

Handling Tessellated Free Shape Objects with a Morphing Mesh Procedure in Comsol Multiphysics®

P. Franciosa¹, S. Gerbino^{*2}

¹University of Naples Federico II, School of Engineering

²University of Molise, School of Engineering

*Corresponding author: Via Duca degli Abruzzi - 86039 Termoli (CB), Italy
salvatore.gerbino@unimol.it

Abstract: Tessellated models are more and more used in several engineering fields. The need to use such models to quickly perform computer simulations, related to coupled physical phenomena, implies the use of dedicated software, allowing to solve, into an integrated environment, multiphysics problems. In the present work, Comsol Multiphysics® 3.5a has been used and its ability to handle tessellated models has been highlighted. Some its limits have been pointed out and to overcome them, a general procedure, fully-implemented in MatLAB®, has been proposed. Finally, two case studies have been analyzed showing how the proposed MatLAB-based tool, linked to Comsol Multiphysics®, works.

Keywords: Tessellated model, Morphing Mesh Procedure, Free shape modeling, MatLAB® integration.

1. Introduction

This work is partially based on and inspired to the PUODARSI (*Product User-Oriented Development based on Augmented Reality and Interactive Simulation*) Italian research project (PRIN), aimed to implement a computer tool to quickly perform multiphysics simulations (starting from real objects, digitalized by means of reverse engineering technologies), visualize results in stereoscopic mode, modify the shape of the object interactively, and, then, update simulation results. This re-design loop is stopped when a satisfactory solution is reached according to the design intent.

The need to analyze multiphysics phenomena implies using of specific software. Comsol Multiphysics® 3.5a was then adopted since it offers a friendly interface and a direct link to MatLAB®, where automatic procedures may be implemented.

Generally speaking, tessellated models are widely used in different engineering fields, such as FEA, computer graphics, gaming, reverse en-

gineering applications, rapid prototyping, 3D image reconstruction.

To improve the flexibility in managing such formats, Comsol Multiphysics® introduced an interesting feature: the new interface with Simpleware® commercial software allows user to directly import 3D mesh model, based on high resolution MRI scan images. These models can be processed in multiphysics simulations [1].

However, we have experimented a weakness in Comsol Multiphysics® when tessellated models, coming from any pre-processing (image reconstruction, digital acquisition, mesh for rendering applications), are imported. Starting from such models, Comsol Multiphysics® tries to create rational Bezier patches, by using internal algorithm mainly based on segmentation and feature recognition approaches [2]. However, all this procedure may generate some errors that user cannot directly control. To fix these bugs, CAD tools available in Comsol Multiphysics® may be used, but without guarantee to reach a good model, able to be processed during the next operations (such as meshing phase and numerical simulation). However, the so-created geometry cannot be easily edited. So, boolean operations, which require geometric decomposition analyses, are not allowed. On the other hand, user could create a B-Rep CAD model, starting from the tessellated one, by using specific commercial software, such as Geomagic Studio®, Rapid-Form®, Rhinoceros® and many others. An example of such a procedure is in [3] where a 3D tessellated model, coming from a 2D image reconstruction, was firstly converted into a 3D CAD model by using internal functions available in Solid Works® CAD system. Then, the so-created CAD model was imported in Comsol Multiphysics® where stress-strain simulations were performed.

However, making usable 3D CAD models, starting from a tessellated geometry, may become a very large time consuming task and it is in contrast with the need to reach a fast solution,

especially when geometry changing and editing are often required.

To meet the need to have an environment allowing to do numerical simulations on tessellated models, and interactively edit such a geometry, a tool, called *ProMESH* – implemented in MatLAB® and linked to Comsol Multiphysics® 3.5a running in background mode – is here described. Further details about numerical algorithms can be found in [4].

The *ProMESH* architecture has been here improved and enriched with a new feature, allowing user to interactively modify the geometry shape, based on the Morphing Mesh Procedure (MMP).

Generally speaking, *ProMESH* allows: (I) loading any tessellated model in *.stl* format; (II) editing the imported geometry (applying the MMP or creating a 3D closed domain starting from an opened shell model); (III) exporting that geometry in Comsol Multiphysics®; and solving a steady fluid dynamic simulation in automatic way. In particular, external flows around free shape objects are accounted and simulated. A fast preview of results is also available in *ProMESH*. For a more accurate post-processing, user can export result data, as structured data array, in Comsol Multiphysics®.

