

# **Interfacing MSC/PATRAN with External Surface Meshers**

R. Shanmugasundaram and Jamshid Samareh  
Computer Sciences Corporation  
3217 North Armistead Avenue  
Hampton, Virginia 23666

## **ABSTRACT:**

This paper describes interfacing external unstructured meshers with MSC/PATRAN. The developed interface called GUMPI(Geolab Unstructured Mesher/Patran Interface) is a collection of PCL(Patran Command Language) and ANSI-C codes that manages the data between MSC/PATRAN and an external unstructured mesher. The details of the data management is transparent to the user. However, efforts were made so that the user inputs to the external mesher is a simple to use as the internal meshing capabilities within MSC/PATRAN. Specifically, two different advancing front unstructured meshers called FELISA (Finite Element London Imperial Swansea Ames) and Vgrid have been interfaced with MSC/PATRAN through GUMPI. The interface demonstrates the ease at which external codes can be interfaced to MSC/PATRAN.

## INTRODUCTION

In an effort to link an adaptive surface mesher to MSC/PATRAN, an interface was developed called GUMPI. GUMPI is a collection of PCL and ANSI-C codes that manages the data between MSC/PATRAN and a user-supplied unstructured mesher. The ability to use external unstructured meshers within the MSC/PATRAN environment has several benefits. One is that user can exploit the meshing properties inherent to a specific mesher. For example, GUMPI currently supports a code called FELISA[1]. FELISA is a system that supports adaptive meshing. Another advantage is that many organizations have developed in-house unstructured meshers to fit within their own design processes. GUMPI provides a method so that MSC/PATRAN along with these meshers can be easily interfaced to existing analysis processes. Specifically, GUMPI automatically exports the surface geometry from the MSC/PATRAN database, executes the designated unstructured mesher, and imports the finite-element model back into the MSC/PATRAN database. The GUMPI process is transparent to the user. When MSC/PATRAN is executed normally, GUMPI appears as an additional menu. However, a session script has also been written for MSC/PATRAN batch mode operations. GUMPI was written so that additional unstructured meshers could be supported with minimum effort. For example, a second unstructured mesher called Vgrid[2] was quickly interfaced within the GUMPI system. In essence, GUMPI launches the appropriate application and expects a success flag upon return before launching the next code. This paper will discuss the various issues associated with developing the GUMPI interface.

## METHOD

GUMPI performs three main functions. One function is to obtain the necessary geometric and spacing information in order to mesh a given geometry. The second function is to launch the appropriate applications in order to generate and import the unstructured mesh. This function involves launching and managing various applications. The third function is to retain the characteristics of MSC/PATRAN environment and to minimize the user interaction. Figure 1 shows the GUMPI flow process diagram.

### USER INPUTS:

The first objective of GUMPI is obtain the user inputs. Figure 2 shows the created form. The user inputs must be simple to use and yet flexible enough to handle various geometry. GUMPI is designed to mesh all the surfaces within the current database. The main idea is to obtain a mesh quickly with minimum user interaction. The resulting mesh can be refined later. Consequently, the mesh spacing is determined by the number of points on the minimum length edge within the geometry as shown in Figure 2, item 2. This is similar to the global length parameter used within MSC/PATRAN when the mesh seeds are not defined. The graphical menu is implemented and managed using several PCL routines. One of the key features of the form is that default values for all the parameters are automatically set. In this way, a user will quickly recognize the range or type of parameters required. Also, the user needs to fill data boxes only as needed. Another advantage is that less error checking is required if every data box always has a value.

## **APPLICATION MANAGEMENT:**

The GUMPI system needs to manage five separate tasks in order to generate the unstructured mesh. Each task is associated with a separate routine or application which is managed by GUMPI. The tasks that GUMPI manages are as follows:

1. Generate and export a surface definition file
2. Convert the surface definition file to the input deck of the selected mesher
3. Execute the selected mesher
4. Import mesh into MSC/PATRAN

As each of the tasks were implemented, several issues inherent to MSC/PATRAN as well as the GUMPI system needed to be addressed.

