1st Conference on
Advances and Applications of GiD

20, 21 and 22 February 2002
Barcelona, Spain

Publication CIMNE Nº-210, Febrero 2002

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

ÍNDICE

Recent Advances in Unstructured Mesh and Point Generation by R. Löhner

Virtual Reality Techniques for Prototyping and Simulation in the Design Process by P. Brunet

The Generation of Unstructured Boundary-Conforming Meshes of High-Order Elements by J. Peiró, J. S. Spencer and S. Giordana

Web Computing: New Generation of Remote Analysis by R. Ribó

Application of GID to Microwave Printed Circuits and Antennas Design by F. Tiezzi and J. R. Mosig

Fractalcoms Project: Meshing Fractal Geometries with GiD by J. Parrón, J. M. Rius, J. Romeu, A. Heldring and G. Bugeda

A GiD-FEAP Interface by A. Ghisi

Customization of GID & FEAP for Scientific Applications by T. Riable, D. Frebmann, R.L. Taylor and P. Wriggers

Application of GiD to 3D Biomechanics Problems, Interfacing with ABAQUS and FEAP by J.C. García, J. Rodríguez and J. M. Goicolea

FinitePoint2D: A Meshless GiD Code for Linear Elasticity by E. Escolano and F. Perazzo

Modelización Numérica y Virtualización de Edificios Históricos. Interface GiD-Sap2000-VRML por F. Muñoz Salinas and J. Maristany i Carreras

Treating the Geometry of a Floating Caisson to Obtain a Structured Mesh in GiD by B. Suárez, A. Foces, J. Pereda and M. Cacho

La Asignación de Condiciones sobre Capas en la Interfaz Gráfica del Programa STAMPACK por G. Socorro, E. Escolano y L. Neamtu

A Methodology for the Constrained Optimization of Tunnels with GiD by L. Gil and A. Andreu

From GiD to Market: COMPASS Experiences in Developing GiD-based RamSeries and Tdyn Programs by J. García-Espinosa

Code_Bright/GiD: A 3-D program for Thermo-Hydro-Mechanical Analysis in Geological Media by J. Vaunat and S. Olivella

Textures in Simulation by M. A. de Riera

New Environment for the Simulation, Optimisation and Control of Food Processing by E. Balsa-Canto, C. González, J. Mora, J. R. Banga and A. A. Alonso

Numerical Simulation of Complex Large Deformation Processes by B. M. Chaparro, J. M. Antunes, L. F. Menezes, S. Thuiller and P. Y. Manach

Finite Element Simulation of Ultramicrohardness Tests by J.M. Antunes, B. M. Chaparro, L. F. Menezes and J. V. Fernandes

Finite Element Modelling and Analysis of Residual Stresses in Al-SiC Metal Matrix Composites with GiD by A. Andrade-Campos, J.A. M. Pinho-da-Cruz and F. Teixeira-Dias

GiD Customization for Applications in Metal Forming Optimization by P. A. Muñoz-Rojas, J. S. O. Fonseca and G. J. Creus

Modelling of Powder Based Complex Shape Manufacturing by O. Mikhailov and M. Shtern

Preculiarities of the Use of Package GiD for the Analysis of Forming Processes in Powder Metallurgy by M. Shtern and O. Mikhailov

RECENT ADVANCES IN UNSTRUCTURED MESH AND POINT GENERATION

R. Löhner[1]

[1]*School of Computational Sciences and Informatics*
M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, USA

SUMMARY: Recent advances in mesh and point generation are reviewed. These include: meshing of discrete surfaces, parallel advancing front methods, improvements in RANS gridding via directional enrichment and tighter sphere packing for discrete particle methods. Several examples are included to illustrate the effectiveness of the developed techniques.

