An interface between an *hp*-adaptive finite element package and the pre- and post-processor GiD

Daniel García-Doñoro [a,*], Luis E. García-Castillo [a], Ignacio Gómez-Revuelto [b]

[a] *Dep. Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Escuela Politécnica Superior (Edificio Torres Quevedo), Avenida de la Universidad, 30 28911 Leganés, Madrid, Spain*

[b] *Dep. de Ingeniería Audiovisual y Comunicaciones, Universidad Politécnica de Madrid, Spain*

A R T I C L E   I N F O

Keywords:
Finite elements
hp-Adaptivity
Geometrical modeling
User interface

A B S T R A C T

An interface between GiD, the interactive graphical user interface used for numerical simulations, developed at the International Center for Numerical Methods in Engineering (CIMNE) of the Universidad Politécnica de Cataluña and the Geometrical Modeling Package (GMP) of the fully automatic *hp*-adaptive finite element (FE) software, developed at the Institute for Computational Engineering and Sciences (ICES) of the University of Texas at Austin, is presented. GiD is used to construct a tessellation of the problem domain into FE-like regions (blocks in GMP terminology), and the interface obtains and transfers all the topological and geometrical information to GMP. Then, GMP automatically constructs a parameterization for each FE-like region of the GMP mesh, which later can be used to generate the actual FE-mesh and support geometry updates during mesh refinements.

1. Introduction

The finite element method (FEM) is based on a variational formulation of the problem that enables the use of "adapted" meshes, not only to the geometry of the problem domain but also to the solution of the problem itself. In this way, accurate solutions are obtained with a minimum number of unknowns [1].

The process to obtain the "adapted" meshes is typically an iterative process that generates a sequence of meshes, often by refining the previous mesh, with an error below that of the previous one. When the sequence of meshes are generated automatically, the process is called self-adaptive process or *self-adaptivity*.

One of the most powerful methods to generate these "adapted" meshes is through *hp*-adaptivity (i.e., by simultaneously varying the size *h* and the polynomial order *p* of the finite elements of the mesh). An example of *h*, *p* and *hp*-adaptivity is shown in Fig. 1. This figure shows how *h*-adaptivity modifies the size and density of the finite elements of the mesh, *p*-adaptivity modifies the polynomial order of several finite elements of the mesh, and *hp*-adaptivity modifies *h* and *p* simultaneously.

hp-adaptivity provides exponential rates of solution convergence, even in the presence of singularities, in contrast to *h* and *p* schemes, in which algebraic rates of solution convergence are, in general, obtained. Thus, very accurate solutions can be obtained with an *hp*-adaptive strategy, even in the presence of singularities. Equivalently, approximate solutions within engineering accuracy can be obtained using a minimum number of unknowns.

In this field, it is worth nothing a fully automatic *hp*-adaptive implementation, developed at the Institute for Computational Engineering and Sciences (ICES) of the University of Texas at Austin, in which some of the authors of this document have contributed. This implementation or *hp*-code provides a sequence of *hp*-meshes supporting anisotropic refinements, 1-irregular meshes (i.e., with *hanging nodes*), and isoparametric elements for one dimensional (1D), two dimensional (2D), and three dimensional (3D), elliptic and Maxwell problems. Currently, the reader can obtain the *hp*-code to solve 2D problems in [2]. This book also contains all the formulations, developments and demonstrations of the code. The *hp*-code for 3D problems is under intense development. For recent advances the reader is referred to [3].

An *hp*-mesh obtained with the *hp*-code is shown in Fig. 2(c). Different colors indicate, according to the scale on the right, the polynomial order *p* of the finite elements (the dark blue being $p = 1$ and the pink $p = 8$). The mentioned *hp*-mesh is generated from the initial mesh shown in Fig. 2(b). It is observed how, despite the approximation of the arc of a circle (used to model the round corner of the structure) by two straight lines in the initial mesh, the adaptivity has been able to generate a mesh that provides a very good approximation of the curved contour. This is possible because geometry updates are allowed during mesh refinements.

* Corresponding author. Tel.: +34 91 6249171; fax: +34 91 6248749.
E-mail addresses: dgdonoro@tsc.uc3m.es (D. García-Doñoro), luise@tsc.uc3m.es (L.E. García-Castillo), igomez@diac.upm.es (I. Gómez-Revuelto).

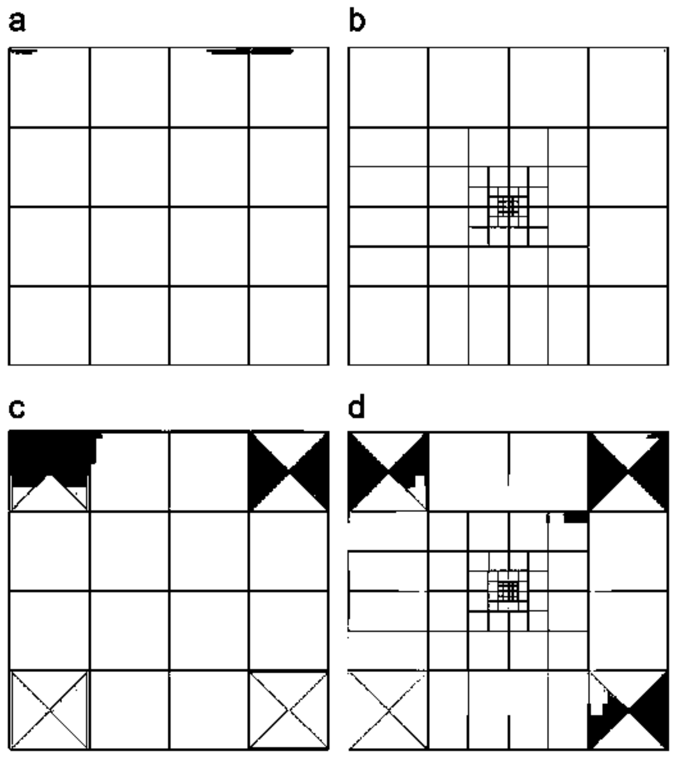


Fig. 1. Illustration of *h*, *p* and *hp* adaptivity. Color indicates polynomial order *p*. Note: The meshes are simple for illustration purposes and they have not been obtained with any numerical code. (a) Initial mesh. (b) *h*-adaptivity. (c) *p*-adaptivity. (d) *hp*-adaptivity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

