

Hexahedral Finite Elements Mesh Generation Method with Applications to Plastics Parts

STEFAN TABACU^{1*}, ANTON HADĂR², NICOLAE DORU STĂNESCU¹, SORIN ILIE¹, DANIELA IOANA TUDOR²

¹University of Pitești, Mechanics and Technology, 1 Targu din Vale Str., 110040, Pitești, Romania

²University Politehnica of Bucharest, IMST, 313 Splaiul Independenței, 060042, Bucharest, Romania

Most of the problems of modern engineering are solved using numerical methods. In order to use a numerical method the first step that must be performed is to generate a numerical model that contains all the input data in terms of geometry, material data, boundary conditions (loads and support). Ideally a structure must be defined only using hexahedral elements that are capable to provide the best results. This paper presents a method for generation a fully hexahedral finite elements model using a grid based procedure. The method is a good support for the engineering involved in product design. An application to a plastic manufactured part is presented. The results obtained using the all hexahedral finite elements models are compared with the results obtained using a numerical model generated using conventional techniques.

Keywords: numerical model, hexahedral finite elements, mesh generation method

Most of the problems of modern engineering are solved using numerical methods. In order to use a numerical method the first step that must be performed is to generate a numerical model that contains all the input data in terms of geometry, material data, boundary conditions (loads and support).

The results provided using a numerical method are in most of the cases influenced by the numerical model. This is especially the case of finite elements method [1].

Although the material data, boundary conditions, parts' interferences may be correctly specified by the user the results may be influenced by the definition and quality of the finite elements model. There are very strict requirements regarding the shape and type of finite elements used to build the model for simulations (applications of the numerical methods).

Ideally a structure must be defined only using hexahedral elements that are capable to provide the best results and there are commercial software packages that can provide these numerical models.

There are many cases when the geometrical model does not provide enough support for the user to generate an all hexahedral finite elements model. In other cases the geometrical model is very complex and the time required for the generation of the finite elements model using conventional techniques can be time consuming. If the design process is not completed and preliminary analyses are required then the time cost increases even more. This is the case of parts manufactured using plastic, thermoplastics or other similar materials. For the engineers involved in process of analyzing this kind of parts a useful method to generate the finite elements model is presented.

This paper presents a method for generation a fully hexahedral finite elements model using a grid based procedure [2-9]. One of the major advantages of the present method is that the numerical model can be build using a user specified element size required in order to decrease the computational resources in case of time explicit applications. On the other hand the method is quite simple and it can be easily implemented or further developed.

Hexahedral finite elements mesh generation method

The method used to generate the all hexahedral finite elements model of the structure consists in a number of procedures used for reading the geometrical model, definition of the intersection planes and intersection contours, definition of the contour grid based quadrilateral mesh, definition of the section mesh, definition of the hexahedral finite elements and export of the numerical model [2, 5, 7] useful for parts with complex geometrical model like components manufactured using plastic and other similar materials.

These procedures were implemented in a custom written code using Matlab [10].

The structure of the program is presented in figure 1.

Input of the geometry

The geometrical model in this file format is converted (using ParaView) in PLY (polygon) file format using an ASCII format is selected for output.

A PLY file consists of a header followed by a list of vertices and then a list of polygons. The header specifies how many vertices and polygons are in the file, and also states what properties are associated with each vertex, such as (x, y, z) coordinates, normals and colour. The polygon faces are simply lists of indices into the vertex list, and each face begins with a count of the number of elements in each list. The vertices are identified and the patches are constructed.

Using the coordinates of the vertices, the dimensions of the model can be computed

$$\begin{aligned}x_{\max} &= \max(x_i); \quad y_{\max} = \max(y_i); \quad z_{\max} = \max(z_i); \\x_{\min} &= \min(x_i); \quad y_{\min} = \min(y_i); \quad z_{\min} = \min(z_i)\end{aligned}\quad (1)$$

and also the geometrical centre of the model is identified.

$$x_c = \frac{\sum_{i=1}^n x_i}{n}; \quad y_c = \frac{\sum_{i=1}^n y_i}{n}; \quad z_c = \frac{\sum_{i=1}^n z_i}{n}\quad (2)$$

A reference plane is constructed through this point. All the operations are performed with respect to the global orthogonal coordinate system. Thus, in some cases, there

