6th Workshop on Numerical Methods in Applied
Science and Engineering (NMASE 07)
Vall de Núria, 9 a 11 de enero de 2007
©LaCàN, www.lacan-upc.es

AN INTERACTIVE MESH GENERATION ENVIRONMENT FOR GEOMETRY-BASED SIMULATIONS. PROGRESS UPDATE.

X. Roca and J. Sarrate

Laboratori de Càlcul Numèric (LaCàN),
Departament de Matemàtica Aplicada III,
Universitat Politècnica de Catalunya,
Jordi Girona 1-3, E–08034 Barcelona, Spain
e-mail: {xevi.roca,jose.sarrate}@upc.edu,
web: www-lacan.upc.edu

Key words: Finite element method; mesh generation; geometric modeling; non-manifold geometry; graphical environment; object–oriented design.

Abstract. *We presented an overview of an interactive mesh generation environment in our previous work [1] that it is being developed in the Laboratori de Càlcul Numèric (LaCàN). We started the development of the software in order to unify available legacy code, new developments and research algorithms in only one mesh generation package. This paper presents an update to the last environment overview after significant implementation and conceptual development. We provide a brief summary of: new features, as submapping and data output suitable for geometry-based methods as [2]; current developments, as GUI improvement and 3D modelling tools; and future features, such as the command interpreter. Finally, we conclude that chosen mesh generation paradigm [3] and software engineering concepts [4] have allowed us to improve and scale the environment since last revision.*

1 Introduction

Nowadays mesh generation software allows to discretize complicated domains with several mesh types including hexahedra, tetrahedra, triangles and quadrilaterals. These discretizations are demanded by computational methods in a wide range of practical situations going from the most simple and standard academic example to the the most complex and challenging assembly model. Although mesh generation advances during last decades, the process of obtaining a mesh is still time consuming and not fully automatic. This means that in practical applications meshes are obtained using both advanced graphical user interfaces (GUI) and simple and specific hand made procedures. In particular, GUI mesh generation environments have shown their power allowing to model, repair, visualize and discretize, more and more complex models with less time and less user interaction.

Hence software industry and research groups are investing a lot of effort in developing their own GUI mesh generation environments.

As we mentioned in our previous work [1] the Laboratori de Càlcul Numèric (LaCàN) is developing a mesh generation environment with a GUI. Previous to this development, the mesh generation algorithms developed by our group consists in a sort of scripts and tools that allow to obtain meshes for certain types of elements and analysis. With this environment LaCàN facilitates to its members the generation of meshes by means of an easy to use and unique toolkit. In particular the mesh environment unifies legacy, new and research code developed by our group. Moreover, the development team is constrained to provide a mesh environment that is customized for the special discretization necessities of LaCàN, such as high order element meshes or NEFEM geometry based analysis [2].

In this paper we present the progress update since last revision of the mesh generation environment [1]. Specifically, we present these new developments classified by groups of related features. First, in section 2, we detail new geometrical modeling features. Once the model is designed we may need to assign boundary conditions or material attributes. These features are presented in section 3. Third, when the model is well defined we use mesh generation tools in order to obtain a suitable discretization. Section 4 describes the new functionalities related to meshing algorithms. In section 5 we present the graphical user interface improvements. In section 6 we summarize the current state and we list the future developments. Finally we present the concluding remarks.

2 Geometrical modeling

One important goal of the mesh environment is to provide an easy to use modeler. The modeler has to allow to design very simple and academic examples and at the same time more complex models. Thus, in this section we review the new and powerful modeling features that have been include: 3D primitives, extrusion, boolean operations and single and array transformations. They allow to model both simple and complex geometries.

2.1 Three dimensional primitives

Previous version of the environment only provides 2D primitives tools. Nowadays, users can work with three new 3D primitives, see figure 1:

- **Box**. We provide two corner points for determining a box.
- **Sphere**. This primitive is defined by a center and a radius.
- **Cylinder**. This creates a cylinder with circular base and parallel to the z-axis. The center, the radius and the height are the parameters that user can choose.