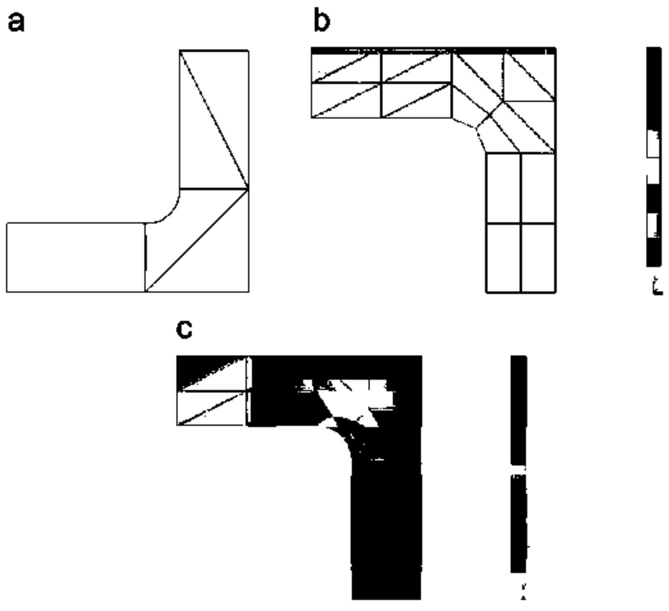


Fig. 2. Illustration of *hp*-mesh refinement. (a) Initial geometry. (b) Initial *hp*-mesh. (c) Optimal *hp*-mesh delivered by *hp*-code.

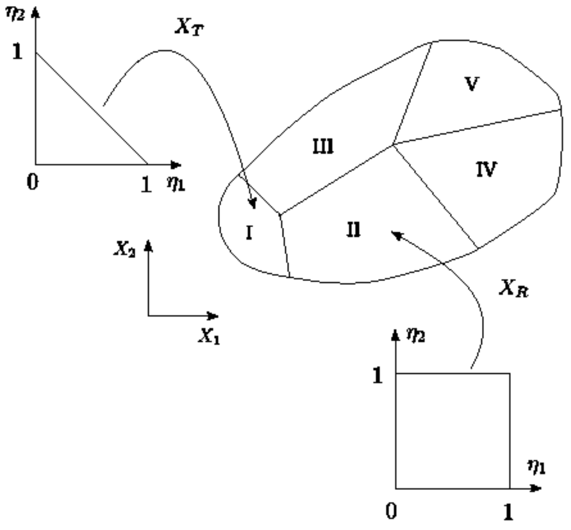


Fig. 3. GMP modeling of 2D structure.

That important feature requires the knowledge of the exact geometry of the structure. For that purpose, the so-called Geometrical Modeling Package (GMP) [4] is used by the *hp*-code to support geometry information, independently of the mesh.

GMP model structures as a combination of blocks (triangles, rectangles, prisms, hexahedrons, etc.), each block being defined in terms of explicit or implicit parameterizations. Thus, a FE-like mesh is obtained. An example in 2D is shown in Fig. 3. However, it is important to note that the actual FE-mesh for the analysis is generated from GMP information (the FE-like mesh) by using a multi-block *hp*-mesh generator integrated in the *hp*-code.

GMP defines seven different objects for geometrical modeling at time of issue of [2]: *surfaces*, *points*, *curves*, *triangles*, *rectangles*, *prisms* and *hexahedrons*. In addition, *Tetrahedrons* and *Prisms* are being developed at this moment for 3D problems. The definitions of these objects include complete geometrical and topological (connectivity) information. This information is entered into GMP by a text file (ASCII) that can be written in several formats. This file can be generated by hand. However, for complex engineering structures, this process is very slow and costly. To better illustrate this complexity, an example of GMP text file corresponding to the geometry of Fig. 2(a) is shown in File 1. In this context, the automatic generation of the GMP file arises as an important objective in order to improve the usability of the *hp*-code. This document presents an interface between GiD and GMP. GiD [5] is an interactive graphical user interface employed for the definition, preparation and visualization of all the data related to a numerical simulation. These data include the definition of the geometry, materials, conditions, solution information and other parameters. The program can generate a mesh for finite element, finite volume or finite differences analysis and write the information for a numerical simulation program in its desired format. It is also possible to run these numerical simulations from within GiD and then visualize the results of the analysis. It should be noted that the last feature is not used in this work because the *hp*-code contains its own process to visualize the result of the analysis, so this work only uses the pre-processing facilities of GiD.