In a preliminary design stage, where the geometry has to be analyzed for different configurations, after modifying it interactively, *ProMESH* tool gives a fast answer to the designer and helps him to understand the physical behavior of the object being analyzed. Obviously, accuracy of numerical simulation results is not the main goal during this phase.

The paper is arranged as follows: Section 2 summarizes the key features of the *ProMESH* tool; Section 3 describes two case studies; finally, Section 4 draws final remarks.

2. *ProMESH* architecture

2.1 Thickening tessellated model

Tessellated model, especially when coming from digitalization processes, may be opened. In other words, the model does not envelope a closed 3D domain. For those geometries a special feature to tick shell models has been implemented into *ProMESH*. Further mathematical details are in [4].

Generally speaking, any tessellated model may be described by means of vertices and planar triangular faces connecting the same vertices (V matrix and F matrix can be introduced for vertices and faces, respectively). An opened tessellated model is thickened into two consecutive steps. In the first one, each vertex is moved along the relative local normal (offset procedure). The local normal for a given vertex is here calculated by using the weighted mean method, as in [5]. After offsetting mesh, closure triangles along boundary edges are introduced to obtain a closed 3D domain.

It can be pointed out that this procedure works well for small thickness values. No intersection test of triangles is actually implemented. This may become necessary for high curvature models.

2.2 Converting tessellated model

The whole tessellated model (either opened or closed) is converted in a Comsol Multiphysics®'s object as union of 3D planar faces. Firstly, each triangular face of the tessellated model is transformed into a trimmed planar triangular patch by using the “*face3*” function [2]. “*Geomcoerce*” function [2] is then applied to generate a whole single model.

The basic MatLAB® pseudo-code to convert a tessellated model into a Comsol Multiphysics®'s object is given below.

```
function sldObj=crObj(Vertex,Face)

% -- INPUT:
% --Vertex: vertices matrix.
% --Face: faces matrix.

% -- OUTPUT:
% --sldObj: comsol's object.

% -- start code...
nFace = size(Face,1);

% --initialize loop variable...
fObj = cell(1,nFace);

% --begin loop...
for I=1:nFace
    f = Face(I,:);
    lcV = Vertex(f,:);
    fObj{I} = face3(lcV(:,1),...
                  lcV(:,2),...
                  lcV(:,3));
end

% --create comsol's object
sldObj = geomcoerce('solid',fObj);
```

2.3 Morphing Mesh Procedure

Morphing mesh is a well known technique used in computer graphic applications as a powerful tool for free-shape modeling and designing ([6] and [7]). In [8] MMP was developed to generate variational parts to be used in statistical assembly processes. Starting from this methodology, in the present work an extended procedure has been implemented.

MMP may be summarized as follows. User defines a set of control points on the model, by picking them from the graphical interface. Then, the relative influence radius is assigned for each point. Control points directly influence final shape of the deformed object, and this shape can be fine-tuned by adjusting the influence radius or the position of each control point.

Let P_c be the matrix of control points. R_i is the influence radius related to the i -th control point. The influence radius defines the 3D region within which any vertex is influenced by the related control point (influence hull).

The displacement, ΔV_j , of any vertex V_j of the tessellated model is calculated as a weighted mean of all control points P_c , as stated in (1).

$$\Delta V_j = W(V_j) \cdot M \quad (1)$$

where W and M are the weighted and the morphing matrix, respectively.

The i -th element of the weighted matrix depends on: (I) position of the point V_j ; (II) position of control point P_{ci} and (III) its influence radius R_i , as stated in (2), where f is the weight function.

$$\begin{aligned} W_i(V_j) &= f(V_j, P_{ci}, R_i) = \dots \\ &= f\left(\frac{\|V_j - P_{ci}\|}{R_i}\right) = f(d) \end{aligned} \quad (2)$$

In [8] f was assumed as a polynomial function. In the present work, f is assumed as a piecewise Bezier curve (see Figure 1), which can be generated by applying the de Casteljau algorithm [9]. Function f is equal to 1 when V_j and P_{ci} are coincident, and tends toward zero for points V_j whose distance from P_{ci} is greater than zero. To assure smoothing shape, function f has zero slopes at end points. User can interactively modify the shape of f function by moving control

points of each Bezier curve. In Section 3 some examples will show how this procedure works.