2.2 Extrusion

CAD programs provide several modeling tools. Most of them are specializations of generic concepts that allow to model complex geometries. One of the most powerful

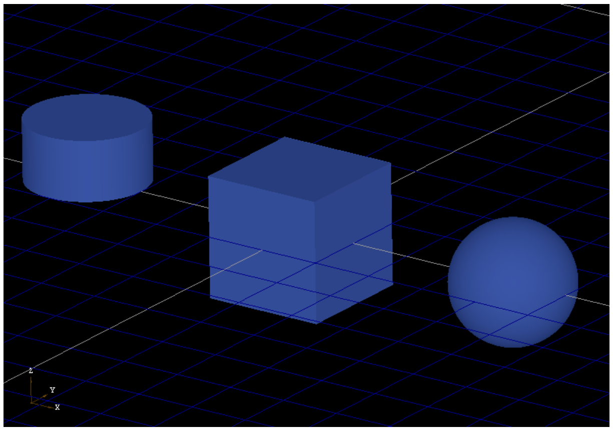


Figure 1: Result of the new 3D primitives: box, sphere and cylinder.

generic model tools is the extrusion. With the extrusion user can sweep a surface along a curved path in order to obtain a closed volume where the cap surfaces are copies of the initial one, see figure 2. Moreover, with this tool we can define extrusion volumes that afterwards can be meshed with hexahedra by means of our sweep generation algorithm, [5, 6, 7].

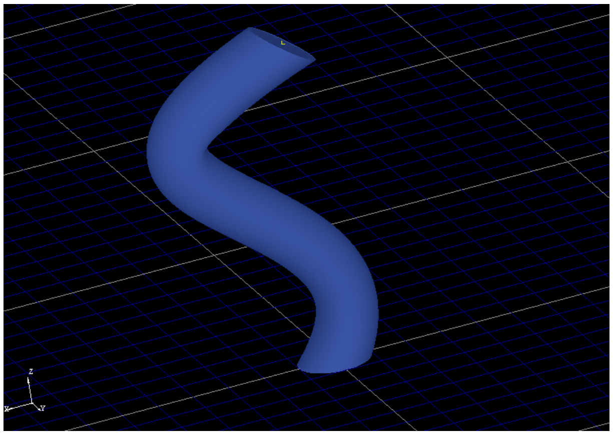
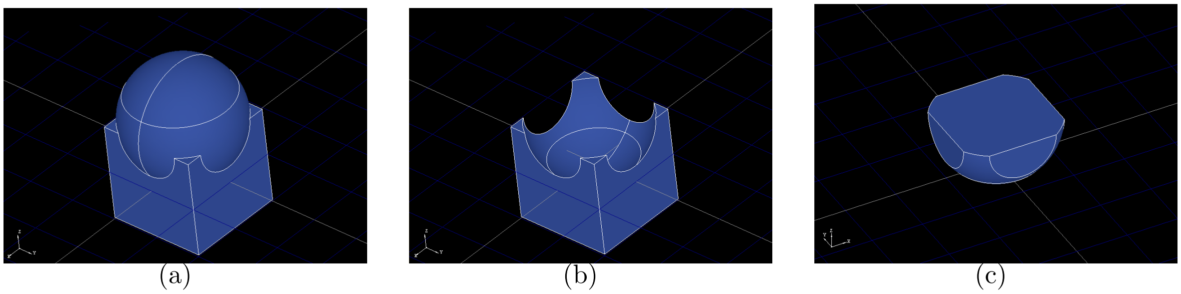


Figure 2: Extrusion geometry obtained by sweeping a surface along a curve.