### Exporting Geometry Information:

The first issue is transferring of surface data to external applications. The surface geometry information stored within a MSC/PATRAN database must be made available to the external mesher. One approach is to take advantage of the Neutral File 2.5 capability within MSC/PATRAN. Unfortunately, the Neutral File 2.5 format only supports bicubic patches. In general, this representation is not adequate for many applications. Often, geometry is brought in from a CAD(Computer-Aided Design) system into MSC/PATRAN via an IGES(Initial Graphics Exchange Standard) file with NURBS (Non Uniform Rational B-Spline) representation and it is desirable to maintain this representation. Also, one must consider how the external surface mesher represents the surfaces. In this case, it was decided to represent the surfaces by an ordered mesh of points. The required data box is illustrated in Figure 2, item 1. If the surfaces are highly curved, a dense mesh is required. Another issue is obtaining the coordinate locations of points on the surface. Initially, surface curves were generated on each of the surfaces. Then, each curve was decomposed by uniformly distributing points. This method was rather slow and required that the edges of adjoining surfaces be treated as special cases. However, PCL has a surface evaluator function which greatly improved the routine.

### Creating the Input for the External Mesher:

This task is completed by an external ANSI-C code. This code takes as input the surface definition, a map file and a flag indicating which external mesher was selected. As previously discussed, the surface is defined by a structured mesh of points. The map file contains a geometric tolerance and some boundary condition flags for future use. The code converts the surface definition into the appropriate input deck. The code automatically performs several tasks to generate the input deck. The code generates the required patch definitions by eliminating the duplicate points and surface edge curves from within the database. The functions that construct and manipulate the patches are based on routines found in a code called GridTool[3]. The geometric tolerance is used by the code to determine if two points are coincident. The required data box is shown in Figure 2, item 3. The main advantage of using ANSI-C over PCL is that data structures can be used efficiently. Most unstructured surface meshers required similar data as input. Consequently, a general data structure was setup that can be used for various unstructured meshers. An additional advantage is that a code was developed that can generate

input for various unstructured meshers within the user's analysis process and can be used outside the MSC/PATRAN environment.

#### Executing the Selected External Mesher:

The key issue for this task is to develop a method to execute the selected external mesher that is generic enough to handle any future mesher that can be added to the system. Since PCL has the capability of spawning UNIX processes easily, it was decided that a UNIX script could be written for each supported external mesher. The appropriate script along with the necessary parameters will be executed by GUMPI. Specifically, the user only needs to select the required mesher from the pull-down menu as shown in Figure 2, item 6.

#### Importing the Mesh in MSC/PATRAN:

The main issue to consider is the file format the external mesher stores the newly generated mesh. It was decided that converting the mesh to a Neutral File 2.5 format was the easiest method to implement since file import function is already built into PCL. The main disadvantage to this method is that the association between an element and surface is not preserved. However, since MSC/PATRAN has the element to surface association command, the element-surface association can be obtained for most cases.

#### **MINIMIZING USER INTERACTIONS:**

Minimizing the user interactions with the management of the external applications is a very important function. Several tasks are accomplished by GUMPI automatically that is completely transparent to the user within the auto-exec mode(see Figure 2, item 5). The key item that accomplishes this function is the project name that is defined in the menu form. The project name is used to generate a unique filename for each application within a particular mesh generation process. In this way, the filename of a file that is generated by a particular application is known and can be accessed by another application later in mesh generation process. The required data box is shown in Figure 2, item 3. Another advantage of minimizing the user interactions, is that GUMPI can take advantage of MSC/PATRAN batch mode. A script has been written for batch mode operations. In this way, the mesh generation and MSC/PATRAN can be placed with in a design optimization loop. Also, model containing large number of surfaces can be generated automatically. Another item that integrates GUMPI within MSC/PATRAN is the used of the PCL error message functions. Each application after it is launched by the system, returns a success flag. If unsuccessful, a form describing the problem is automatically created. This creates a nearly seamless environment between the external applications and MSC/PATRAN.

### **CONCLUSION**

Interfacing external unstructured meshers to MSC/PATRAN can be accomplished fairly easily. The issues involved are addressed fairly easily using the routines already built into the MSC/PATRAN system. The additional capabilities offered by other meshing algorithms can be added to the MSC/PATRAN environment. GUMPI provides such an interface so that these meshers can be accessed and used as easily as the built-in meshing algorithms already within MSC/PATRAN.