KEYWORDS: Unstructured Grid Generation, Parallel Meshing, Point Generation

1. INTRODUCTION

Field solvers based on unstructured grids play an ever increasing role in physics and engineering. Presently, all of the large-scale commercial software packages in Computational Structural Dynamics (CSD), Computational Fluid Dynamics (CFD) and Computational Thermodynamics (CTD) are based on finite element or finite volume methods operating on unstructured grids. The relentless advance in the physical complexity that may be modeled by these codes, together with increases in algorithmic efficiency and computer power, have placed a premium on the reliable, automatic generation of grids for complex geometries that are suitable for the problem at hand. The present paper reviews some recent developments in this direction. The first topic discussed is surface meshing, a key bottleneck faced in many industries as the virtual prototyping process is automated. The second topic is concerned with fast meshing for very large problems, focusing on parallel meshing. The third topic is reliable gridding of regions that require elements with extreme stretching, such as boundary or shear layers in flows. Finally, as a fourth topic, the generation of suitable clouds of points for the finite point or discrete particle methods is discussed.

2. DISCRETE SURFACE MESHING

The rapid, user-friendly definition of the surfaces defining the computational domain has been an important goal during the last decade. Surfaces can be defined either analytically (using B-Splines, NURBS, Coon's patches, etc.) or via triangulations. The latter option is particularly interesting for data sets stemming from remote sensing data (e.g. geographical data) or medical imaging [Ceb01]. An interesting observation made over the last years is that an increasing number of data sets used to define the geometry of CFD domains is given in the form of triangulations, even though the

CAD data is available. The reason for this shift in data type is that a watertight triangulation defines in a unique way the domain considered, and does not require any futher geometric cleanup operations. This is not the case with native CAD datasets, in which we frequently encounter very large numbers of patches, overlapping patches, gaps, and other geometric pathologies that require user intervention. These developments have renewed the interest in robust surface meshing of so-called discrete surfaces (DS). Of the many innovations introduced during the last years, we mention:

- Automatic preprocessing/improvement of the DS;
- Introduction of a visibility horizon filter for close points/ sides;
- Strict enforcement of continuous topology;
- Improved 2D cross-check; and
- Adaptive background grid element size definition.

In the sequel, we expand on a few of these. For a detailed description, see [Til02].

Visibility Horizon: The advancing front method adds a new surface triangle by removing a side from the active front. Among the decisions required is whether to take an existing point to form the new triangle, or to introduce a new point. The list of close (i.e. possible) points is obtained from a proximity search. This list of possible close points is reduced by several tests (visibility, angles, etc.). Perhaps the most important validation test is based on the neighbour to neighbour search on the given DS. The starting face for the search is given by the underlying DS face at the midpoint of the side being removed from the active front. The direction is given by the close point. Any close point that can not be reached on the given DS using the neighbour to neighbour search is removed from the list. A similar procedure is used to filter close sides, which are required to test if the new triangle crosses the existing active front of sides.

Continuous Topology: A typical neighbour to neighbour search will not stop at internal DS geometry lines (given by sharp edges). Therefore, a so-called visibility horizon was introduced for the neighbour to neighbour search. All neighbour to neighbour edges given by internal geometry lines or angles beyond a certain tolerance are marked. In this way, the neighbour to neighbour search can recognize them. The neighbour to neighbour search stops at these internal geometry lines. The close point is marked as unreachable and removed from the list of candidates.

Adaptive Background Grid Based on DS: For complex geometries, the specification of desired element size can be a tedious, time-consuming process. Adaptive background grids [Löh96, Löh97] offer the possibility to reduce drastically the required level of human input. DS offer, by their way of defining the surface, a natural way to refine the background grid and to define the mesh size required for a proper definition of the geometry.

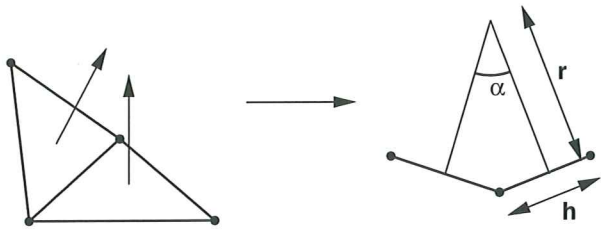


Figure 1 Measuring Surface Curvature

With the notation defined in Figure 1, the angle between two faces is given by

$$\frac{h}{2r} = tan(\frac{\alpha}{2}) \quad . \tag{1}$$

This implies that for a given element size h_g and angle α_g, the element size for a prescribed angle α_p should be:

$$h_p = h_g \frac{tan(\frac{\alpha_p}{2})}{tan(\frac{\alpha_g}{2})} \quad . \tag{2}$$