2.3 Boolean operations

Existent modeling tools use several constructions paradigms. On of the most popular is the *Constructive solid geometry* (CSG) paradigm, because a modeler can use a set of simple objects, *primitives*, to model very complex geometries. The user obtain the complex model by using Boolean operations:

- **Union**. Two solids are merged into one solid, see figure 3(a).



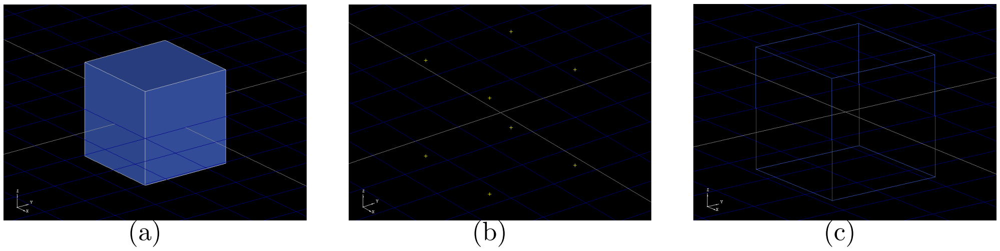
(a) (b) (c)

Figure 3: Obtained geometries with boolean operations: (a) The merge of two solids. (b) The substraction of one solid from the other one. (c) The region common to both solids.

- **Difference**. Tool asks for two solids. The second solid is subtracted from the first one, see figure 3(b).
- **Intersection**. The procedure provides as result the common region of two solids, see figure 3(c)

2.4 Subcomponents access

An interesting feature of a CAD program is to provide access to the composing entities of the assembly model. To this end we have implemented an explode tool that returns the composing entities of certain dimension. In particular, we can get access to subentities, of selected objects, of the types: vertex, edge, wire, face, shell, solid and compound solid. For instance, figure 4(a) shows a cube that can be decomposed with the explode tool in its vertices, figure 4(b), or in its edges, figure 4(c).

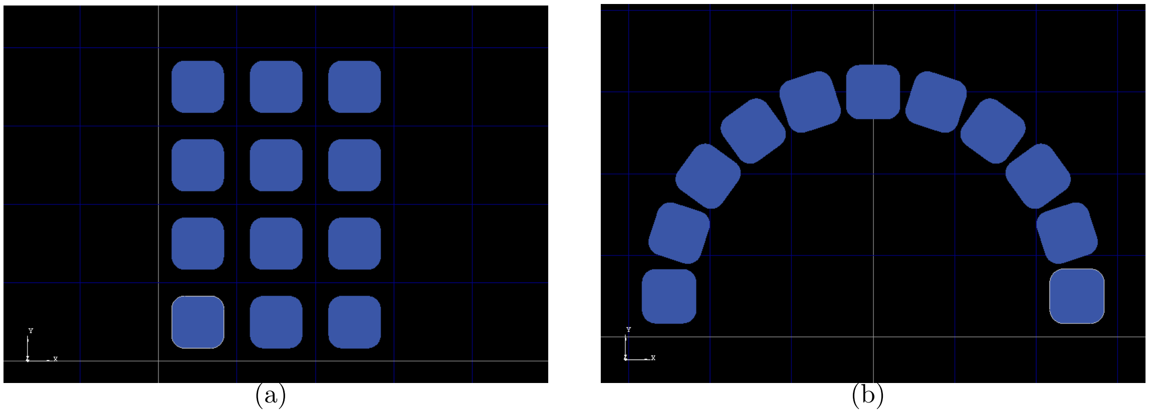


(a) (b) (c)

Figure 4: Explode operation on a cube solid. (a) Original cube. (b) Composing vertices. (c) Composing edges.