File 1: Example of GMP text file.

```
     2      2       ...NDIM,MANDIM


         0          ...NUMBER OF SURFACES


         9          ...NUMBER OF POINTS

1                   ...TYPE OF POINT 1
3                   ...CONNECTED CURVES
1  5  7             ...CURVES NUMBERS
0.000 -1.000        ...COORDINATES

1                   ...TYPE OF POINT 2
3                   ...CONNECTED CURVES
1  2  12            ...CURVES NUMBERS
-1.000 0.000        ...COORDINATES

1                   ...TYPE OF POINT 3
5                   ...CONNECTED CURVES
2  3  6  10  13     ...CURVES NUMBERS
-1.000 2.000        ...COORDINATES

1                   ...TYPE OF POINT 4
2                   ...CONNECTED CURVES
3  4                ...CURVES NUMBERS
2.000 2.000         ...COORDINATES

1                   ...TYPE OF POINT 5
4                   ...CONNECTED CURVES
4  5  6  9          ...CURVES NUMBERS
2.000 -1.000        ...COORDINATES

1                   ...TYPE OF POINT 6
2                   ...CONNECTED CURVES
7  8                ...CURVES NUMBERS
0.000 -5.000        ...COORDINATES

1                   ...TYPE OF POINT 7
2                   ...CONNECTED CURVES
8  9                ...CURVES NUMBERS
2.000 -5.000        ...COORDINATES

1                   ...TYPE OF POINT 8
2                   ...CONNECTED CURVES
10  11              ...CURVES NUMBERS
-5.000 2.000        ...COORDINATES

1                   ...TYPE OF POINT 9
3                   ...CONNECTED CURVES
11  12  13          ...CURVES NUMBERS
-5.000 0.000        ...COORDINATES

         13         ...NUMBER OF CURVES

-2                  ...TYPE OF CURVE 1
1  2                ...END POINTS
1                   ...CONNECTED FIGURES
-22                 ...FIGURES NUMBERS
-1.000 -1.000       ...CENTER COORDINATES

1                   ...TYPE OF CURVE 2
```

```
2   3               ...END POINTS
2                   ...CONNECTED FIGURES
-22   31            ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 3
3   4               ...END POINTS
1                   ...CONNECTED FIGURES
-11                 ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 4
4   5               ...END POINTS
1                   ...CONNECTED FIGURES
-11                 ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 5
5   1               ...END POINTS
2                   ...CONNECTED FIGURES
12   -22            ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 6
5   3               ...END POINTS
2                   ...CONNECTED FIGURES
22   -11            ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 7
1   6               ...END POINTS
1                   ...CONNECTED FIGURES
12                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 8
6   7               ...END POINTS
1                   ...CONNECTED FIGURES
12                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 9
7   5               ...END POINTS
1                   ...CONNECTED FIGURES
12                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 10
3   8               ...END POINTS
1                   ...CONNECTED FIGURES
21                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 11
8   9               ...END POINTS
1                   ...CONNECTED FIGURES
21                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 12
9   2               ...END POINTS
1                   ...CONNECTED FIGURES
31                  ...FIGURES NUMBERS

1                   ...TYPE OF CURVE 13
9   3               ...END POINTS
2                   ...CONNECTED FIGURES
21   -31            ...FIGURES NUMBERS


        3           ...NUMBER OF TRIANGLES

1                   ...TYPE OF TRIANGLE 1
```

```
 -6   -4   -3          ...THREE CURVES NUMBERS

 1                     ...TYPE OF TRIANGLE 2
 10   11   13          ...THREE CURVES NUMBERS

 1                     ...TYPE OF TRIANGLE 3
 2   -13   12          ...THREE CURVES NUMBERS


         2             ...NUMBER OF RECTANGLES

 1                     ...TYPE OF RECTANGLE 1
 5   7   8   9         ...FOUR CURVES NUMBERS

 2                     ...TYPE OF RECTANGLE 2
 -5   6   -2   -1      ...FOUR CURVES NUMBERS



 2000000               ...FRONTAL SOLVER


 5000 5000 20000 1 ...GENERAL OPTIONS


 2 3                   ...SUBDIV AND APPROX
 000                   ...BOUNDARY CONDITIONS
 2 3                   ...SUBDIV AND APPROX
 000                   ...BOUNDARY CONDITIONS
 2 3                   ...SUBDIV AND APPROX
 000                   ...BOUNDARY CONDITIONS
 2 2 23                ...SUBDIV AND APPROX
 0000                  ...BOUNDARY CONDITIONS
 2 2 23                ...SUBDIV AND APPROX
 0000                  ...BOUNDARY CONDITIONS

 11
 21
 31
 12
 22                    ...ORDER OF FIGURES


 0                     ...NUMBER OF PROFILES
```

To create this interface, a module (*problem type*, in GiD terminology) named *GiDtohp* has been developed. A tessellation of the problem domain into FE-like regions (blocks in GMP terminology) is constructed using GiD tools extended by *GiDtohp* module. Once the blocks are defined, the logic implemented in *GiDtohp* obtains and transfers all the topological and geometrical information to GMP (by writing the GMP input text file mentioned above). Then, GMP automatically constructs a parameterization for each FE-like region of the GMP mesh, which later can be used to generate the actual FE-mesh and support geometry updates during mesh refinements. *GiDtohp* provides two different modes: *geometry mode* and *mesh mode*, to obtain and transfer all the topological and geometrical information to GMP. In the *geometry mode*, each of the FE-like regions or blocks (surfaces in 2D and volumes 3D) are created directly by the user employing available GiD drawing tools and utilities. In the *mesh mode*, a mesh is generated by GiD using the boundary of the present geometry. This last mode is oriented to structures that require a large number of blocks, large enough to render their manual definition impractical.

Until present, work on this research line had been focused on using several FE mesh generators, as shown in [6]. In this work,

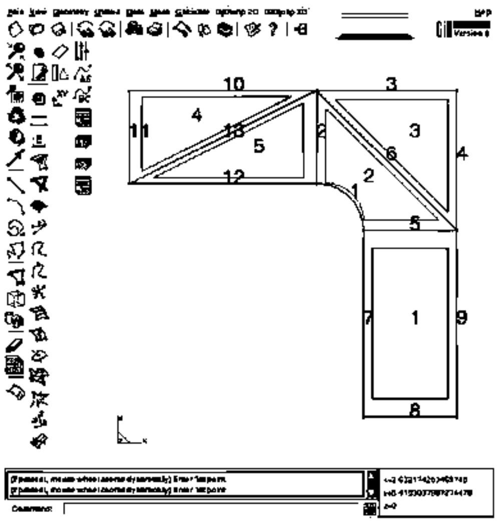
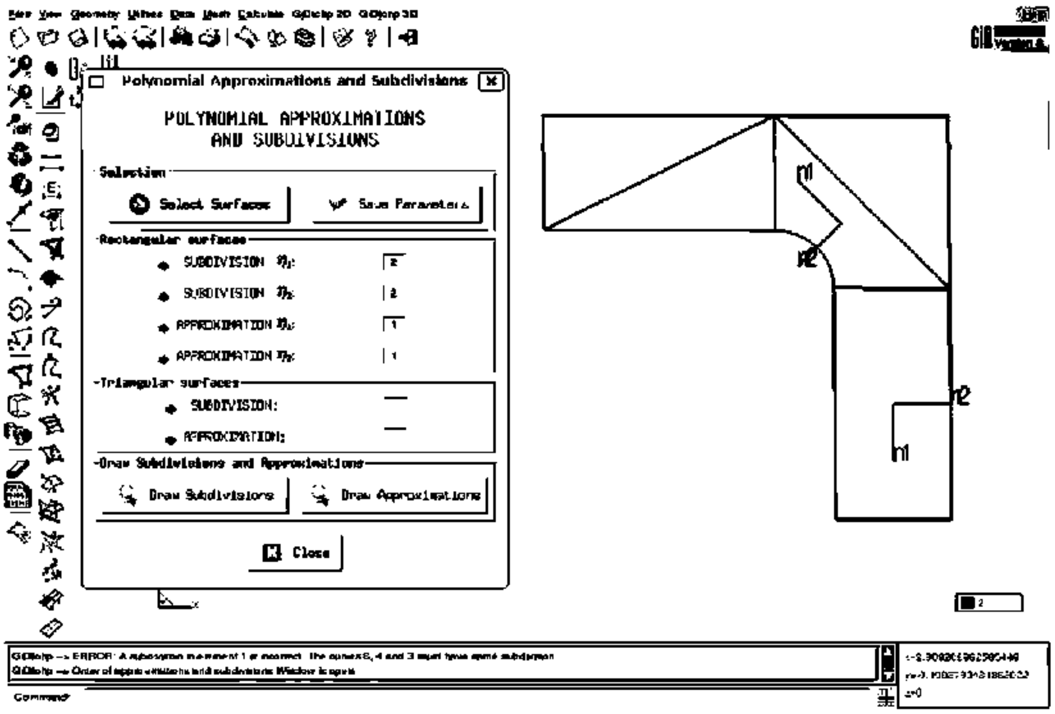