For other measures of surface accuracy, similar formulae will be encountered. Given a prescribed angle α_p, the point-distances of the given DS surface triangulation are compared to those obtained from Eqn.(2) and reduced appropriately:

$$\delta_i = \min(\delta_i, h_p) \quad . \tag{3}$$

These new point-distances are then used to adjust and/or refine the background grid. As an example of the effective use of adaptive background grids, we consider the air flow in the bronchii and lungs. A segmented image, together with the cuts at the extremities of the smaller branches, is shown in Figure 2a. The mesh sizes are automatically obtained from an adaptive background grid with 6 levels of refinement. This produced the surface mesh shown in Figure 2b. One can discern the smaller elements in regions of higher curvature and smaller vessel diameter. The volume mesh had approximately 1 million elements. In this first study, only the steady airflow was considered. The results obtained can be seen in Figures 2c,d, which show surface pressures and iso-surfaces of constant absolute value of velocity.

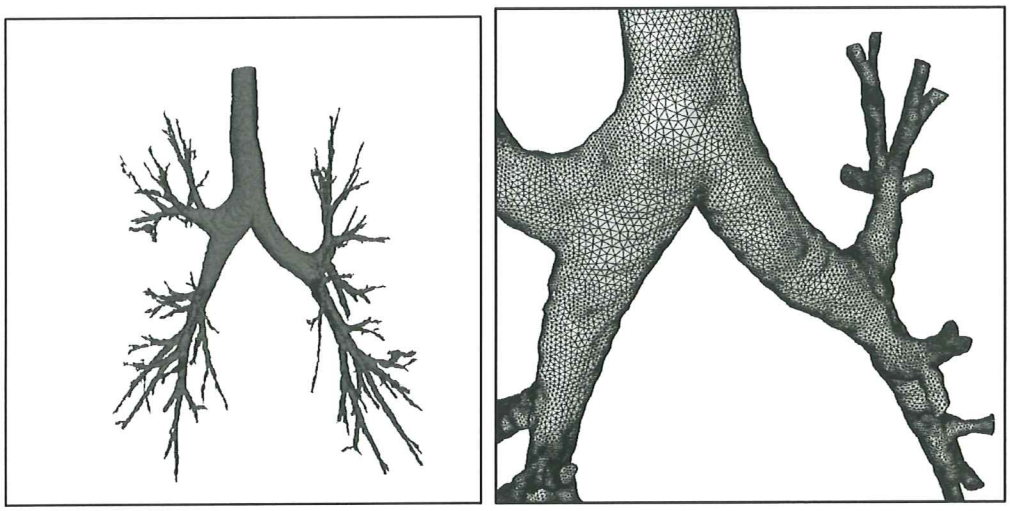


Figure 2a,b Thorax: Segmented Image and Surface Mesh

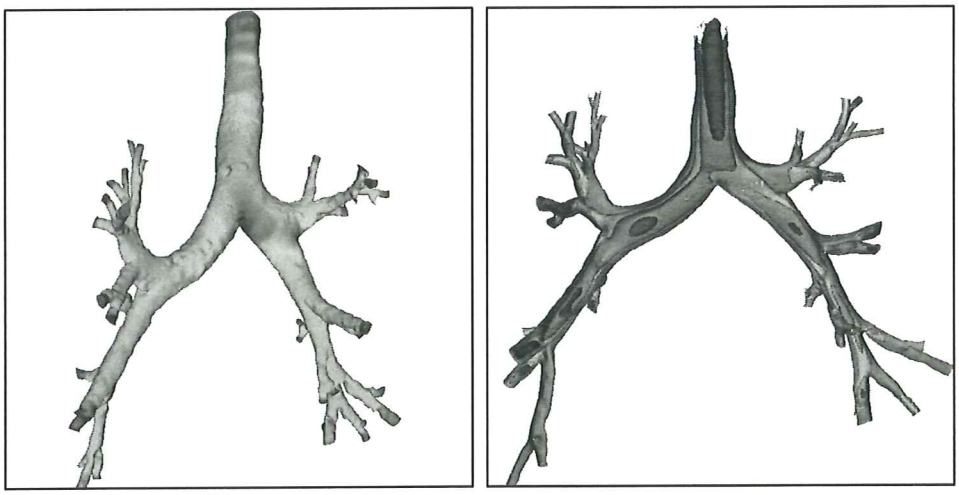


Figure 2c,d Thorax: Surface Pressure and Isosurfaces of Speed

3. PARALLEL GRID GENERATION

Over the last decade, major efforts have been devoted to harness the power of parallel computer platforms. While many CFD and CSD solvers have been ported to parallel machines, grid generators have lagged behind. For applications where remeshing is an integral part of simulations, e.g. problems with moving bodies [Löh90, Mes93,

Mes96, Bau96, Kam96, Löh98b, Has98] or changing topologies [Bau98, Bau99], the time required for mesh regeneration can easily consume more than 50% of the total time required to solve the problem. Faced with this situation, a number of efforts have been reported on parallel grid generation [Löh92, dCo94, Sho95, dCo95, Oku96, Che97, Oku97, Sai99].