2.5 Transformations

In order to have a fully CSG kernel it is important to develop tools for relocating, orienting, resizing, mirroring and copying objects in the model space. Thus, several geometric transformations tools are implemented in the environment. All of these tools



(a) (b)

Figure 5: Practical use of array transformations: (a) Result of a cartesian array transformation. (b) Several copies by means of polar array transformation.

allow to optionally preserve a copy of the original object after the transformation. We can classify them in two groups:

- **Single**. In this group we obtain as a result one deformed copy of each selected object. The environment provides tools for translate, rotate, mirror and scale objects.
- **Array**. In this case we obtain several transformed objects for each original one. The first tool is the *cartesian array* that creates an $m \times n$ matrix of translated objects for each original one, see figure 5(a). The second tool is the *polar array* that creates several rotated copies of an original object according a center and an angle, see figure 5(b).

3 Model attributes

Computer Aided Engineering programs (CAE) has to deal with some properties applied to the geometrical entities that characterize the model to simulate. These properties usually correspond to boundary conditions and material properties of the model. We denote these properties as *model attributes* and we provide an abstract and generic procedure to deal with them. The user can mark with labels the geometrical entities that compose the model in order to properly assign boundary or material conditions. Each label can be applied to several entities of the same dimension. In addition, we can apply different labels to the same entity. Moreover, the user can choose if the labels are applied over the nodes or over the elements of the entities. In the first case, we can mark with the same label the nodes of a group vertices, edges, faces or solids. In the second case, we can mark with the same label: the edges over a group of curves; the faces over a group of surfaces; or the cells over a group of solids.

In some CAE programs, commercial and non-commercial, user needs to remesh the model each time that changes or assigns new conditions. We have developed a data

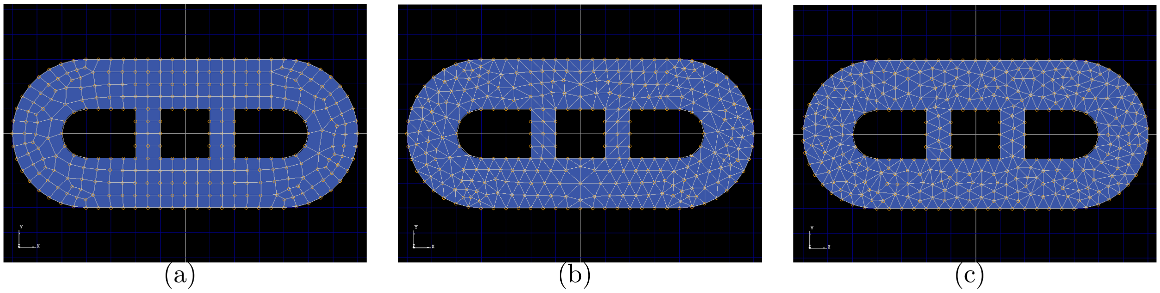


Figure 6: A domain with holes meshed with several unstructured meshers. (a) Gen4u algorithm. (b) Quadrilateral maximum angle splitting. (c) Delaunay mesh with Triangle.

structure that overcomes this drawback. Specifically, mesh entities lie inside `Mesh` objects which are associated to geometrical entities. When a mesh entity asks for its attributes this query is delegated to the container mesh that asks to corresponding geometrical entity. Not storing the value of the attributes in the mesh entities and querying to the corresponding entity allow to avoid the time consuming task of remeshing the model when a given attribute is changed.

4 Mesh generation

In this section we present new added features in the mesh generation context of the environment. Note that several of the new features are not usual in existent commercial and freeware pre-processing tools. However they are of major priority for our research group since they allow to create and to renumber high-order meshes for $p \geq 1$.

4.1 Domains with holes

Some mesh algorithms work only with simply connected domains, i.e. domains without holes. If we want to mesh a domain with holes we need to convert the multiply connected domain into a simply connected one. To this end, it is usual to connect holes to outer loop using roundtrip curves and obtaining a unique simply connected outer loop that enclose the whole domain. In previous versions of the pre-processor user needs to convert, by hand, the domain with holes to a simply connected one. In the current version this process is automatically done and allows to mesh domains with holes with all available unstructured meshers, see figure 6.