Fig. 4. *Geometry mode* for 2D structure.

the transfer of all topological and geometrical information is made using CUBIT [7], a mesh generator developed at Sandia National Labs, and GMP is used to construct a parameterization for each FE-like region of the mesh. The transfer of all information into GMP is, in some way, equivalent to the *mesh mode* presented here. However, [6] does not offer the possibility to create the GMP blocks without the mesher (CUBIT), option that *GiDtohp* supports in its *geometry mode*. Another improvement of the current work is that *GiDtohp* adds a new mathematical representation model in GMP named NURBS (non-uniform rational B-splines) [8]. This new model allows to define a wide range of structures, from simple lines in 2D to the most complex volumes and surfaces in 3D. In this way, a powerful feature of GiD is used, since GiD works almost entirely with NURBS.



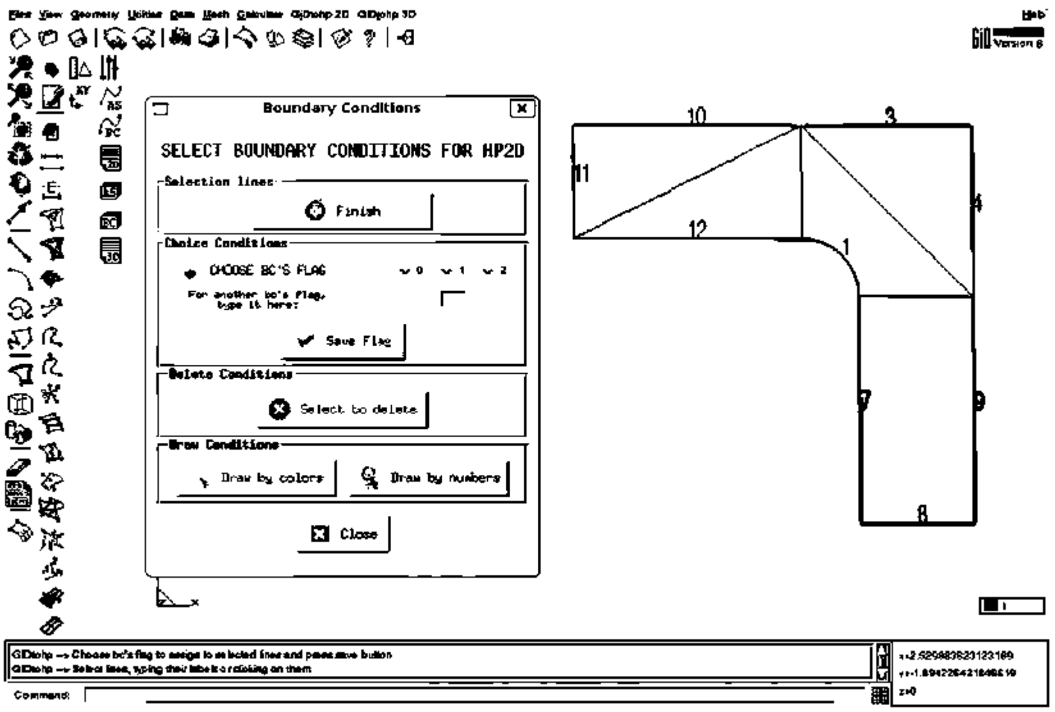
File View Geometry Utilities Data Mesh Calculate GiDtohp 2D GiDtohp 3D
Polynomial Approximations and Subdivisions
POLYNOMIAL APPROXIMATIONS AND SUBDIVISIONS
Selection
Select Surfaces
Save Parameters
Rectangular surfaces
Triangular surfaces
Draw Subdivisions and Approximations
Draw Subdivisions
Draw Approximations
Close
n1
n2
n2
n1
Command

SUBDIVISION η_1: 2
SUBDIVISION η_2: 2
APPROXIMATION η_1: 1
APPROXIMATION η_2: 1
SUBDIVISION:
APPROXIMATION:

GiDtohp --> ERROR: A subdivision in element 1 is incorrect. The curves 8, 4 and 3 must have same subdivision
GiDtohp --> Order of approximations and subdivisions Window is open

z=0

Fig. 5. Assignment of subdivisions and polynomial order for future finite elements of the blocks.

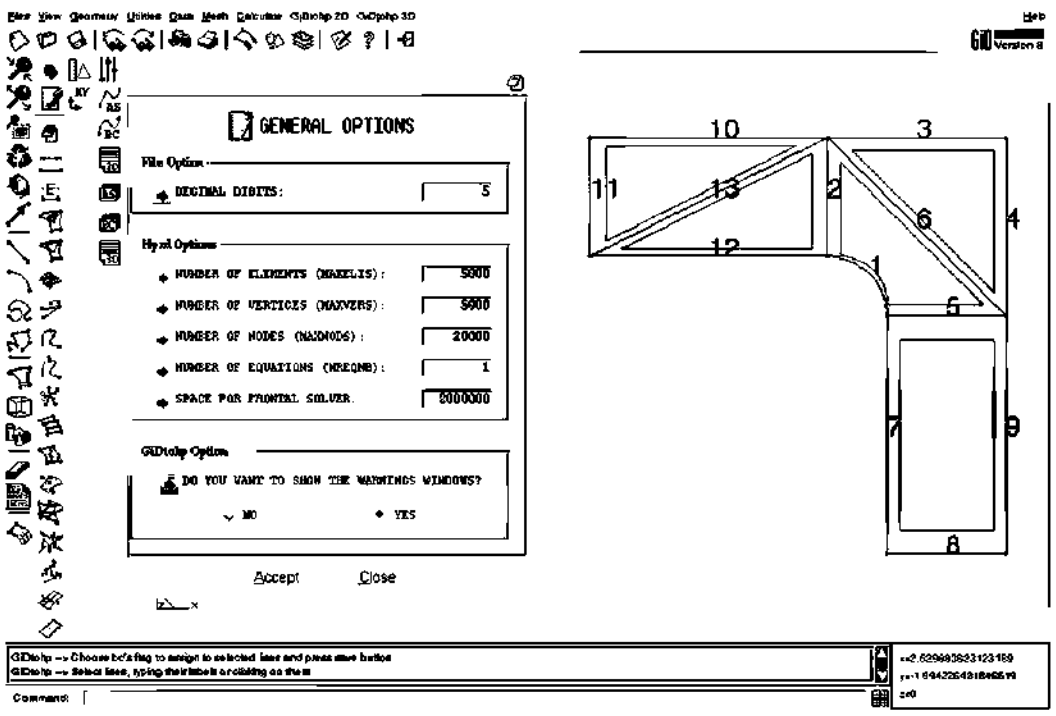


CHOOSE BC'S FLAG ✓0 ✓1 ✓2
For another bc's flag, type it here:

GiDtohp --> Choose bc's flag to assign to selected lines and press save button
GiDtohp --> Select lines, typing their labels or clicking on them

x=2.525483820123189
y=-1.89422642184866 19
z=0

Fig. 6. Assignment of boundary conditions labels for boundary blocks.