* Tel.: 0722983592

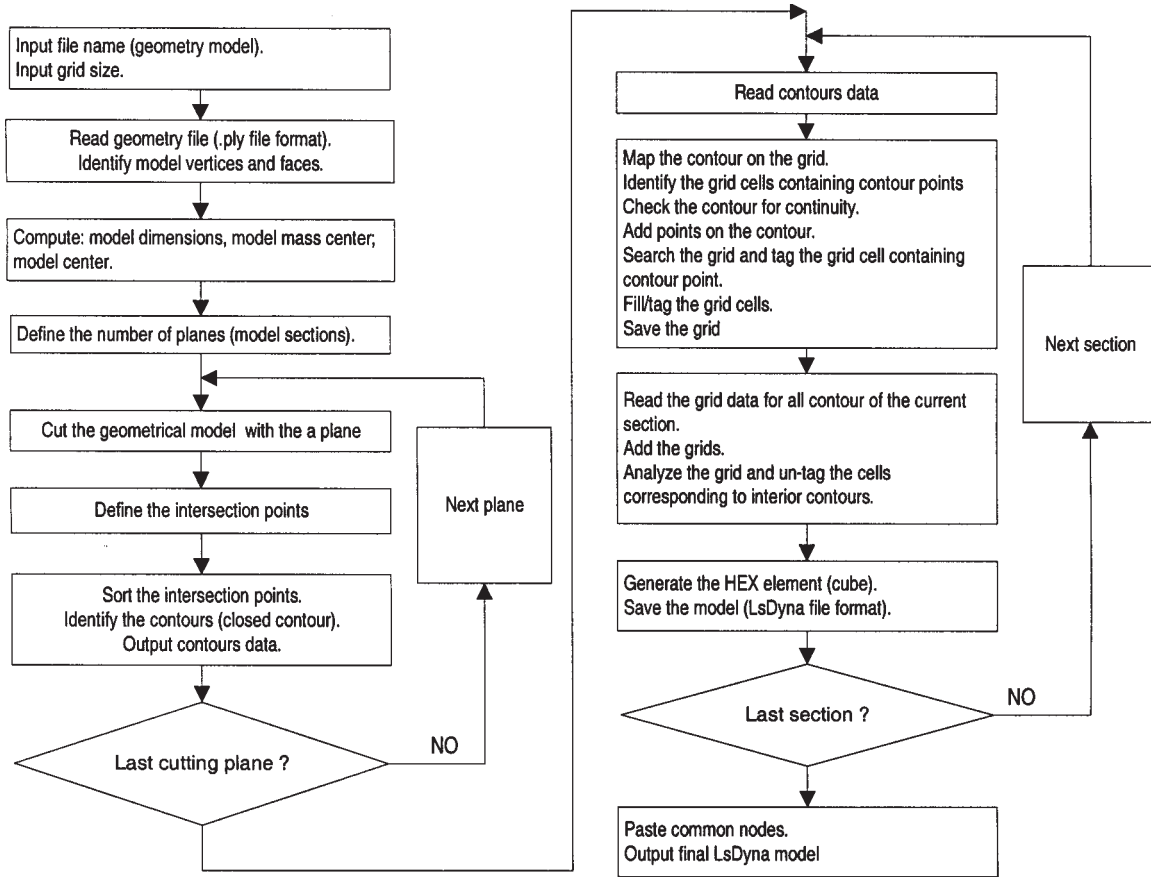


Fig. 1. Procedures used to generate the hexahedral finite elements model

may be necessary to align the part in order to obtain the best results because the cutting planes are parallel with the global OXY reference and these planes are equally spaced along OZ axis.

Definition of the cutting sections

The user can specify the size of the finite element thus the number of cutting planes (equation 4) that depends on the model dimensions and the specified element size.

$$z_{size} = \max(z_i) - \min(z_i) \quad (3)$$

$$N_{cutting\ planes} = \frac{\max(z_i) - \min(z_i)}{element\ size} \quad (4)$$

The cutting plane is defined using three points. The first point has the coordinates $\mathbf{p}_0 = (x_c, y_c, z)$ where z is measured along OZ axis that is defined by z_c and the number of the current cutting plane. The second point has the coordinates $\mathbf{p}_1 = (x_c + x_{size}/2, y_c, z)$ and the third point has the following coordinates $\mathbf{p}_2 = (x_c, y_c/2, z)$.

The geometrical mode is composed of triangular patches. Using the vertices defining these patches three lines can be constructed.

Only the lines that are intersecting the cutting plane are identified and used to find the intersection points.

The parametric equation of the intersection between a line and a plane is used in order to define the contour cut by the plane.

The equation of a line passing through two vertices $\mathbf{v}_i = (x_i, y_i, z_i)$ and $\mathbf{v}_j = (x_j, y_j, z_j)$ of a face defined in the model is

$$\mathbf{v}_i + (\mathbf{v}_j - \mathbf{v}_i) \cdot t = 0; t \in \mathbb{R} \quad (5)$$

The parametric equation of the plane is:

$$\mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0) \cdot u + (\mathbf{p}_2 - \mathbf{p}_0) \cdot v = 0; u, v \in \mathbb{R} \quad (6)$$

The point at which the line intersects the plane is therefore described by setting the line equal to the plane in the parametric equation:

$$\mathbf{v}_i + (\mathbf{v}_j - \mathbf{v}_i) \cdot t = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0) \cdot u + (\mathbf{p}_2 - \mathbf{p}_0) \cdot v \quad (7)$$

which can be expressed in matrix form as:

$$\begin{bmatrix} x_i - x_c \\ y_i - y_c \\ z_i - z \end{bmatrix} = \begin{bmatrix} x_i - x_j & x_c + x_{size}/2 - x_c & x_c - x_c \\ y_i - y_j & y_c - y_c & y_c + y_{size}/2 - y_c \\ z_i - z_j & z - z & z - z \end{bmatrix} \cdot \begin{bmatrix} t \\ u \\ v \end{bmatrix} \quad (8)$$

Equation is solved and with respect to t from the solution obtained, the point of intersection is then equal to:

$$\mathbf{i}_i = \mathbf{v}_i + (\mathbf{v}_j - \mathbf{v}_i) \cdot t \quad (9)$$

Figure 2 presents the reference plane and the current cutting plane, as well as the intersection points that were computed.

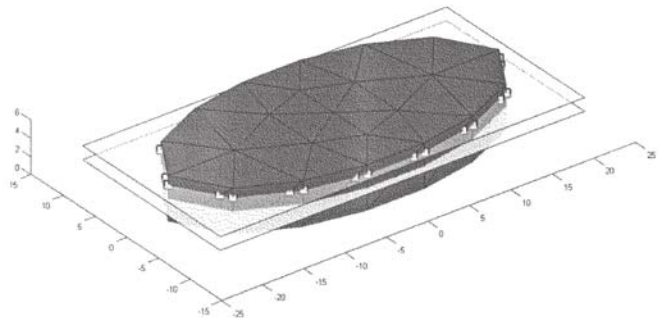


Fig. 2. Definition of the cutting plane and intersection points

From the total of intersection points identified for the position of the current cutting plane a number of ordered lists are created and closed contours are identified.

A given section can have a number of contours. Also, for each triangular patch there are two vertices that are shared with the neighbors. Therefore, for each intersection line (that as mentioned before, may belong to two triangular patches) two coincident intersection points are identified. In the list with the intersection points the triangular patch used to generate it is stored. This way the intersection line that connects the intersection points for each triangular patch is defined. By now the intersection lines are not sorted because the intersection points were generated according to the identification number (or list entry) of the triangular patch.

Two elements are required to create a continuous contour. The first element is that the connections between the contour lines are represented by coincident intersection points. The second element is represented by the other intersection point that defines the intersection line. Figure 3 presents the method used to generate the continuous contour.