4.2 Shells: groups of adjacent faces

Hierarchically based mesh paradigm [3] is adopted in the data structure of the environment. As it is detailed in [1] this allows to obtain conform meshes in groups of adjacent faces, called shells. Specifically, edges are meshed only one time and the obtained mesh is re-used from adjacent faces that share this edge. This works well, without any modification, for meshing shells with triangular elements, but not for quadrilaterals. It is

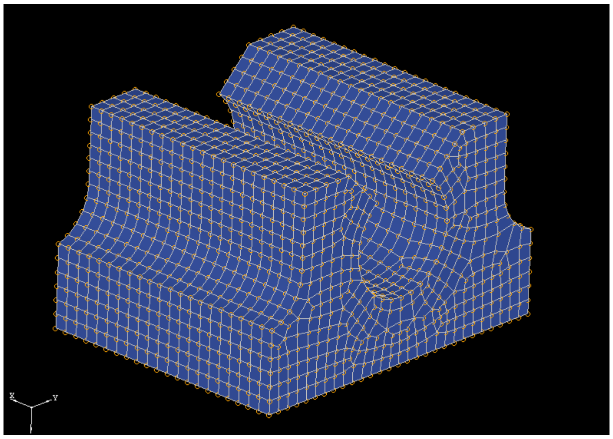


Figure 7: Conform quadrilateral mesh on a closed shell of faces.

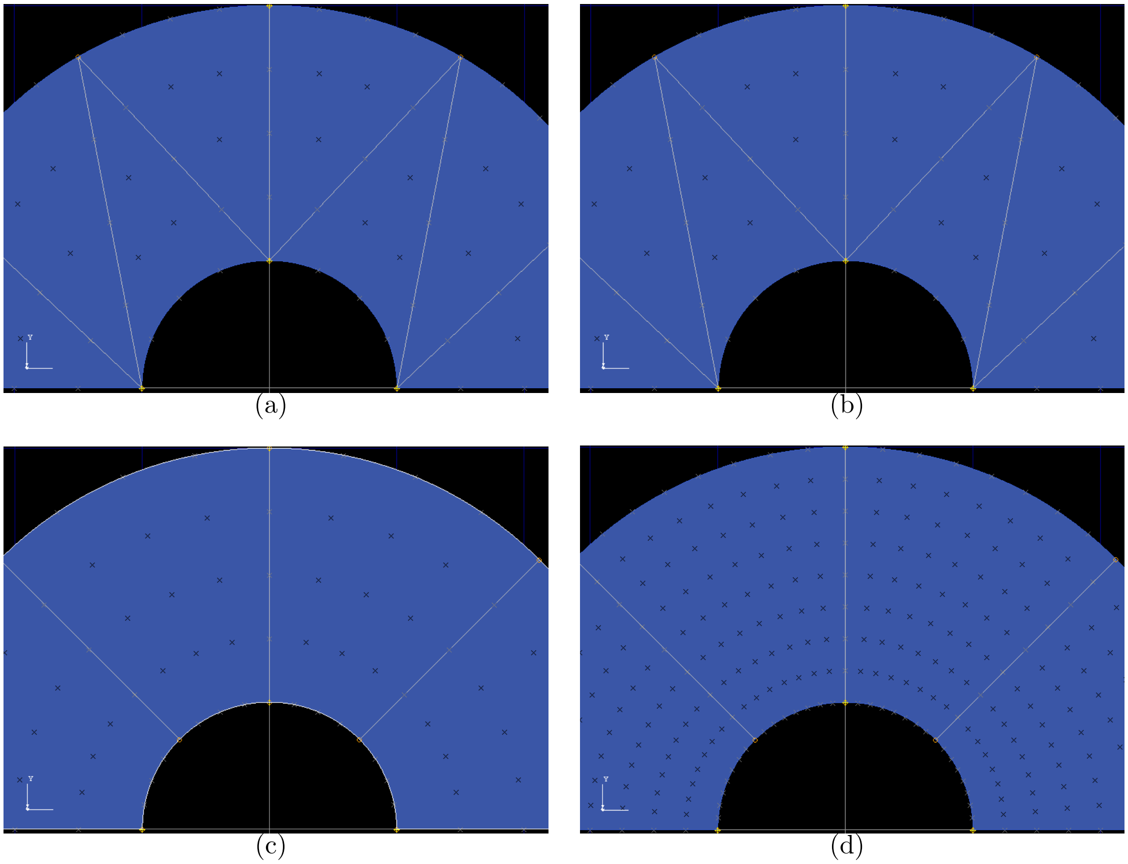
well-known that a domain can be meshed with quadrilaterals if and only if the boundary is meshed with an even number of nodes. One solution is that the user assign by hand the desired number of nodes in each edge subject to that condition. This solution is implemented by several commercial and freeware packages, but requires an error-prone interaction with the user. Our implementation automatically assigns the parity condition prior to mesh groups of adjacent faces with quadrilateral elements, see figure 7.