File View Geometry Utilities Data Mesh Calculate GiDtohp 2D GiDtohp 3D
Help
GiD Version 8
10
3
11
13
2
6
4
12
1
5
7
9
8
Accept
Close

GENERAL OPTIONS

File Options

DECIMAL DIGITS: 5

Hp2d Options

NUMBER OF ELEMENTS (MAXELIS): 5000

NUMBER OF VERTICES (MAXVERS): 5000

NUMBER OF NODES (MAXNODS): 20000

NUMBER OF EQUATIONS (MREQNS): 1

SPACE FOR FRONTAL SOLVER. 8000000

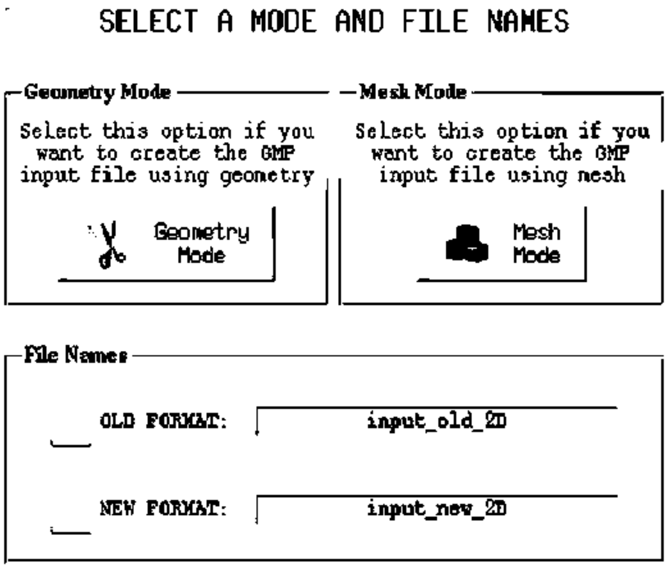
GiDtohp Option

DO YOU WANT TO SHOW THE WARNINGS WINDOWS?

NO [✓ YES]

```
GiDtohp --> Choose bc's flag to assign to selected lines and press assign button
GiDtohp --> Select lines, typing their labels or clicking on them
Command:
x=2.629693623123159
y=1.594226431846519
z=0
```

Fig. 7. Assignment of general options.



SELECT A MODE AND FILE NAMES

Geometry Mode

Select this option if you want to create the GMP input file using geometry

Geometry Mode

Mesh Mode

Select this option if you want to create the GMP input file using mesh

Mesh Mode

File Names

OLD FORMAT: input_old_2D

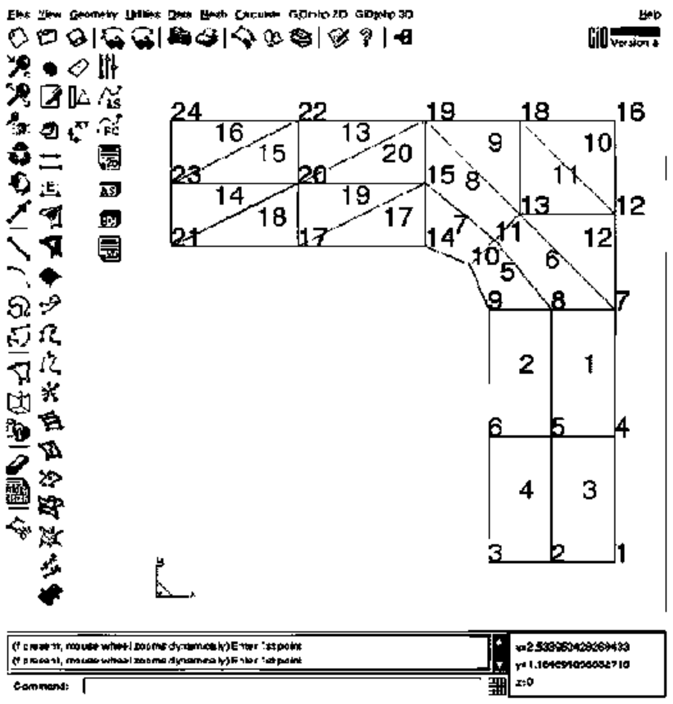
NEW FORMAT: input_new_2D

Fig. 8. 2D modes selection window.

As it will be shown later, *GiDtohp* provides a simple, and easy to use, graphical interface to GMP modeling, extending the capabilities and flexibility of previous approaches. This is mainly due to the graphical and customization features of the general pre- and post-processor used, namely GiD.

2. *GiDtohp* features

As mentioned above, there are two main working modes for *GiDtohp* interface. In the first one (*geometry mode*), the user employs available GiD drawing tools and utilities to create the FE-like regions or blocks of the geometry.



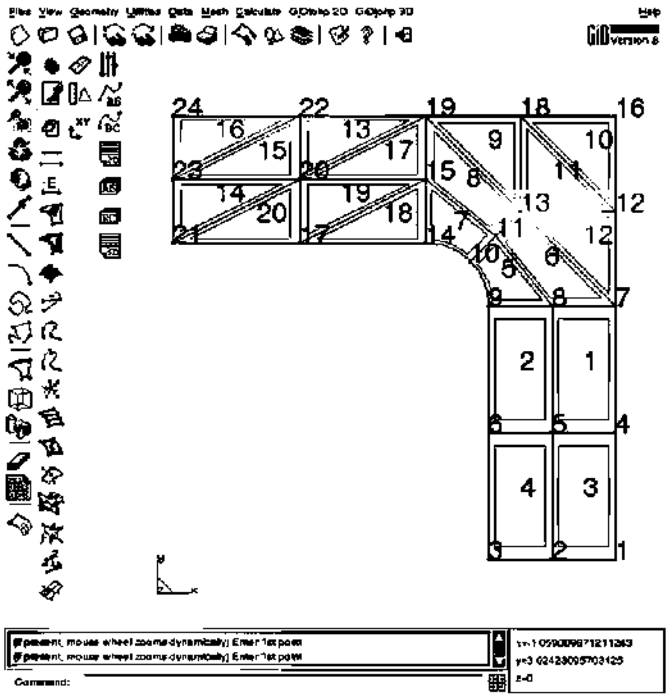
```
(f cmesh, mouse wheel zooms dynamically) Enter 1st point
(f cmesh, mouse wheel zooms dynamically) Enter 1st point
Command:
x=2.533950428269433
y=1.164691056052710
z=0
```

Fig. 9. *Mesh mode* for 2D structure.