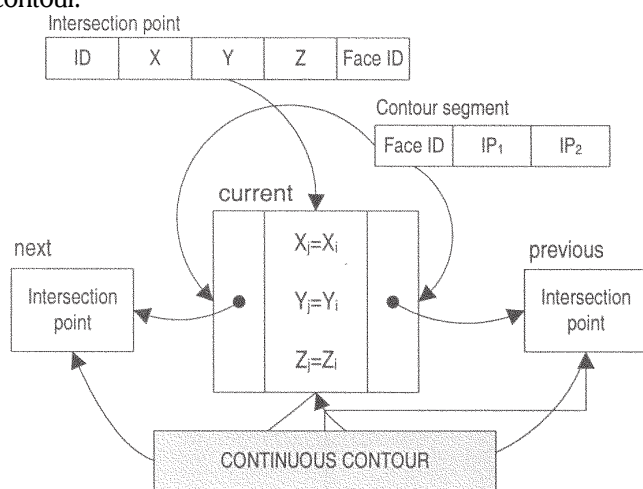


Fig. 3. Method for generating the continuous contour

If a contour is closed, the coordinates of the first point in the list must be the same with the coordinates of the last point in the list. Otherwise, if no points are added during a search cycle the contour is open. Figure 4 presents a structure intersected by a number of cutting planes while figure 5 presents the lines that are joining the intersection points before ordering them.

Once the succession of intersection points is determined (fig. 6a) the duplicates are removed from the list and the continuous closed contour is identified (fig. 6b).

The results are saved as a list of points for each individual contour. (e.g. for figure 6b a number of 5 files with the contour points will be created).

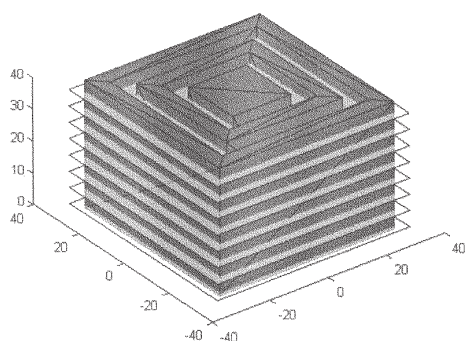


Fig. 4. Cutting planes

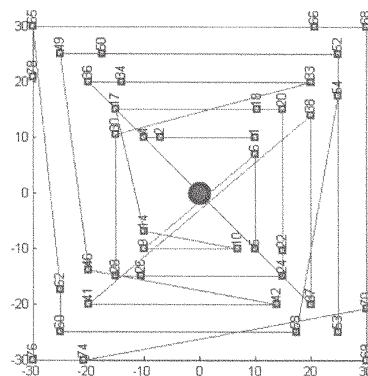
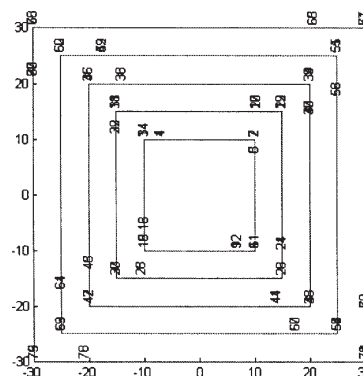


Fig. 5. Intersection points



a)

b)

Fig. 6. Contours definition

a) Continuous contour, b) Duplicate intersection points are removed

Definition of the grid based quad and hex elements

The hexahedral elements are generated from quadrilateral elements that are generated for each intersection plane. A grid with the dimensions defined by the model size and elements size is generated. Actually, the same grid will be used for each cutting plane. It is centered with respect to the model centre and the size is defined by the overall model dimensions.

The contour data are loaded and analyzed in order to generate the quadrilateral elements. The grid cells containing intersection points are marked. The method is simplified because once an intersection point is inside a grid cell the cell is selected. The position or number of intersection points that are inside a cell does not affect the selection of the cell.

As the contour is closed so the grid that contains the intersection points must be. In figure 7a it can be noticed that only a few cells will be selected. Therefore the number of points that are mapped on the grid must be supplemented. The spanning between the marked grid cells

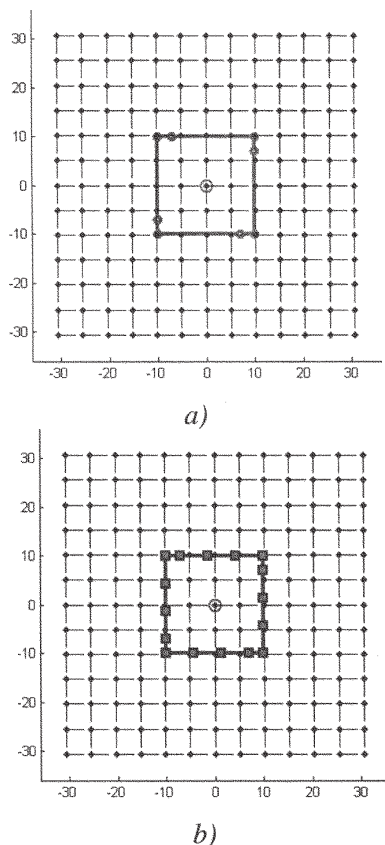


Fig. 7. Identification of the grid cells containing contour points
a) initial contour points; b) supplemented contour points

is analyzed and, if in horizontal or vertical direction there is a gap, of one or more cells, supplementary points are added (fig. 7b).

At the beginning, to all grid cells a 0 value is assigned. The cells containing intersection points will have the 1 value assigned and the grid map of the contour is defined. Using this map, the cells that are inside the contour are identified and these cells will have also 1 assigned. Each line containing grid cells is analyzed and the grid cells containing contour points that were previously identified are used to check where the cells should have a value of 1 assigned. The cells colored with black are the ones that contain contour points, while the grid cells colored in yellow are those defined by analyzing the grid lines.

One of the major advantages is that the same method can be used for all the contours defined and a procedure will be further presented to identify the voids or empty spaces in the model.