4.3 Submapping

The last version of the environment did not include a tool for generating structured quadrilateral meshes on complex domains. To this end we have included an implementation of the submapping algorithm. This meshing procedure splits the surface into several patches, logically equivalent to a quadrilateral, and then meshes each patch separately. The submapping algorithm can only be applied to geometries such that each angle between consecutive edges is, approximately, an integer multiple of $\pi/2$. User assigns desired size and the implemented procedure automatically classifies vertices angles, divides each edge with a compatible number of segments and finally generates the structured mesh.

4.4 High-order meshes

One desirable feature of any mesh generation software is the possibility of generating high order p-meshes, where p is the interpolant degree. Some mesh environments are able to export high order meshes for $p = 2$, but not many of them are able to generate meshes for p greater to 2. However several numerical methods, such as NEFEM [2], need high order discretizations. These meshes have to be automatically generated and not by means of hard coded program specific for a particular domain. To cover this gap the environment has a high order export feature, for $p \geq 1$, that generates middle edge nodes over curves of the domain, and inner face nodes following curved edges of the element, see figure 8.



(a) (b) (c) (d)

Figure 8: High order meshes for a curved domain. (a) Triangular elements with $p = 4$. (b) Triangular elements with $p = 8$. (c) Quadrilateral elements with $p = 4$. (d) Quadrilateral elements with $p = 8$.

4.5 Renumberer

It is usual that pre-processors renumber the nodes and elements of the mesh in order to minimize bandwidth of finite element method matrices. Usually, for linear meshes, the nodes and elements are ordered by a forward advancing front that covers all the mesh through the vertex nodes adjacency [8]. An adaption of one of this standard renumbering techniques is implemented in order to renumber properly high-order meshes.

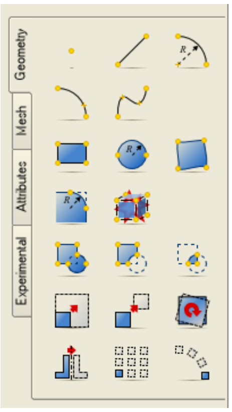


Figure 9: New icons aspect of the Geometry tab.

5 Graphical user interface

The graphical user interface is the mediator between the user and the geometry and meshing kernel. Thus, its design and implementation play a major role in the mesh environment. The main priority is to provide a clean, simple and easy to use interface that allows modeling and meshing academic examples and more complicated models. In this section we present new advances on the GUI.

5.1 View and tool icons

A new icon design is used for current revisions of the mesh environment. It is important that icons show an homogeneous and consistent style and are designed in order to be simple and sketchy but attractive at the same time. Some modern icon style guides have been taken into account in order to develop this new icon set. For instance, figure 9 shows the aspect of the icons for the geometry tools.

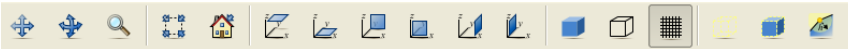


Figure 10: Aspect of the new tool bar and its icons.