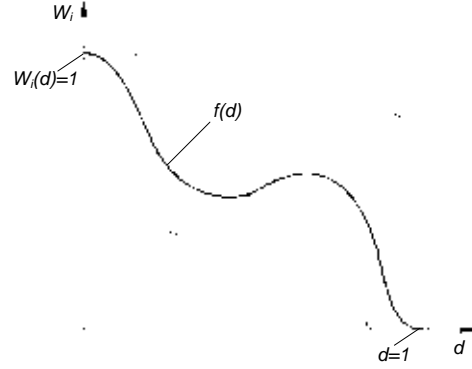


Figure 1. Definition of the weight function, f

Morphing matrix, M , is initially unknown and can be calculated by applying equation (1) to all control points. Then, it can be written:

$$\begin{cases} \Delta P_{c1} = W(P_{c1}) \cdot M \\ \Delta P_{c2} = W(P_{c2}) \cdot M \\ \dots \\ \Delta P_{cr} = W(P_{cr}) \cdot M \end{cases} \rightarrow \Delta P_c = W(P_c) \cdot M \quad (3)$$

where ΔP_c is the matrix of displacement assigned at control points. These displacements are calculated when user interactively picks and moves any control point from the graphical interface.

Then, M matrix can be calculated by solving the following linear system:

$$M = W(P_c)^{-1} \cdot \Delta P_c \quad (4)$$

It can be highlighted that matrix $W(P_c)$ is singular if two control points are coincident each other. In these cases MMP cannot be applied.

Moreover, MMP does not assure “well-shaped” triangles. In fact, when stretching and morphing model, low quality triangles may be generated. Since such a model is then directly imported in Comsol Multiphysics® as described in Section 2.2, it is hopeful to have a high quality model, which may be easily meshed with 3D elements (tetrahedral elements).

At the present, no optimization shape algorithm is implemented, but a Laplacian smoothing or a constrained shape optimization may be integrated as described in [10] and [11].

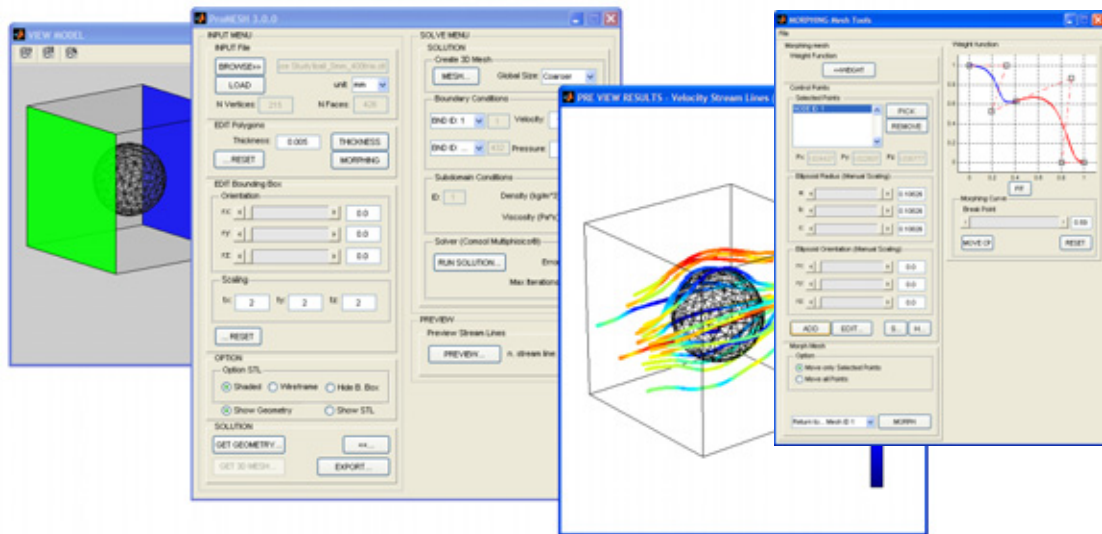


Figure 2. *ProMESH* Graphical User Interface

2.3 Graphical User Interface

Figure 2 shows the *ProMESH*'s MatLAB-based Graphical User Interface (GUI).

GUI drives user to load any tessellated model (either opened or closed) into *.stl* format and visualize it (*VIEW MODEL* window).

Since the aim is to simulate and calculate external flows around object, *ProMESH* automatically calculates its bounding box as described in [4]. In a CFD analysis bounding box is the computational domain used to solve fluid dynamic PDE problems. User may modify its sizes and orientation through the *EDIT Bounding Box* sub-menu.

Open imported tessellated model may be closed by applying the thickness procedure summarized in Section 2.1 (*THICKNESS* button).