For the structure presented in figure 4 and the contours identified in figure 6b the grid maps corresponding to each contour are presented in figure 8.

The individual grid maps are saved and then added in order to obtain the final map of the current section (fig. 9). Odd number (1,3,5,...) means that a finite element will be generated for the corresponding grid cell and 1 will be assigned on the final grid map. Even number (2,4,6,...) means that an empty space exist for the corresponding grid cell and 0 will be assigned for the final grid map.

Using the grid map and the defined finite elements size the hexahedral finite elements are generated and saved in the output file. The resultant finite elements model and the associated final grid map are presented in figure 10.

The final finite elements model is generated using the same procedure. The novelty on the method resides in the addition procedure of the individual grid maps of the

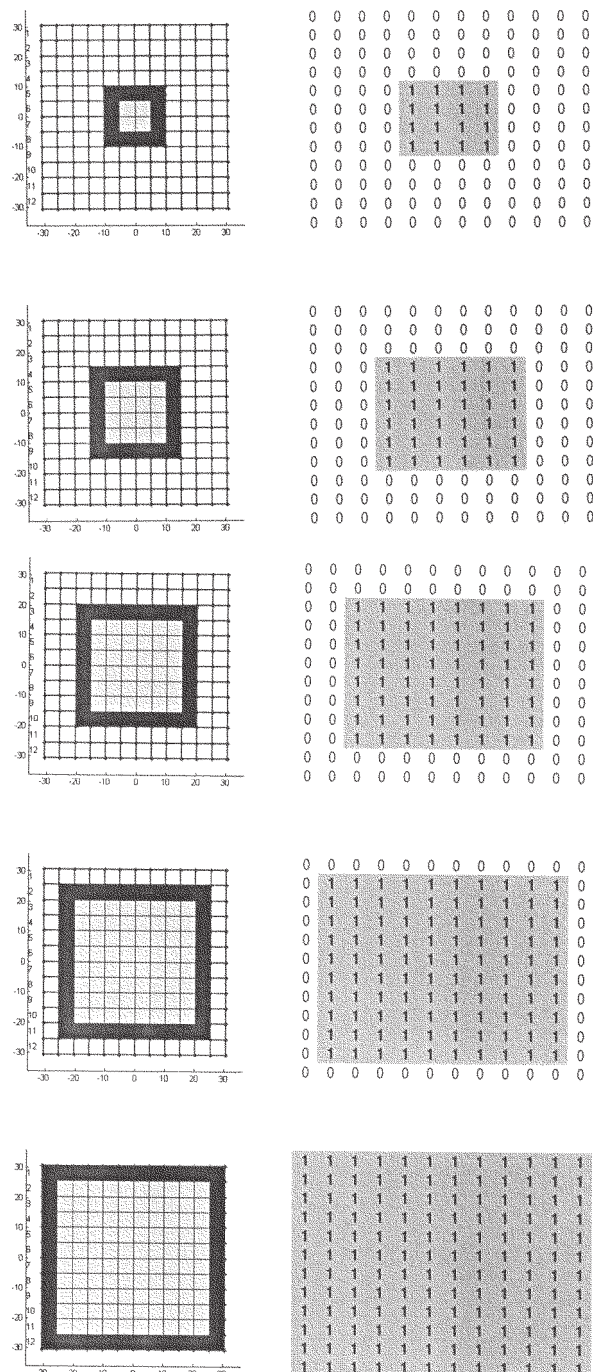


Fig. 8. Grid maps associated with the contours

contours identified on the current section. Therefore there are not required any specialized algorithms that must identify the relative position of the contours.

Applications

The mesh generation method is used to generate the numerical model of a plastic handle that is used for a vehicle's hood latch assembly. The handle is operated by hand in order to ensure the full release of the hood from the vehicle's engine compartment (fig. 11).

The numerical model of the plastic handle was generated using the hexahedral finite elements generation method (figure 12a) described above and using a commercial software (fig. 12b).

The method for generating the geometry based numerical model consists in defining the closed volume of the part and then meshing it using tetrahedral elements. It is true that the commercial software has the capabilities to generate also hexahedral elements but in this case the model generation process is time consuming because the