Fig. 4 shows a 2D example of this working mode in which we observe GiD points, lines, and surfaces. Those GiD entities, and the connectivity information associated to them, are transformed into GMP points, curves, rectangles, triangles, and connectivity information compatible with GMP. Parameterizations for each triangle or rectangle (the blocks for the 2D case) are constructed by GMP. From them, an initial *hp*-mesh is generated (see Fig. 2(b)).

The module contains specific procedures using TCL-TK and TKWidget [9], providing a graphical and comfortable environment to the user for several tasks. For instance, the assignment to each block of the number of subdivisions that will be used to generate the initial *hp*-mesh is illustrated in Fig. 5. The user may select with the mouse surfaces (2D) or volumes (3D) to which one wants to assign the number of subdivisions. The user must select surfaces of the same type, i.e., the user selection must include either only triangular or only rectangular type surfaces. This restriction is imposed by GMP, because GMP defines the number of subdivisions for rectangles and triangles differently.

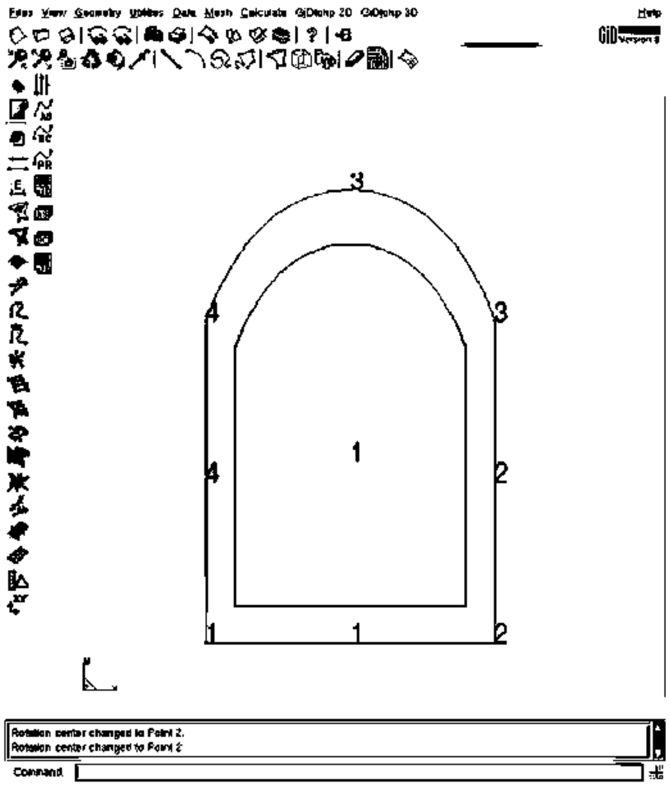
GiDtohp has been programmed to provide the user a robust and stable interface. It contains an additional logic that checks the data entered by the user to avoid mistakes. For instance, when the user selects surfaces (or volumes), in order to assign them a number of subdivisions as shown in Fig. 5, *GiDtohp* checks if all entities selected are of the same geometrical type. If the user's selection is right, and only in that case, the data entries of the window are activated. Thus, inconsistencies in the GMP input file are avoided.



```
[pan, mouse wheel zooms dynamically] Enter 1st point
[pan, mouse wheel zooms dynamically] Enter 1st point
```

```
x=1.05900987121126
y=3.02423095703125
z=0
```

Fig. 10. Auxiliary geometry of *mesh mode*.


Files View Geometry Utilities Data Mesh Calculate GiDtohp 2D GiDtohp 3D
Help
GiD Version 8
3
4
3
1
4
2
1
1
2
Command

```
Rotation center changed to Point 2.
Rotation center changed to Point 2
```

Fig. 11. Geometry with a NURBS created by GiD.

Fig. 12. *hp*-Mesh (isoparametric element order $p = 1$).

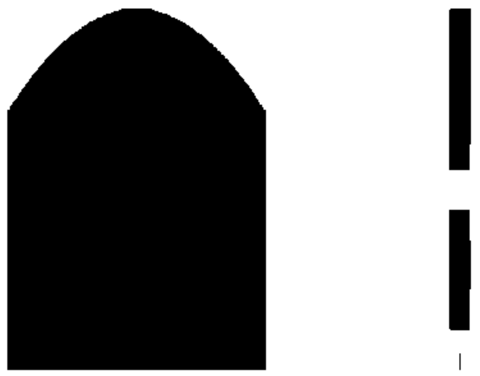


Fig. 13. *hp*-Mesh (isoparametric element order $p = 2$).

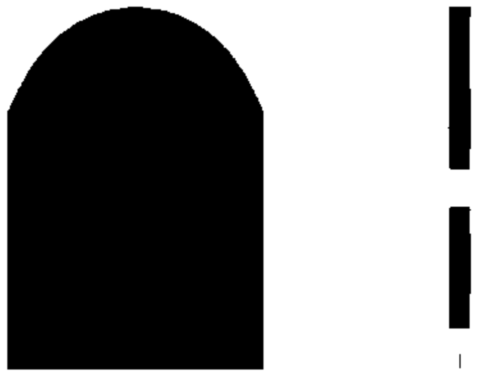


Fig. 14. *hp*-Mesh (isoparametric element order $p = 6$).