User may interactively modify geometry shape through the *MORPHING* menu. GUI allows to pick, from the *VIEW MODEL* window any control point belonging to the geometry and set the relative influence hull, controlling its sizes and orientation.

At the present, influence hull is assumed as an ellipsoid domain (its sizes are parameterized by means of its three radiuses). Morphed shape may be easily tuned and controlled modifying the *Weight* function. User can change control points of the piece-wise Bezier curves by acting on the graphical interface.

Once the tessellated model is ready to be post-processed, *GET GEOMETRY* button creates the Comsol Multiphysics® geometry object and automatically generates a 3D mesh with tetrahedral elements.

After defining physical properties of fluid and assigning boundary conditions, *RUN SOLUTION* button solves fluid dynamic PDEs. User has to set the maximum number of iterations and the global tolerance error, required by the non-linear solver (PARDISO solver is set as default).

Finally, *PREVIEW* button allows to post-process simulation results and visualize the chosen number of stream lines. A more accurate post-processing may be achieved by exporting result data into Comsol Multiphysics® (*EXPORT* button).

By using *ProMESH* interface, designer can quickly analyze different shape designs and understand which is the best one according to his design intent. The final shape then can be obviously used in more accurate computer simulations.

3. Applications

Below, two case studies are illustrated. In the first one the imported tessellated model is a closed sphere (Figure 3). An opened tessellated geometry, instead, is used in the second application (Figure 4).

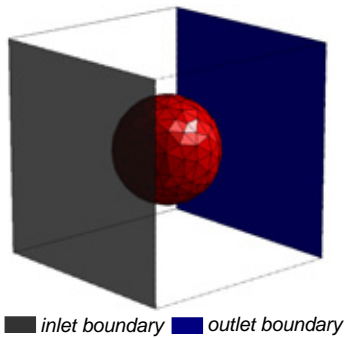


Figure 3. Initial geometry for first application

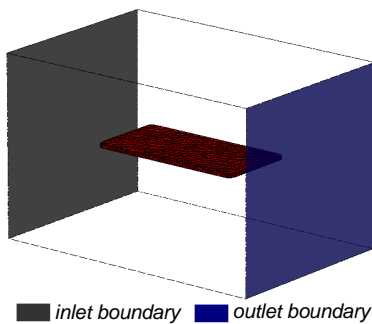


Figure 4. Initial geometry for second application

The aim is to show how *ProMESH* allows to handle tessellated model, modifying its shape by means of MMP, and to run fluid dynamic simulation.

Fluid flow was described by means of Navier-Stokes equations. Fluid density and dynamic viscosity were assumed as 1.19 kg/m^3 and $1.85 \cdot 10^{-5} \text{ Pa}\cdot\text{s}$, respectively.

Moreover, a linear shape function was adopted.

3.1 Case 1: starting from closed sphere

Figure 3 shows the imported tessellated model (161 vertices and 318 triangles). Sphere is approximately 100mm.

A constant velocity of 16.0 m/s was assigned to the inlet boundary, while a zero pressure on the outlet boundary (see Figure 3). All other boundaries were constrained with no-slip conditions.

Figure 5 depicts the velocity field for three different geometry shapes: second and third shapes were derived from the imported tessellated model by applying MMP.

3.2 Case 2: starting from planar surface

In this application the imported geometry is a planar tessellated surface of approximately $200 \times 100 \text{ mm}$. Firstly, the model was thickened of 5mm by applying the procedure summarized into Section 2.1. Then, different free shape geometries were created. Fluid dynamic simulations were performed by assigning an inlet velocity of 1.0 m/s and a zero output pressure (see Figure 4). Figure 6 depicts flow fields for three morphed shapes.

4. Final Remarks

The present work focused on a MatLAB-based tool, called *ProMESH*, which allows to quickly accomplish fluid dynamic simulations by using Comsol Multiphysics® in background mode, starting from any 3D tessellated model.

The implementation of *ProMESH* was mainly motivated by the weakness of Comsol Multiphysics® to well manage imported tessellated models. Moreover, in order to have a friendly GUI, allowing to interactively and intuitively modify geometry shapes, a Morphing Mesh Procedure was implemented.

ProMESH's algorithms are very efficient only when handling tessellated models with a relatively small number of triangles (less than 10.000). A decimation algorithm could be added to make easier working on imported big models.