Previous versions of the environment have mouse and key bindings for navigation operations such as rotate, translate and zoom. In order to simplify the user interface, the new version also provides a new icon tool bar with all the navigation and options related to the view widget. Following the icon order of the toolbar, see figure 10, we have the next groups of icons:

- **Basic navigation**. This group provide icons for activating panning, rotation and zoom modes of the view.
- **Object view recovering**. In order to view all the displayed objects we dispose of a fit all and home icons.

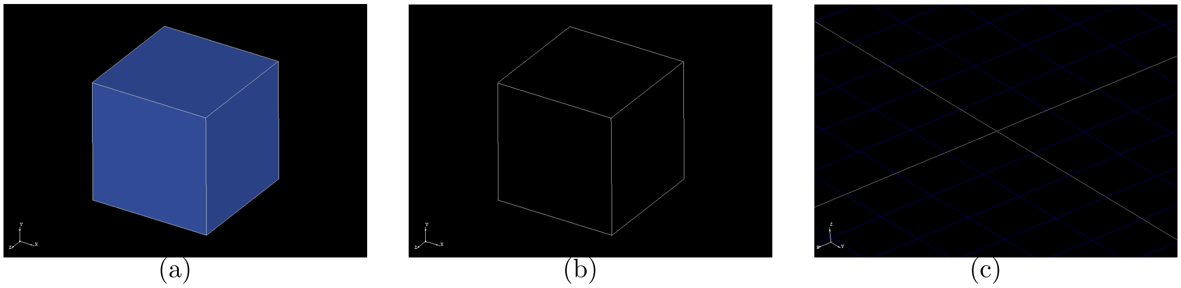

(a) (b) (c)

Figure 11: View properties examples. (a) A cube displayed in shaded mode. (b) A cube displayed in visible edges mode. (c) View of the reference grid.

- **Default views**. Six default default perspectives can be activated, view from: top, bottom, front, back, right and left view.
- **View properties**. From here we can activate the shaded or visible edges view mode. Moreover, we can activate or deactivate the reference grid with the grid toggle icon, see figure 11.
- **Object view properties**. In this group we have tools that set properties of selected objects. We can set the wireframe or shaded mode for selected objects. In addition we can display the mesh sizes applied to vertices of our model, see figure 12.

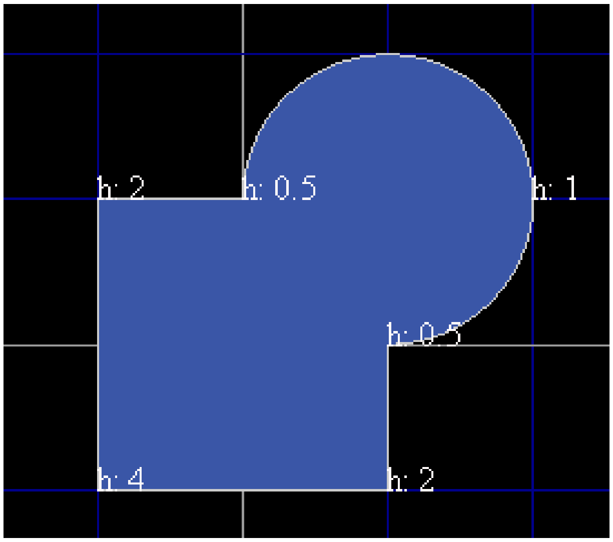


Figure 12: Several size values on the vertices of a surface.