The two most common ways of generating unstructured grids are the Advancing Front Technique (AFT) [Per87, Per88, Löh88a,b, Per90, Per92, Jin93, Fry94, Löh96] and the Generalized Delaunay Triangulation (GDT) [Bak89, Geo91, Wea92, Wea94, Mar95a]. The AFT introduces one element at a time, while the GDT introduces a new point at a time. Thus, both of these techniques are, in principle, scalar by nature, with a large variation in the number of operations required to introduce a new element or point. While coding and data structures may influence the scalar speed of the 'core' AFT or GDT, one often finds that for large-scale applications, the evaluation of the desired element size and shape in space, given by background grids, sources or other means [Löh96] consumes the largest fraction of the total grid generation time. Unstructured grid generators based on the AFT may be parallelized by invoking distance arguments, i.e. the introduction of a new element only affects (and is affected by) the immediate vicinity. This allows for the introduction of elements in parallel, provided that sufficient distance lies between them.

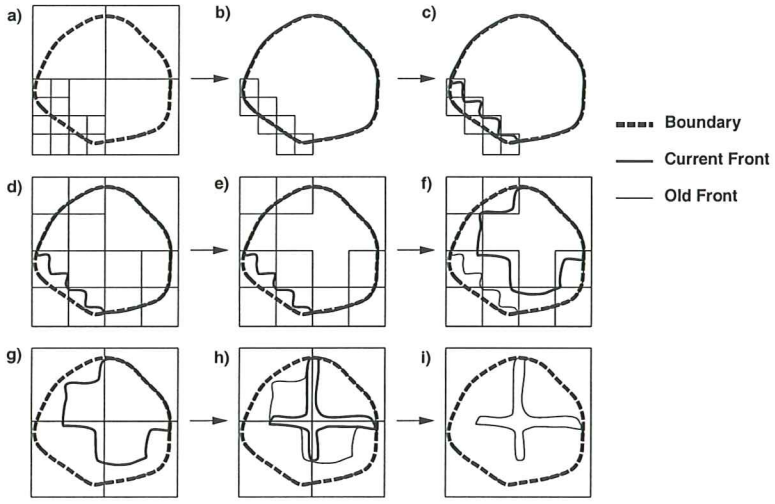


Figure 3 Parallel Grid Generation

A convenient way of delimiting the possible zones where elements may be introduced by each processor is via boxes. These boxes may be obtained in a variety of ways, i.e. via bins, binary recursive trees, or octrees. We have found the octree to be the best of these possibilities, particularly for grids with a large variation of element size. In order to recover a parallel gridding procedure that resembles closely the advancing front technique on scalar machines, only the boxes covering the active front in regions