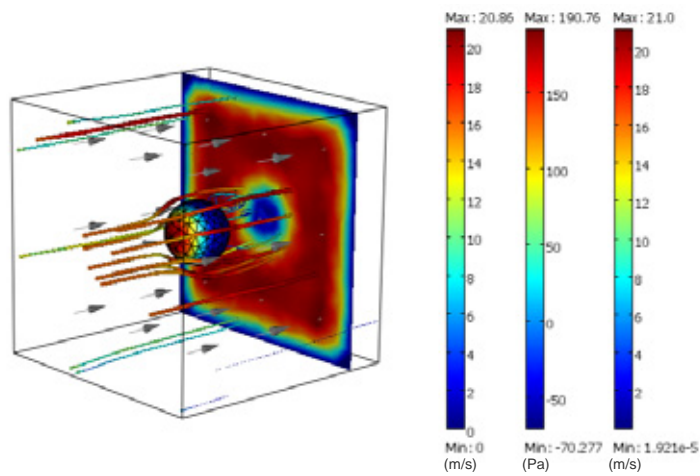


Figure 5.a. simulation results for the imported geometry

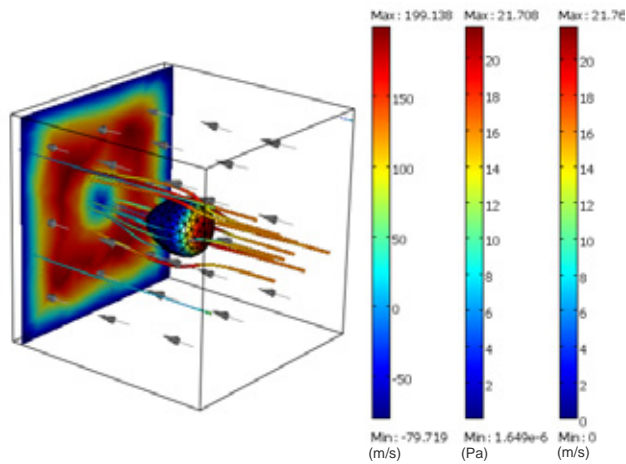


Figure 5.b. simulation results for the morphed geometry-first shape

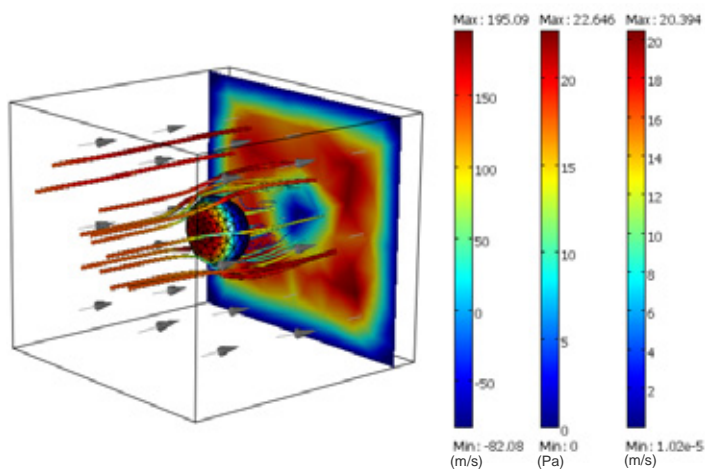


Figure 5.c. simulation results for the morphed geometry-second shape
(Slice: Velocity field. Boundary: Pressure. Stream line: Velocity field)

