from which the geometry under investigation must be constructed. The geometric primitives available are: rectangular brick, circular cylinder, general cylinder, and body of revolution.

Now one might say: *What? No sphere? No cone? No elliptical waveguide? No (whatever)?* To which one may respond: *A sphere and a cone are bodies of revolution, an elliptical waveguide is a cylinder.*

With these primitives it is possible to generate fairly complex structures quite easy. Most of the geometric primitives may be specified in a very general form. For example, it is possible to discretise a circular cylinder. But the circular cylinder is not restricted to have its axis in one of the cartesian directions, the axis may point in any direction.

In addition to these primitives, it is possible to define a geometry with the aid of so called 'STL'-files. This is a dataformat that can be produced by many CAD-systems. An STL-dataset describes a closed volume via a list of triangles.

2 SPECIFICATION OF ARBITRARY AXES

The geometric primitives 'circular cylinder', 'general cylinder' and 'body of revolution' all have axes. All these primitives may have their axes pointing in any direction. In order to build the mesh, the mesher has to decide for many points whether the point lies within or without the body. The decision for a geometric primitive with an axis in a general direction can be reduced to the decision for a primitive with an axis in z-direction by pre-multiplying the actual point with a matrix which is defined by the wanted axis. As an example of an application Fig. 1 shows the meshing of a 90^o step twist.

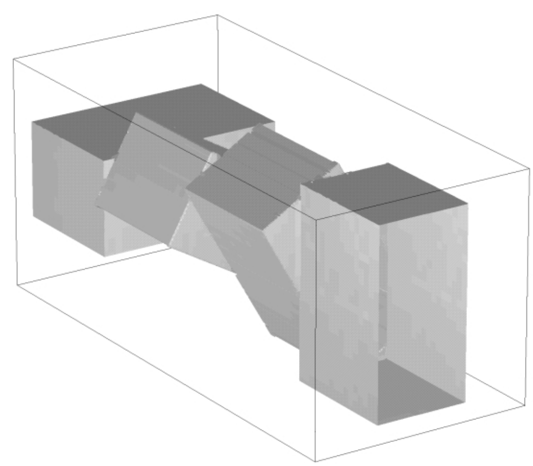

Figure 1: The discretisation of a 90^o step twist. The input-file for this geometry has only 15 lines.

3 A VERY GENERAL CYLINDER

What GdfidL calls a cylinder is a very general one. A GdfidL-cylinder is defined by its footprint, its axis, and an algorithm, how the footprint shall extrude the volume of the cylinder. For example, it is easily possible to mesh a smooth 90^o twist as a single cylinder, by saying that the footprint of the cylinder shall be rotated while it extrudes the volume. A resulting mesh is shown in Fig. 2.

It is also easily possible to specify that the footprint of the cylinder shall grow linearly while it extrudes the volume. Figure 3 shows the mesh of a 'Muffin-Tin' cavity, where at the edges of the cavity a chamfer has been applied.

The decision whether a point lies inside such a general cylinder can be reduced to the decision whether a point lies within a conventional cylinder, by pre-multiplying the coordinates of the point with a matrix, which depends on the wanted axis and also on the coordinates of the point itself.

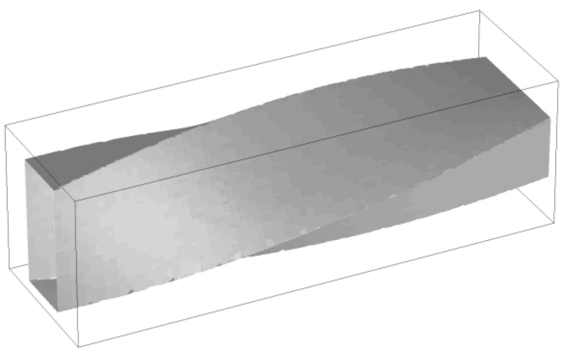

Figure 2: The discretisation of a smooth 90^o twist. The inputfile for this geometry has only 17 lines.