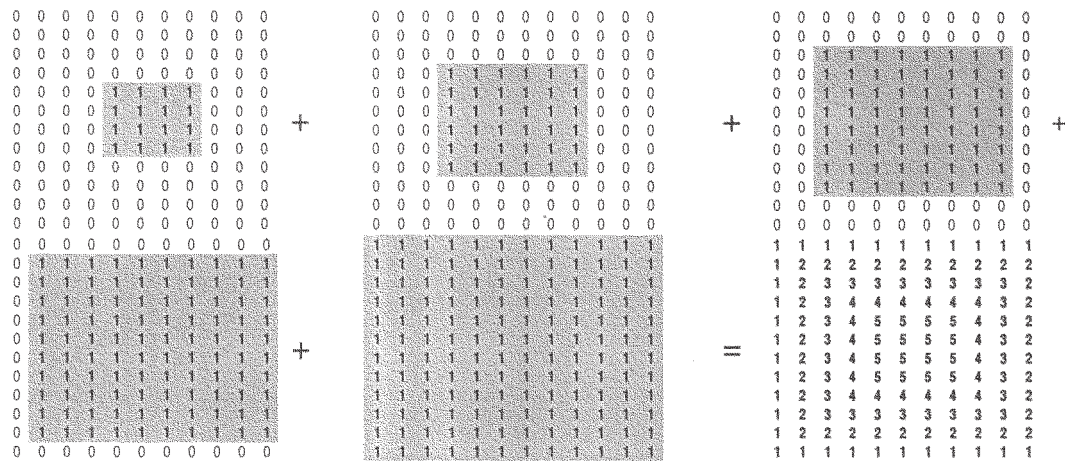


Fig. 9. Definition of the final grid map of the section

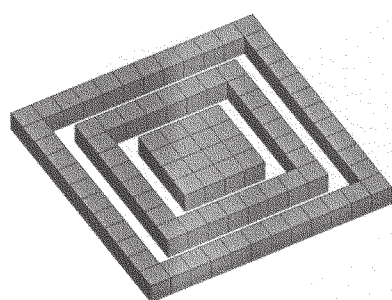


Fig. 10. Finite elements model and the associated grid map

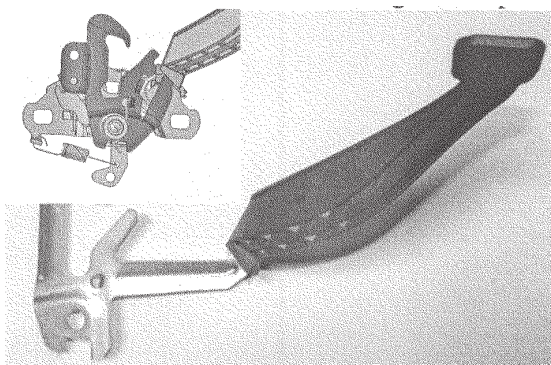


Fig. 11. Plastic handle and the safety hook. Left corner – Hood latch assembly

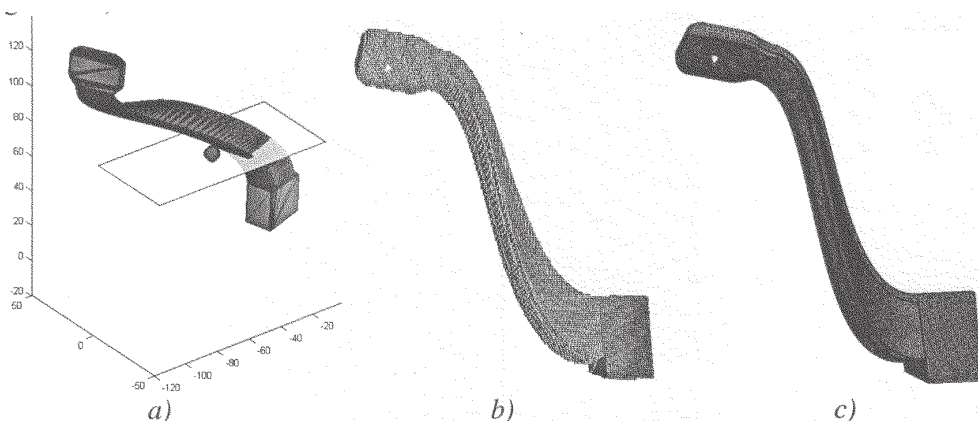


Fig. 12. Finite elements models

a) Geometrical model (.ply format); b) All hexahedral elements c) Geometry based model (tetra)

user must perform a lot of activities. As the method for all hexahedral finite elements model is automatic also for the commercial software the automatic method for mesh generation was selected. The all hexahedral model was generated using an element size of 0.75 mm and a total number of 62912 elements and 92262 nodes were

obtained. The tetrahedral model was generated using all the geometrical features of the part and a total number of 174768 elements and 41879 nodes were obtained. In order to compare the result using the hexahedral finite elements model and the tetrahedral finite elements model two simulation cases were defined. For the first simulation case the tip of the handle was loaded with a user defined force.

	Hexahedral model		Tetrahedral model	
	ELFORM 1	ELFORM 2	ELFORM 4	ELFORM 10
Stress	9.72	9.93	10.29	10.42
Displacement	5.54	4.842	6.474	5.661

Table 1
RESULTS: TOTAL DISPLACEMENT AND VON MISES STRESS (NOMINAL VALUES)

	Average	Hexahedral model (error)		Tetrahedral model (error)	
		ELFORM 1	ELFORM 2	ELFORM 4	ELFORM 10
Stress	10.09 MPa	-3.67	-1.59	1.98	3.27
Displacement	5.62 mm	-1.59	-13.98	15.01	0.56

Table 2
RESULTS: TOTAL DISPLACEMENT AND VON MISES STRESS (ERROR)

For the second case a drop test simulation was performed. The results were compared in terms of total displacement, stress (Von Mises) and time required to solve the numerical model (runtime).

Case 1. User defined load

Regarding the boundary conditions it was stated that the clamping section of the handle was fixed and a load of 10N was applied on the tip of the handle on a normal direction to the top surface of it.

The numerical models were solved using LS-Dyna software [11]. The solver can work using both implicit and explicit calculation methods.

For this case, when a specific load was applied, the solver was set to run with the implicit solver formulation.

Although the method is implicit the solution is not obtained in one step. The structure was loaded from 0 to the nominal force value in a user specified time interval.

The material model is elastic plastic with kinematic hardening. The density of the material is $\rho = 2000 \text{ kg/m}^3$, Young's modulus is $E = 1500 \text{ MPa}$, yield strength $\sigma_y = 50 \text{ MPa}$ and the hardening slope has a value of $E_t = 50 \text{ MPa}$.

It is known that in finite elements method the calculation is performed using the nodes while the elements are used to put into relations the corresponding nodes. Therefore the nodal result depends on the finite elements mathematical model.

In LS-Dyna there are multiple possibilities to assign to a specific finite element a mathematical formulation. Thus, for the hexahedral model ELFORM 1 (constant stress) and ELFORM 2 (fully integrated) mathematical formulations