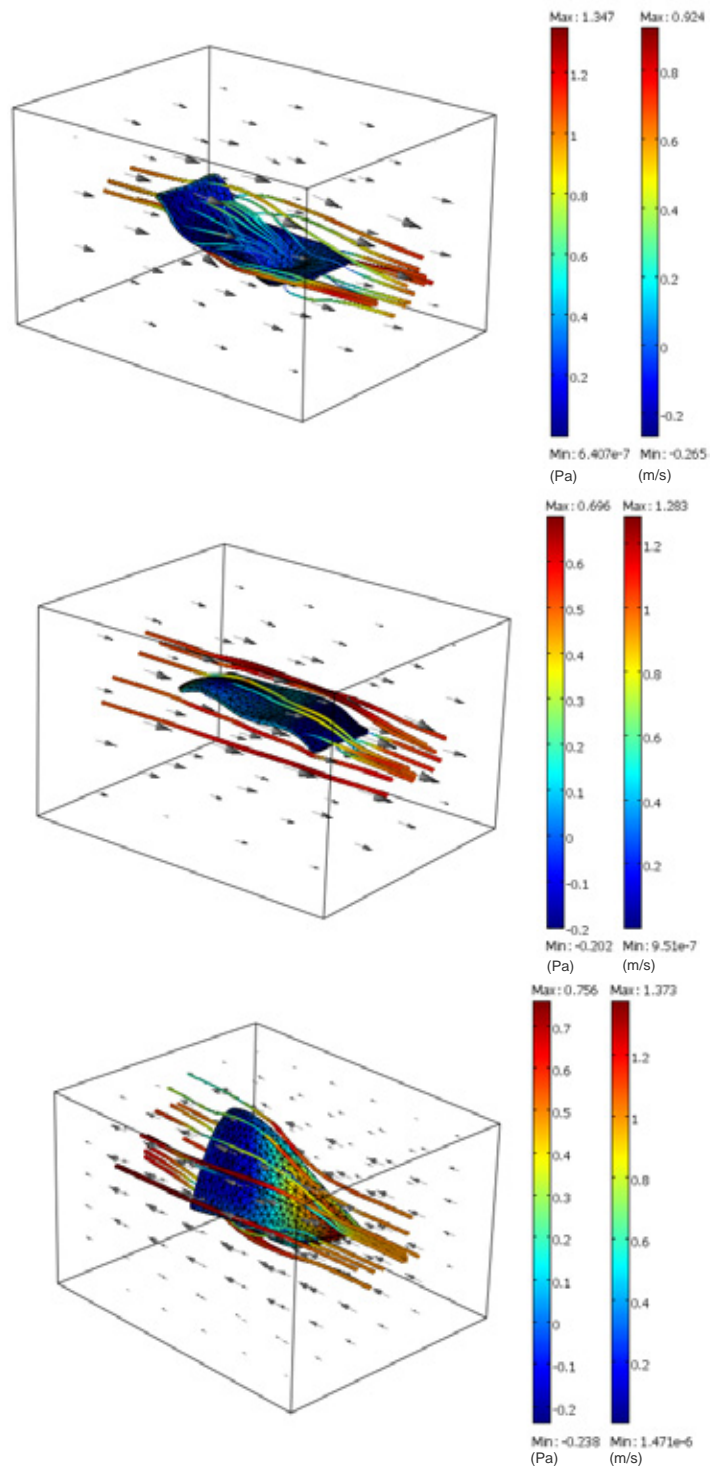


Figure 6. simulation results for three different morphed shapes
(Boundary: Pressure. Stream line: Velocity field)

5. References

1. Said R., Cotton R., Young P., Datta A., El-wassif M., Bikson M., Image Based-Mesh Generation for Realistic Simulation of the Transcranial Current Stimulation, *Proc. of COMSOL Conference 2008*, Boston (2008)
2. Comsol Multiphysics 3.5a®, User's Manual 2009, COMSOL AB
3. Franciosa, P., Martorelli, M., Ausiello, P., 3D Visualization and Fatigue Simulation in Restored Human Teeth by Using Micro-CT Data, *Proc. of the Int. ADM-INGEGRAF*, Lugo (Spain), June 10-12 (2009)
4. Di Gironimo G., Franciosa P., Gerbino S., An RE-CAE Methodology for Re-Designing Free Shape Objects Interactively, *Int. Journal on Interactive Design and Manufacturing*, DOI 10.1007/s12008-009-0082-8 (2009)
5. Lengyel E., *Mathematics for 3D Game Programming and Computer Graphics*, Charles River Media, Second Edition (2003)
6. Borrel P., Rappoport A., Simple Constrained Deformations for Geometric Modelling and Interactive Design, *ACM Trans Graph* **Vol. 13**, pp. 137-155 (1994)
7. Raffin R., Neveu M., Jaar F., Curvilinear Displacement of Free-Form-based Deformation, *The Visual Computer* **Vol. 16**, pp. 38-46 (2000)
8. Franciosa P., Gerbino S., Patalano S., Simulation of Variational Compliant Assemblies with Shape Errors Based on Morphing Mesh Approach, submitted to *TMCE 2010*, Ancona (Italy), April 12-16 (2010)
9. Anand B. V., *Computer Graphics and Geometric Modelling for Engineers*, Clemson University (1992)
10. Knupp P. M., Algebraic Mesh Quality Metrics, *Int. Journal Science Computing*, **Vol. 23**, pp. 193-218 (2001)
11. Canann S. A., Tristano J. R., Staten M. L., An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes, *Int. Meshing Roundtable* (1998)