6 Current state and future developments

Current revisions of the mesh generation environment allow to generate meshes in 2D domains that are modeled or imported into the environment. The resulting output of the program is a discrete model, mesh and attributes, that is suitable for the use with Finite Element Method routines. In particular the features to highlight are:

- Model 2D geometries.
- Import STEP files.
- Import legacy gen4u parameter files.
- Export STEP files.
- Mark the geometrical entities with labels in order to determine the attributes of the model.
- Generate triangular and quadrilateral domains.
- Assign prescribed sizes at vertices of the geometry.
- Assign prescribed sizes by means of a background mesh.
- Mesh domains with or without holes.
- Select between several meshing algorithms.
- Export a discrete model, mesh and attributes, of the continuous model.
- Generate high order meshes for $p > 1$.
- Renumber meshes, including the case of high order ones, for minimizing bandwidth of finite element method matrices.
- Call the program in batch mode without GUI, allowing mesh adaption procedures.
- Model 3D geometries.
- Work interactively with a 3D view of the model.

The mesh environment is under continuous development. Therefore several tasks are scheduled in order to improve or add new features to the package. Following we list the most significant ones:

- Import and export another geometry description formats such as IGES and STL.

- Create a native application format for opening and saving a continuous model and its associated discrete model.
- Assign scalar, vector or tensor values to geometrical entities as attributes of the model.
- Improve the quality, efficiency and robustness of the mesh generations algorithms.
- Output mesh quality after generation.
- Improve the prescribed size controlling features.
- Add 3D mesh generation algorithms such as sweeping and submapping.
- Generate anisotropic meshes over surfaces.
- Generate boundary layer structured meshes.
- Develop new mesh generation algorithms that fit better to domain boundary.
- Write a suitable output for geometry-based methods as NEFEM, [2].
- Implement a command interpreter.
- Totally decouple the environment of the geometry engine, in our case OpenCascade [9].

7 Conclusions

Nowadays we can classify mesh generation programs in two groups. On one hand, advanced and powerful commercial environments which are highly coupled with commercial solvers. On the other hand, home made tools or freeware software that does not cover some research requirements as high order meshes, several kinds of elements, non-manifold geometries and NURBS geometry kernel. Our mesh environment covers the gap between these two groups, and allows to our research team to customize the package for its own and special requirements. Our legacy code and new mesh generation algorithms are added to a centralized environment that can be used more friendly.

The package is a full 2D mesh generation environment that allows fast and easy preprocessing for numerical methods based on discretizations. In the last version, GUI, geometry, file and meshing features have been improved in order to satisfy our research pre-processing requirements.

Finally, we have presented a list of new features and improvements of the environment that are planned to be implemented. It is important to point out the continuous development and the feedback with the users has lead to an environment that covers the main goals that were planned at the first stages of its design.

REFERENCES

[1] Roca X and Sarrate J (2006) An interactive mesh generation environment for geometry-based simulations, 5th Workshop on Numerical Methods in Applied Science and Engineering (NMASE 06)

[2] Sevilla R, Fernández-Méndez S and Huerta A (2005) NURBS-Enhanced Finite Element Method, XIX Congreso de Ecuaciones Diferenciales Y Aplicaciones, IX Congreso de la Sociedad Española de Matemática Aplicada

[3] Athanasiadis AN, Deconinck H (2003) Object-oriented three-dimensional hybrid grid generation *Int. J. Numer. Meth. Engr.*58:301–318

[4] E. Gamma E, Helm R, Johnson R and Vlissides J (1993) Design Patterns, Addison-Wesley

[5] Roca X, Sarrate J, Huerta A (2006) Mesh projection between parametric surfaces *Comm. in Numer. Meth. Engr.* 22(6):591–603

[6] Roca X, Sarrate J, Huerta A (2005) A new least-squares approximation of affine mappings for sweep algorithms. Proceedings of 14th International Meshing Roundtable, 433–448

[7] Roca X, Sarrate J (2006) An automatic and general least-squares projection procedure for sweep meshing, Proceedings of 15th International Meshing Roundtable, 487–506

[8] Cuthill E, McKee J (1969) Reducing the bandwitdh of sparse symmetric matrices, Proceedings of the 1969 24th national conference, 157–172

[9] OpenCascade Technology, `http://www.opencascade.org/`