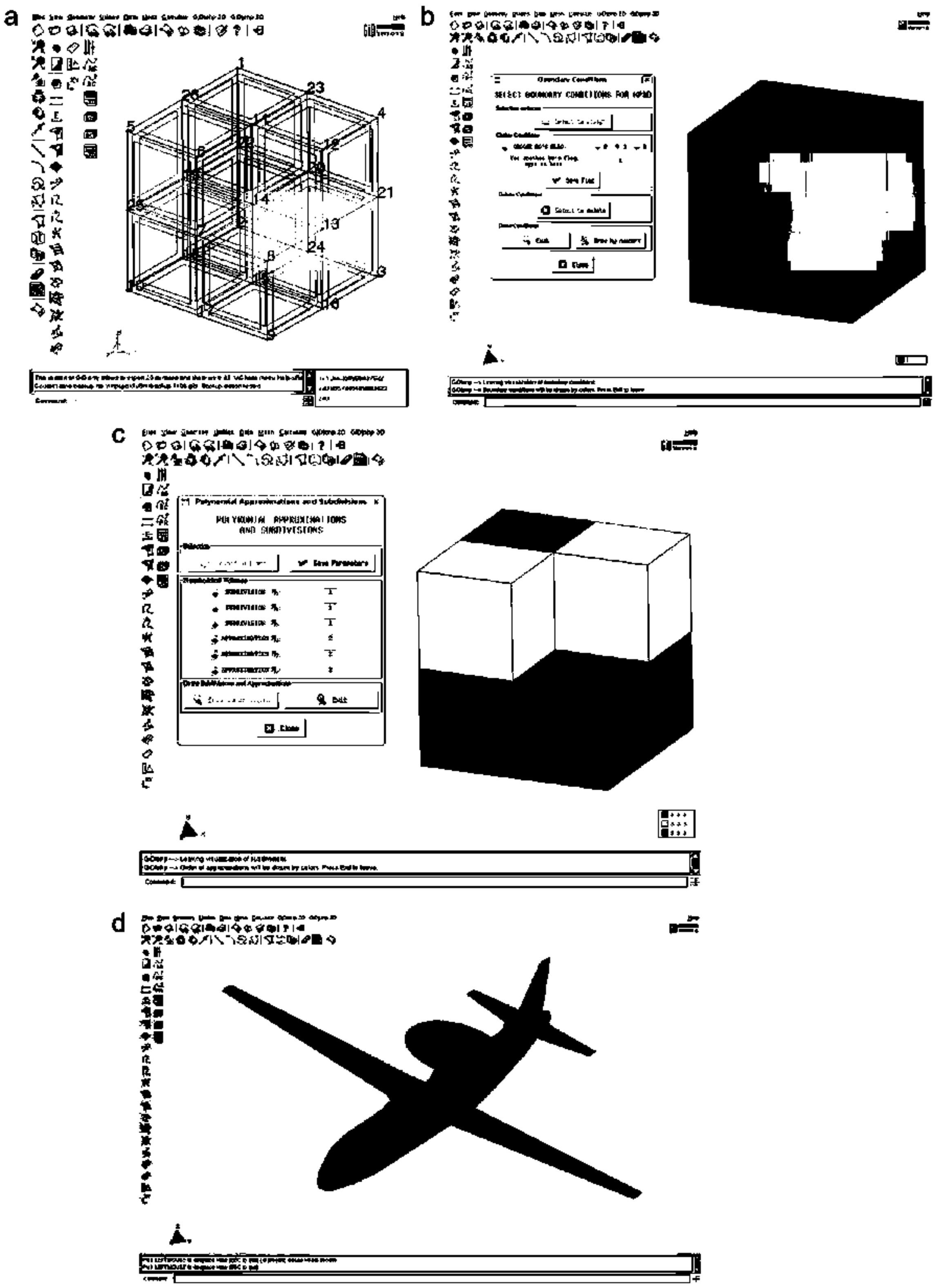

a
b
c
d
1
23
26
4
5
12
21
13
24
3
8
16
Boundary Conditions
SELECT BOUNDARY CONDITIONS FOR HP3D
Polynomial Approximations and Subdivisions
POLYNOMIAL APPROXIMATIONS AND SUBDIVISIONS
Save Parameters
Close

Fig. 15. Some screenshots of *GiDtohp* for 3D. (a) Example of 3D structure (Fichera's corner). (b) Assignment of boundary conditions labels for 3D version. (c) Assignment of subdivisions and polynomial order for 3D version. (d) Example of 3D complex structure by GiD.

The polynomial order associated to each block, which will be used later to set the order of the FE of the initial *hp*-mesh, is also introduced using the same window (see Fig. 5). In this way, the user can assign simultaneously the value of both parameters: number of subdivisions and polynomial order of each block. Note that in our example the polynomial order has been set to one for all blocks. Also, *GiDtohp* provides graphical procedures to assign boundary conditions. The user must select boundary lines in 2D or boundary surfaces in 3D to assign them boundary conditions labels (specifically, numerical labels). The user may perform this action through the window shown in Fig. 6.

Other GMP and *hp*-code parameters needed for the analysis are also introduced by using specific windows. The user can choose the values of some *hp*-code parameters, and some general options of *GiDtohp* by using the window shown in Fig. 7. The *hp*-code parameters are related to memory allocation of several data structures (number of initial elements, vertices, and so on); at present, the GMP input file includes these parameters, which are not strictly related to GMP. Other options that may be set by the window shown in Fig. 7 are the number of decimal digits used in the representation of real numbers in the GMP input files (values of coordinates, arcs, and so on), and a flag to control the appearance of a number of pop-up windows with different warning messages.

Once the user has completed the definition of the blocks, one may execute the option to generate the GMP input text files by selecting the *geometry mode* button on the left part of the window shown in Fig. 8. An example of *hp*-mesh generated by the *hp*-code from the initial mesh of Fig. 2(b) is shown in Fig. 2(c). Geometry refinements around the curved bend are observed.

The other working mode (*mesh mode*) allows to generate the blocks of the present geometry automatically. To do this, a mesh is generated by GiD using the boundary of this geometry. This mode is named *mesh mode* because it makes use of the mesh capabilities of GiD. An example is shown in Fig. 9. Each of the finite elements of the GiD mesh will become a block for GMP. This process is performed when selecting the *mesh mode* button on the right part of the window shown in Fig. 8. However, the process can not be directly implemented using the GiD mesh. This is due to the fact that the GiD mesh data structure only stores information about finite element connectivity (using the element nodes) and the node coordinates. The *hp*-code needs complete geometrical information in terms of points, lines, arcs of a line, and so on; furthermore, information about boundary conditions labels, and other information associated later to the blocks, must be retained. For this purpose, an auxiliary *geometry* (a geometry data structure in the GiD sense) is created on a new layer of GiD being not visible by the user. This auxiliary geometry data structure is created using the information of the GiD mesh data structure and the original geometry data structure. The mentioned auxiliary geometry is used by *GiDtohp* to generate the GMP text files following almost identical procedures as those used for the *geometry mode*. The auxiliary geometry corresponding to the example of Fig. 9 is shown in Fig. 10. For instance, it is illustrative to comment that the straight lines between nodes 14–10 and 10–9 of the GiD mesh are converted to arcs of a circle between points 14–10 and 10–9 in the auxiliary GiD geometry. It is worth remarking that the auxiliary geometry is never seen by the user and that the illustration of Fig. 10 was obtained by tweaking *GiDtohp*.