where the smallest new elements are being introduced are considered. This has been shown schematically in Figure 3a,b for a simple 2-D domain. After these boxes have been filled with elements (Figure 3c), the process starts anew: a new octree is built (Figure 3d), new boxes are created (Figure 3e) and meshed in parallel (Figure 3f). This cycle is repeated until no faces are left in the active front (Figures 3g-i).
At the end of each parallel gridding pass, each one of the boxes gridded can have an internal boundary of faces. For a large number of boxes, this could result in a very large number of faces for the active front. This problem can be avoided by shifting the boxes slightly, and then regridding them again in parallel, as shown in Figure 4. This simple technique has the effect of eliminating almost all of the faces between boxes with a minor modification of the basic parallel gridding algorithm.

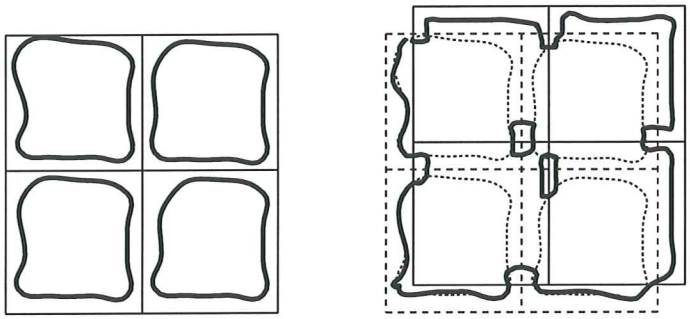


Figure 4 Shift and Regrid Technique

If we define as d_{min} the minimum element size in the active front, and as s_{min} the minimum box size in which elements are to be generated, the parallel AFT proceeds as follows:

`WHILE`: There are active faces left:

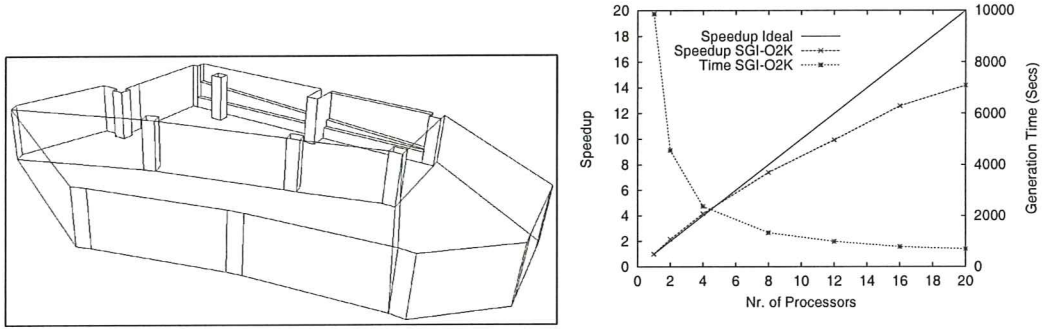
- Form an octree with minimum octant size s_{min} for the active points;
- Retain the octants that have faces that will generate elements of size d_{min} to $c_l \cdot d_{min}$;
- If too many octants are left: agglomerate them into boxes;
- `DO ISHFT=0,2`:
  - `IF: ISHFT.NE.0`:
    Shift the boxes by a preset amount;
  - `ENDIF`
  - Generate, in parallel, elements in these boxes, allowing only elements up to a size of $c_l \cdot d_{min}$;
- `ENDDO`
- Increase $d_{min} = 1.5 * d_{min}$, $s_{min} = 1.5 * s_{min}$;

`ENDWHILE`

The increase factor allowed is typically in the range $c_l = 1.5-2.0$. Recent improvements to the described parallel advancing front grid generator include:

- A more reliable workload estimation for spatial domains;
- The introduction of a limit for the number of elements a processor may generate (to prevent one processor from consuming much more CPU time than all the others);
- A reduction in the number of processors used if the number of remaining faces is too small (so-called 'scalar endgame'); and
- A reduction of processors used in the post-generation/improvement phase if the number of elements modified is too small.