## ACKNOWLEDGMENTS

GUMPI was developed in the Geometry Laboratory (GEOLAB) at the NASA Langley Research Center. The authors wish to thank Kim Bey of Thermal Structures Branch of the Structures Division for initiating the requirements for the interface. For further information on GUMPI see the web site at <http://geolab5.larc.nasa.gov/GUMPI>. For further information on the supported unstructured mesher FELISA, send e-mail to [felisa@larc.nasa.gov](mailto:felisa@larc.nasa.gov). For further information on the Vgrid unstructured mesher, contact Neal Frink at [n.t.frink@larc.nasa.gov](mailto:n.t.frink@larc.nasa.gov).

## REFERENCES

1. Peiró, J., Peraire, J., and Morgan, K., *FELISA SYSTEM Version 1.1 Reference Manual*, January 1995.
2. Parikh, P., Pirzadeh, S., and Löhner, T., "A Package for 3-D Unstructured Grid Generation Finite Element Flow Solution and Flow Field Visualization", *NASA CR-182090*, September 1990.
3. Samareh, Jamshid, "GridTool: A Surface Modeling and Grid Generation Tool," Proceedings of the Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions, *NASA CP-3291*, May 9-11, 1995.

FIGURES

# Geolab Unstructured Mesher / Patran Interface

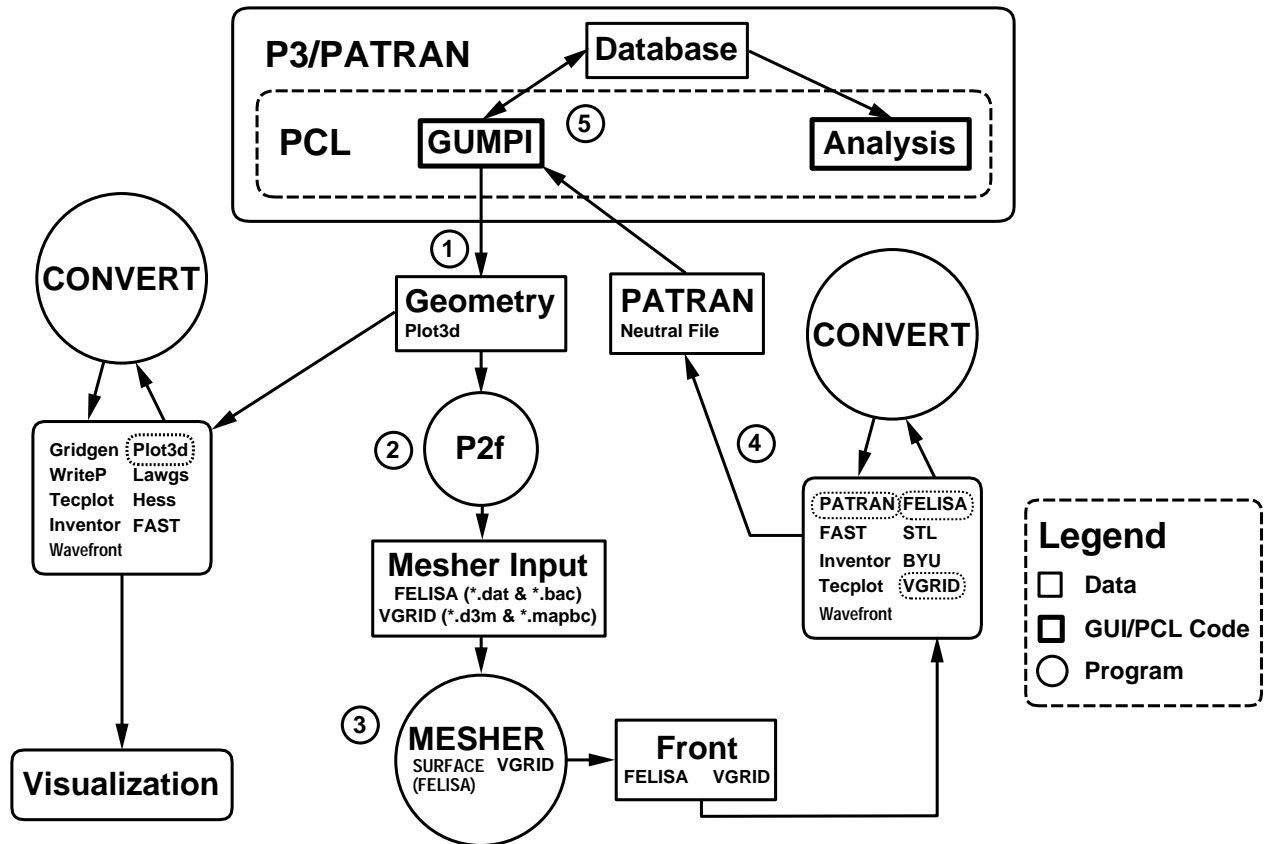


Figure 1: GUMPI System Process Diagram

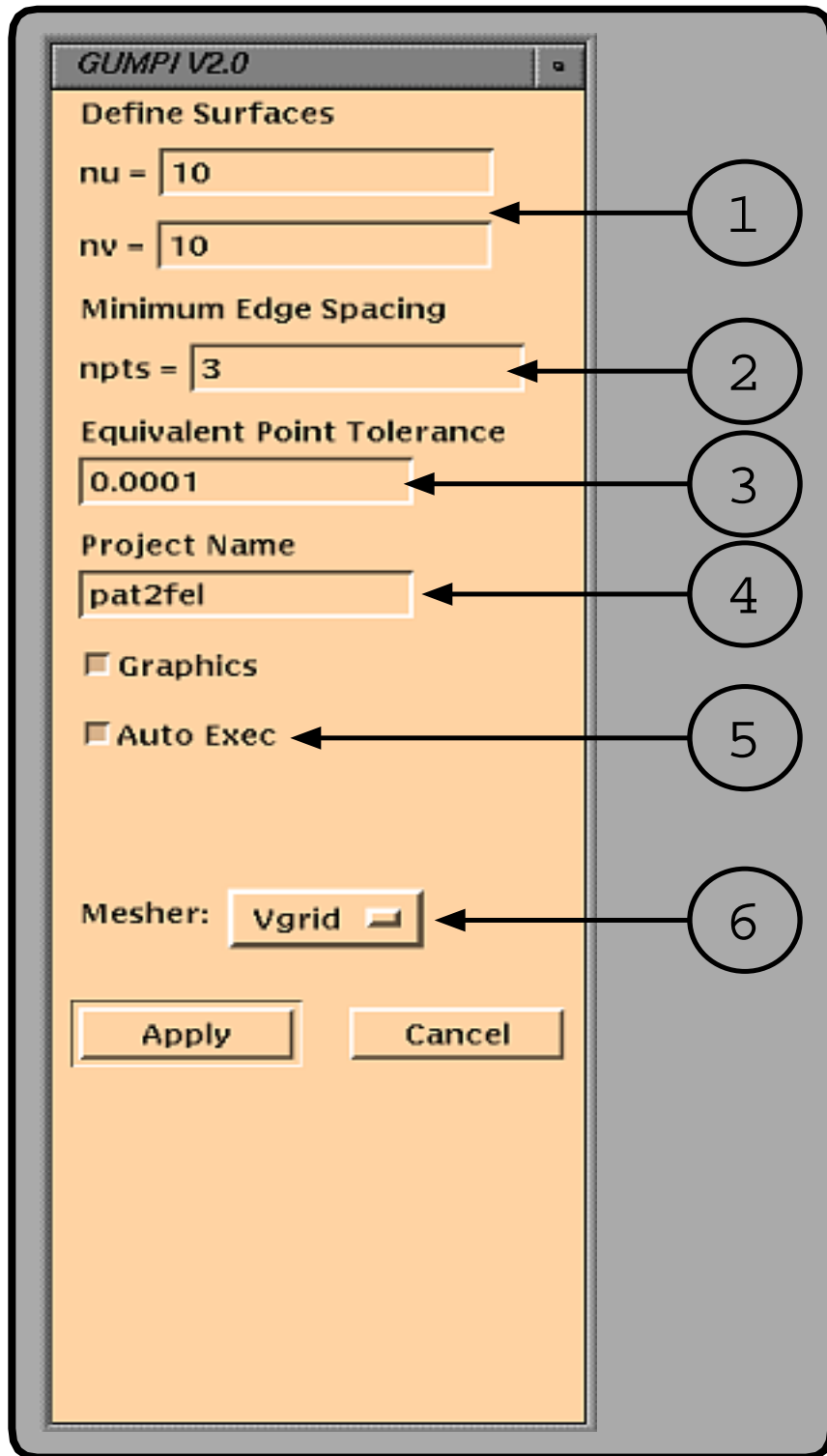


Figure 2: GUMPI Input Form