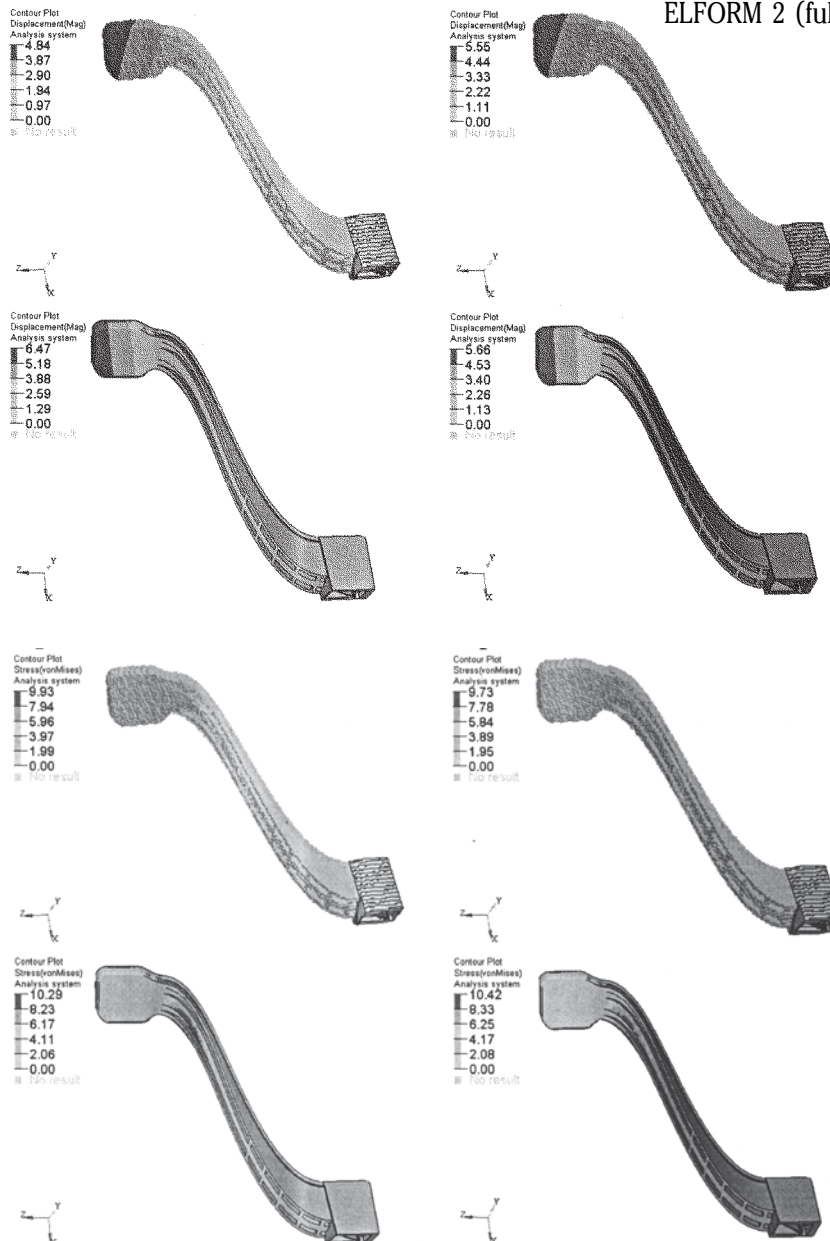


Fig. 13. Total displacement
a) Hex finite elements (ELFORM 2);
b) Hex finite elements (ELFORM 1);
c) Tetra finite elements (ELFORM 4);
d) Tetra finite elements (ELFORM 10)

Fig.14. Stress (Von Mises)
a) Hex finite elements (ELFORM 2);
b) Hex finite elements (ELFORM 1);
c) Tetra finite elements (ELFORM 4);
d) Tetra finite elements (ELFORM 10);

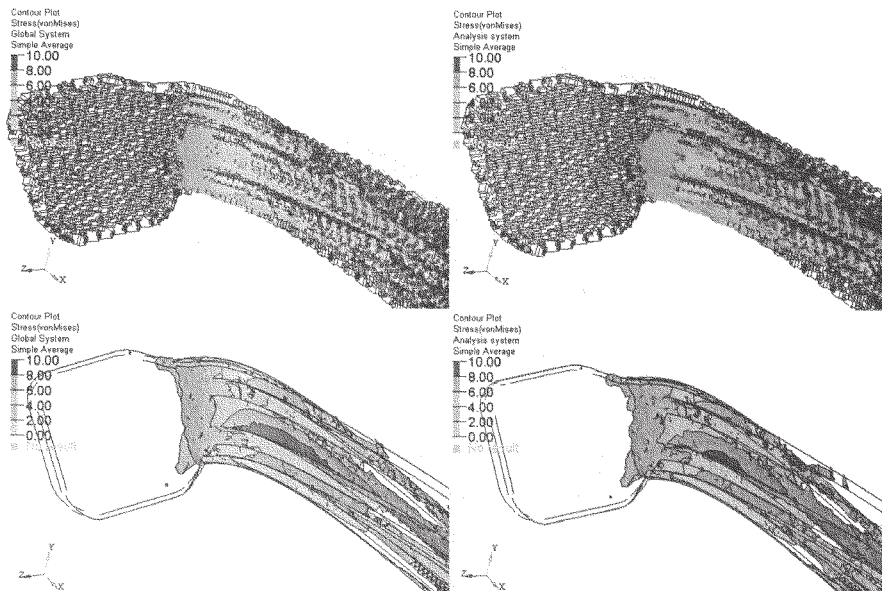


Fig.15. Stress (Von Mises – Averaged result. Iso - Contours)
a) Hex finite elements (ELFORM 2);
b) Hex finite elements (ELFORM 1);
c) Tetra finite elements (ELFORM 4);
d) Tetra finite elements (ELFORM 10);

for the elements were selected. For the tetrahedral model ELFORM 4 (fully integrated) and ELFORM 10 (1 point) mathematical formulations for the elements were selected.

The results were evaluated in terms of total displacement, Von Mises stress and total runtime. Table 1 presents the results obtained running these models.

In order to evaluate the result the averaged displacement and the averaged stress were used as a reference and the results compared.

Figure 13 presents the displacements maps for the models that were analysed.

Figure 14 presents the stress (Von Mises) maps for the models that were analysed.

Figure 15 presents a detailed view of the stress (Von Mises) iso-contours. It worth noticing that the areas with the high stress values are in good agreement for all models.

The models were solved using a PC machine with and Intel®Core2Duo™ at 2.4 GHz CPU and 2 Gb of RAM. The total runtime for each model is presented in table 3 as an interesting parameter that must be considered when a numerical model is developed. It can be noticed that the hexahedral models required a less runtime for both default

formulation (ELFORM 1) and for the fully integrated formulation (ELFORM 2).