We include two examples to demonstrate the effectiveness of parallel mesh generation. Both examples were obtained on an SGI Origin 2000 running in shared memory mode. Figures 5a,b show the outline and timings obtained for the garage of an office complex. The mesh had approximately 9.2 million tetrahedra. As one can see, although not perfect, speedups are comparable to those of production CFD codes.



Figures 5a,b Garage: Wireframe and Speedups Obtained

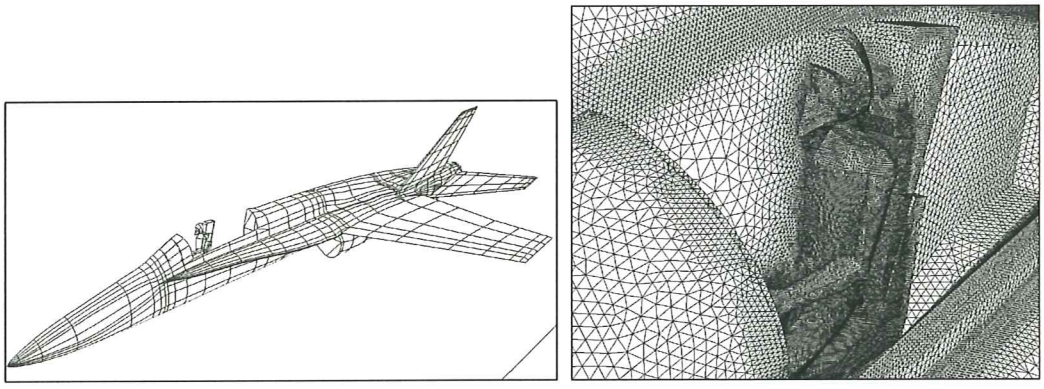


Figure 6a,b F18: Outline of Domain and Surface Mesh (Closeup)

Figure 6 shows a pilot seat ejection case, a typical problem with moving bodies that

requires several remeshings during the course of a simulation [Bau93, Bau95, Bau97, Sha00]. The outline of the domain is shown in Figure 6a. The surface triangulation of the final mesh, which had approximately 14 million tetrahedra, is shown in Figure 6b. The smallest and largest specified element side lengths were 0.65 cm and 250.00 cm respectively, i.e. an edge-length ratio of approximately $1 : 4 \cdot 10^2$ and a volume ratio of $1 : 5.6 \cdot 10^7$. The spatial variation of element size was specified via approximately 110 sources [Löh96a, Löh97].

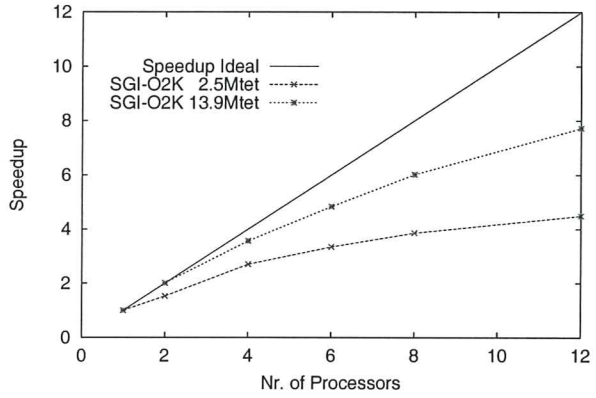


Figure 6c F18: Speedups Obtained

The speedup obtained for two different grid sizes is displayed in Figure 6c. One can see that the grid generator scales well with the number of processors and that scalability improves with the amount of work required. For more details on the parallel grid generator, see [Löh01].