The 2D version of the interface supports straight lines, arcs and NURBS. To illustrate the use of NURBS in *GiDtohp*, a geometry with a NURBS (line 3) is shown in Fig. 11. Its corresponding initial mesh with GMP blocks is shown in Fig. 12 for order $p = 1$, in Fig. 13 for order $p = 2$, and in Fig. 14 for order $p = 6$. Note how the line in Fig. 12 is a straight line because the order of approximation of the element is $p = 1$ (linear approximation). As we increase the order of approximation of the element, the NURBS of the *hp*-mesh presents a better behavior and the quality of the approximation improves markedly. To illustrate this matter, approximation for element orders $p = 2$ and 6 are shown in Figs. 13 and 14, respectively.

The 3D version is an extension of the module in 2D. *GiDtohp* presents the same features in both cases. As for the 2D version, in 3D the user may create the geometry to analyze (with *geometry mode* or *mesh mode*), and further the user may assign the boundary conditions of the problem or the subdivision and the order of approximation to each volume of the geometry. The only restriction that the 3D version presents at the moment, is that it does not support non-planar surfaces and the only blocks available are hexahedrons. In the future, and when GMP supports other kind of volumetric elements, such as tetrahedrons or prisms, *GiDtohp* will be updated to support them. Some screenshots of this version are shown in Fig. 15. A complex structure (Boeing 767) created with *GiDtohp* in *geometry mode* is shown in Figs. 16 and 17. This aircraft was created by creating all volumes within GiD (see Fig. 16) and exporting the information to GMP input files. Then, the *hp*-code is able to generate an initial mesh using GMP parameterizations (see Fig. 17).

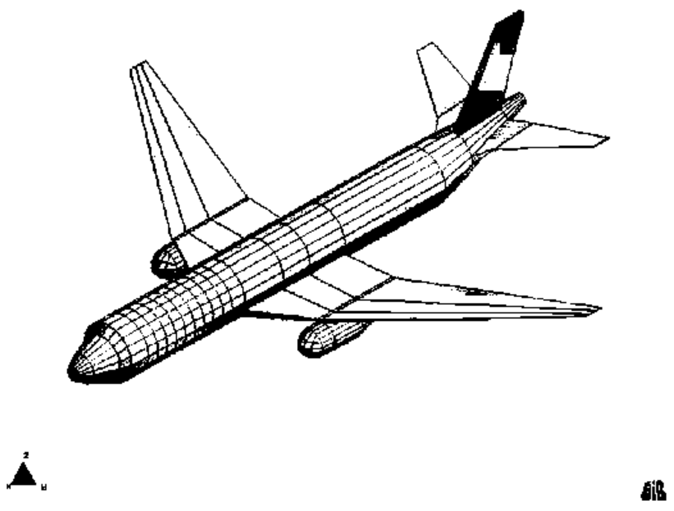


Fig. 16. *Geometry mode* for a complex 3D structure (Boeing 767).

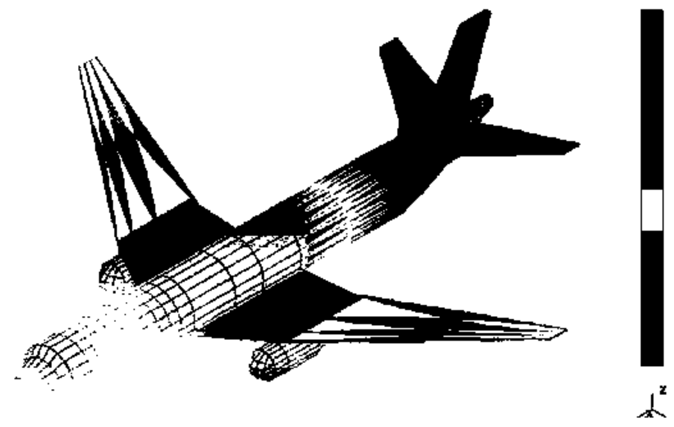


Fig. 17. Initial *hp*-mesh for the 3D structure (Boeing 767) of Fig. 16.

3. Conclusions

An interface between GiD and the Geometrical Modeling Package (GMP) of the fully automatic *hp*-adaptive FE software, developed at the University of Texas at Austin, has been presented. The main features of the interface, named as *GiDtohp*, have been described, with the help of a number of illustrations and screenshots. *GiDtohp* provides a modern, comfortable and robust environment for the user to produce geometrical input to the *hp*-adaptive code in such a way that geometry updates can be performed during mesh refinements. *GiDtohp* is available from the authors.

Acknowledgment

The authors want to acknowledge the support of Ministerio de Educación y Ciencia of Spain under project TEC2007-65214/TCM.

References

[1] M. Salazar-Palma, T.K. Sarkar, L.E. García-Castillo, T. Roy, A.R. Djordjevic, Iterative and Self-adaptive Finite-elements in Electromagnetic Modeling, Artech House Publishers Inc, Norwood, MA, 1998.

[2] L. Demkowicz, Computing with hp Finite Elements. I. One- and Two-Dimensional Elliptic and Maxwell Problems, Chapman & Hall, CRC Press, Taylor and Francis, London, Boca Raton, FL, 2007.

[3] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, A. Zdunek, Computing with hp Finite Elements. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications, Chapman & Hall, CRC Press, Taylor and Francis, London, Boca Raton, FL, 2008.

[4] D. Xue, L.F. Demkowicz, Geometrical modeling package, version 2.0, Technical Report 30, Institute for Computational Engineering and Sciences, August 2002.

[5] GiD: The personal pre and postprocessor, International Center for Numerical Methods in Engineering (CIMNE). URL: ⟨www.gidhome.com⟩.

[6] D. Xue, L.F. Demkowicz, A. Zdunek, An interface between Geometrical Modeling Package (GMP) and mesh-based-geometry (MBG), Technical Report 20, Institute for Computational Engineering and Sciences, 2003.

[7] S.N. Laboratories, The CUBIT geometry and mesh generation toolkit. URL: ⟨http://cubit.sandia.gov/⟩.

[8] G. Farin, Curves and Surfaces for CAGD. A Practical Guide, fifth ed., Academic Press Inc., New York, 2001.

[9] B. Welch, K. Jones, J. Hobbs, Practical Programming in Tcl and Tk, fourth ed., Prentice-Hall, Englewood Cliffs, NJ, June 2003.