Case 2. Drop test

For this case the structure was dropped over a flat rigid surface. In order to define a more efficient simulation model the part was positioned close to the rigid surface an initial velocity of 5 m/s was applied to the nodes. This corresponds to dropping the part from an initial height of 1.25 meters.

In these cases the simulations were performed using the explicit solver implemented in LS-Dyna. Thus the time step becomes an important parameter that must be considered. Given the simulation time (required by the simulated event) the time step defines the total runtime required for the machine to solve the problem. Thus a larger time step will give a smaller number of solution steps and this way the runtime decreases. A very small time step can lead to unreasonable runtimes and to a waste of hardware resources.

Using the hexahedral finite elements mesh generation method the numerical model can be developed having as an reference an user specified time step.

Table 3
RUNTIME – USED DEFINED LOAD

HEXAEDRAL model	ELFORM 1	4 minutes 19 seconds
	ELFORM 2	3 minutes 15 seconds
TETRAHEDRAL model	ELFORM 4	1 hours 59 minutes 26 seconds
	ELFORM 10	7 minutes 37 seconds

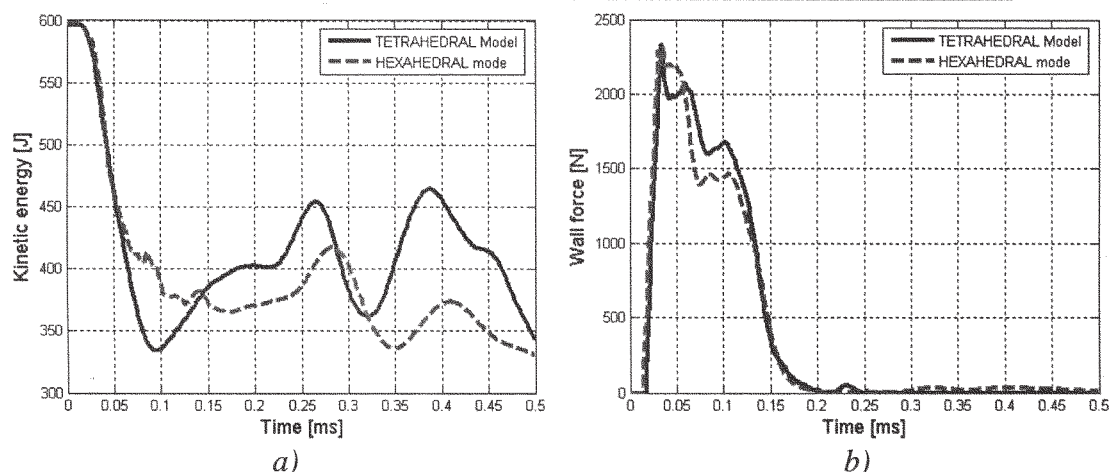


Fig.16. Drop test results. a) kinetic energy; b) contact force

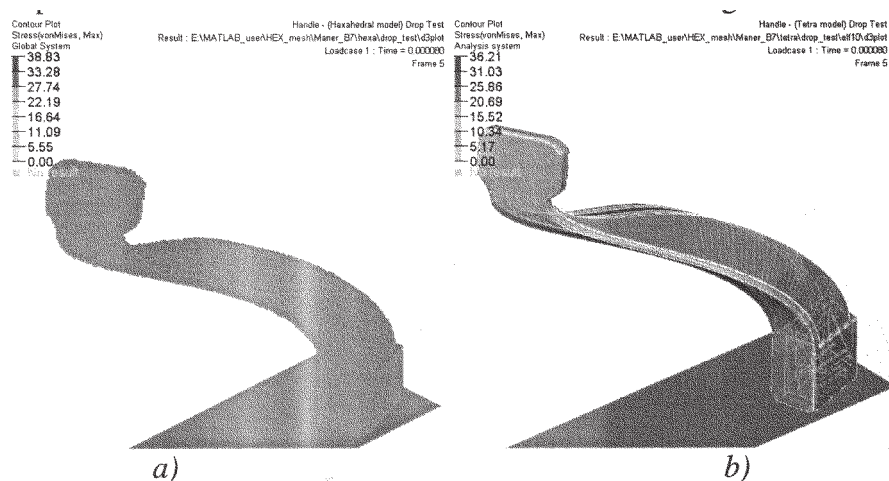


Fig. 17. Stress (Von Mises) distribution *a)* hexahedral model; *b)* tetrahedral model

Table 4
RUNTIME – DROP TEST

Model	Runtime	Element processing (% CPU time)
HEXAHEDRAL model	1 minutes 7 seconds	90.1
TETRAHEDRAL model	4 minutes 44 seconds	99.42

For 8 node solid elements (hexahedral) the time step is defined as:

$$\Delta t = \frac{1}{c} \cdot \frac{V_e}{A_{e,\max}} \quad (10)$$

where:

V_e is the element's volume (the minimum volume is considered),

$A_{e,\max}$ is the maximum area of the selected element
 c is the sound speed (for the corresponding material and element type).

For the 4 node solid element (tetrahedral) the time step is defined as:

$$\Delta t = \frac{1}{c} \cdot (\min l_c) \quad (11)$$

where l_c is the element altitude.

The kinetic energy and the contact force were evaluated using the numerical model. Results are presented in figure 17.

There are good similarities between the kinetic energy results obtained using the tetrahedral model and the hexahedral finite elements model.

Regarding the wall force for both cases the same peak force was recorded. Also the time when the peak value was recorded is the same and more of the contact time with the rigid wall is the same.

Figure 18 presents the stress contour after the contact with the rigid wall.

Table 4 presents the results of the simulation in terms of runtime and time required by the CPU to process the elements. The hexahedral model due to the larger elements size compared to the tetrahedral model required a shorter time to solve. Given the same rate of writing data on the disk it can be noticed that the total time to process the model is even shorter for the hexahedral model.