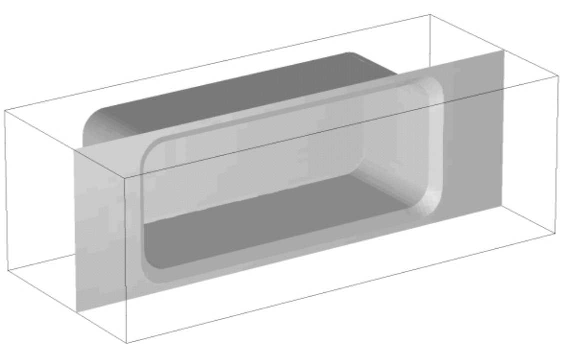

Figure 3: The discretisation of a muffin tin cavity where at the edges of the iris a chamfer has been applied. The geometry is described as two cylinders with the same specification of the footprint. One cylinder is a conventional cylinder, the other is a cylinder with linear growth. The inputfile for this geometry consists of 32 lines.

4 MESHING AN STL-DATASET

An STL-dataset describes a closed body via a list of triangular patches. When meshing such a geometry in a grid with N-cells, at minimum N decisions must be made whether a gridpoint lies inside or outside the body. The decision whether a point lies inside a closed surface can be made by counting the number of patches that would be punctured if one would shoot from the point to infinity. When the data-set contains M triangles, a naive implementation would require MxN decisions, whether a test-ray crosses a triangle. By sorting the triangles in, say, 100 groups, the number of decisions to compute can be reduced roughly by a factor of 100. This sorting in groups has been implemented in GdfidL.

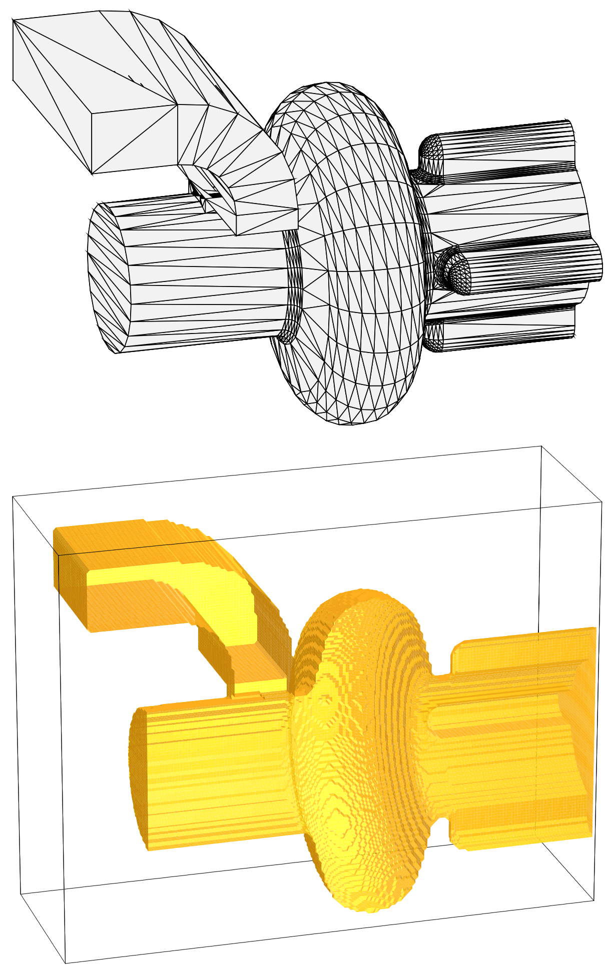

Figure 4: A superconducting cavity of Cornell University, described as a STL-dataset. Above: The 3300 defining faces. Below: The resulting mesh of the cavity. There are 1.7 million cells in the grid. The meshing takes 1 minute on a 1997 vintage PC.

ACKNOWLEDGMENT

We wish to thank Zipi Greenwald of Cornell University for the STL-dataset.

REFERENCES

[1] W. Bruns, "GdfidL with Generalized Diagonal Fillings and Periodic Boundary Conditions", Proceedings of the X. International Symposium on Theoretical Electrical Engineering, September 6-9 1999, Magdeburg, Germany