Conclusions

The paper presents a method to generate a complete hexahedral finite elements model. The procedure was developed and implemented using Matlab. The user can specify the geometrical model to be used (in .ply file format), specify the element size and can easily obtain the finite elements model. Based on a grid procedure the program cuts the geometrical model with parallel planes,

extract the contour and generates the finite elements model. For the geometrical model used as an application in this paper the time required to generate the finite elements model with a size of 0.75 mm was of 24 min and 4 s, that is a reasonable time considering the fact that the user has not to support the meshing process and in this state the custom code is not yet optimized for parallel processing.

The performance and the results obtained using the hexahedral model were compared with those obtained using the hexahedral model (generated also automatically by a commercially available software). Two load cases were analyzed. The first load case consisted in applying a force of 10 N on the tip of the handle. For the second load case a drop test like analysis was performed.

The results were analyzed and compared and good agreement between the results was found.

Therefore it can be stated that the present method is useful to generate the finite elements model when the tetrahedral finite element is required and even more when the total runtime has to be reasonable. It will be further improved in order to obtain better simulation models [12-14]. Also the finite elements generation method can be linked to commercially available of proprietary solvers in order to increase the applications range [15-17]. The method can be applied to generate finite elements for mechanical assemblies for various applications [18] or for multiphase materials [19, 20].

Some aspects regarding the subject were discussed in [21-23].

Aknowledgments: Part of the work presented was supported of the research study funded by the Romanian Ministry of Education, Research and Innovation through the CNMP grant no. 71-010/2007.

References

- 1.T.J. BAKER, Mesh generation: Art or science?, Progress in Aerospace Science 41, 2005, p. 29
- 2.J.C.M. TEO, C.K. CHUI, Z.L. WANG, S.H. ONG, C.H. YAN, S.C. WANG, H.K. WONG, S.H. TEOH, Heterogeneous meshing and biomechanical modeling of human spine, Medical Engineering & Physics, 29, 2007, p. 277

3. Y. ZHANG, C. BAJAJ, B.S. SOHN, 3D finite element meshing from imaging data, *Computer Methods in Applied Mechanics and Engineering* 194 2005, p. 5083
- 4.Y. ZHANG, C. BAJAJ, Adaptive and quality quadrilateral/hexahedral meshing from volumetric data, *Computer Methods in Applied Mechanics and Engineering* 195, 2006, p. 942
- 5.H. ZHANG, G. ZHAO, X. MA, Adaptive generation of hexahedral element mesh using an improved grid-based method, *Computer Aided Design*, 39, 2007, p. 914
- 6.M.L. STATEN, S. BENZLEY, M. SCOTT, A methodology for quadrilateral finite element mesh coarsening, *Engineering with Computers* 24 (2008) 241
- 7.Y. SUA,B, K.H. LEEB, A. SENTHIL KUMAR, Automatic hexahedral mesh generation for multi-domain composite models using a hybrid projective grid-based method, *Computer Aided Design*, 36, 2004, p. 203
- 8.C.S. CHONGA, A.S. KUMARB, K.H. LEE, Automatic solid decomposition and reduction for non-manifold geometric model generation, *Computer Aided Design*, 36, 2004, p. 1357
- 9.N.M. GROSLAND, K.H. SHIVANNA, V.A. MAGNOTTA, N.A. KALLEMEYN, N.A. DEVRIES, S.C. TADEPALLI, C.LISLE, IA-FEMesh: An open-source, interactive, multiblock approach to anatomic finite element model development, *Computer Methods and Programs in Biomedicine* 94, 2009, p. 96
10. *** The Mathworks Inc., 'MATLAB Users Guide (R14)', The Mathworks Inc.
- 11.*** Livermore Software Technology Corporation, 'LS-DYNA Keyword Manual v970', Livermore, CA, Livermore Software Technology Corporation, April 2003.
- 12.J.F. SHEPHERD, M.W. DEWEY, A.C.WOODBURY, S.E. BENZLEY, M.L. STATEN, S.J. OWEN, Adaptive mesh coarsening for quadrilateral and hexahedral meshes, *Finite Elements in Analysis and Design* (2009) doi:10.1016/j.finel.2009.06.024
- 13.D.R. WHITE, S. SAIGAL, S.J. OWEN, CCSweep: automatic decomposition of multi-sweep volumes, *Engineering with Computers* 20, 2004, p. 222
- 14.J.N. HAN, N.J. CHO, S.W. CHAE, Y. CHOI, Hybrid three-dimensional mesh generation from quad-dominant surface meshes, *International Journal of Automotive Technology* 9 , 2008, p. 633
- 15.N. ILIESCU, A. HADAR, S.D. PASTRAMA, *Mat. Plast.*, **46** , no. 1, 2009, p. 91
16. KIM, K.D.,YANG, D.Y. ,JEONG, J.H. , Adaptive refinement techniques based on tetrahedral and hexahedral grids for finite element analysis of mold filling in casting processes, *Computer Methods in Applied Mechanics and Engineering* 195, 2006, p. 6799
17. FERNANDES, J.L.M., MARTINS,P.A.F., All-hexahedral remeshing for the finite element analysis of metal forming processes, *Finite Elements in Analysis and Design*, 43, 2007, p. 666
- 18.MANESCU, T.S., PETRE, C.C., ZAHARIA,N.I., MANESCU JR,T., BAYER, M., *Mat. Plast.*, **46**, 2009, p. 448
19. LUCA MOTOC, D., CURTU, I., *Mat. Plast.*, **46**, 2009, p. 462
- 20.CHIRICA, E., BEZNEA, F., CHIRICA, R., *Mat. Plast.*, **46**, 2009, p. 243
- 21,TABACU, St., HADAR, A., MARIN ESCU, D., IVANESCU, M., BALASOIU, V., *Mat. Plast.*, **46**, no. 2, 2009, p. 192
22. HODOLIC, J., MATIN, I., STEVIC, M., VUKELIC, D., *Mat. Plast.*, **46**, no. 3, 2009, p. 236
23. NEDELCU, D., FETECU, C., CIOFU, C., MINDRU, D., *Mat. Plast.*, **46**, no. 3, 2009, p. 269

Manuscript received: 0.12.2009