4. IMPROVEMENTS IN RANS GRIDDING

The generation of high-quality grids suitable for RANS calculations of flows in and around complex geometries continues to be an active area of research. The two most common ways of generating highly stretched grids suitable for RANS calculations are the advancing layers technique (ALT) and the directional enrichment technique (DET). The ALT follows the spirit of the advancing front technique (AFT): starting from the 'wetted surface', add thin layers of elements until an isotropic mesh is achieved [Löh93, Pir94, Mar95b, Pir96]. From this point onwards, the mesh is completed with the AFT. The DET is an extension of the Generalized Delaunay Triangulation (GDT). At first, an isotropic mesh is generated (either via AFT or GDT). Then, the points in the regions where stretched elements are to be generated are removed (typically by collapsing edges). Thereafter, points are introduced in the near-wall region so as to obtain the desired RANS mesh. The newly introduced points are reconnected using the GDT [Per96, Löh99]. This latter technique, while general, can still generate bad grids

for complex geometries. A recent improvement was obtained by considering the layer-number of the points when performing diagonal/face swapping in 3-D. The quality of elements that cross several point-layers is reduced, leading to swaps that favour elements with minimal layer jump. The improvement is shown in Figure 7, where a corner is considered.

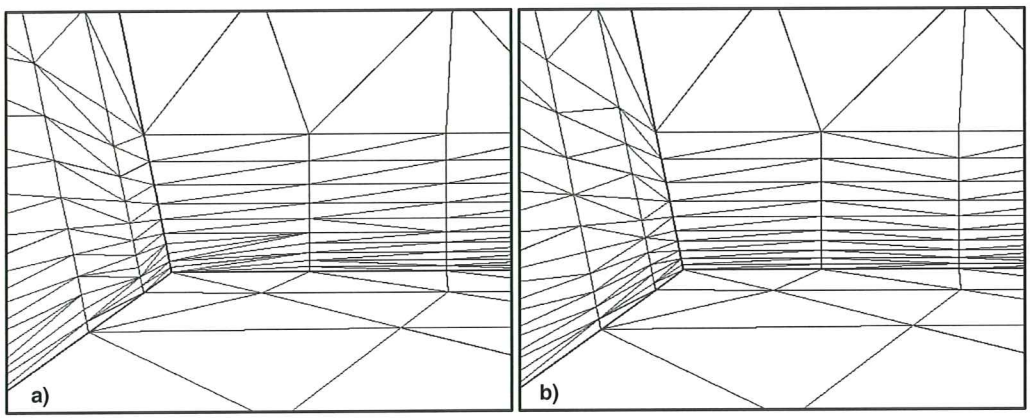


Figure 7 Corner: Surface Meshes Obtained

5. POINT GENERATION

Over the course of the last decade, a number of 'gridless', 'mesh free', or 'discrete particle' schemes have appeared in the literature (see, e.g. [Nay72, Bat93, Bel94, Oña96a, Oña96b, Liu96] and the references cited therein). The interest in these schemes stems from:

a) The perceived difficulty of generating volume filling grids for problems characterized by complex geometries and/or complex physics;
b) The perceived difficulty of producing high-order numerical schemes for complex geometries; and
c) The perceived failure of continuum models to capture complex material behaviour, such as that encountered in concrete failure.

The generation of an appropriate global cloud of points seems, at first, much simpler than the generation of points and elements. Three general ways of generating clouds of points have been reported to date:

- Generate a mesh using traditional grid generators, and remove the elements in a post-processing step;
- Generate a coarse mesh of elements, fill each element with an appropriate number of particles, and remove the elements in a post-processing step;
- Generate directly clouds of points using an advancing front point method [Löh98a].

The first approach defeats the purpose stated as one of the reasons for going to point methods. The second approach will have difficulties in regions where small elements are required due to geometrical constraints.

Only the third scheme allows for the direct generation of clouds of points with the same degree of flexibility as advanced unstructured grid generators [Löh88, Per88, Per90, Geo91, Jin93, Fry94, Löh96a, Löh97]. The mean distance between points (or, equivalently, the point density) is specified by means of background grids, sources and density attached to CAD-entities. In order not to generate points outside the computational domain, one assumes an initial triangulation of the surface that is compatible with the desired mean distance between points specified by the user. Starting from this initial 'front' of points, new points are added, until no further points can be introduced. Whereas the advancing front technique for the generation of volume grids removes one face at a time to generate elements, the present scheme removes one point at a time, attempting to introduce as many points as possible in its immediate neighborhood.

5.1 Advancing Point Generation Algorithm

Assume as given:

- A specification of the desired mean distance between points in space. This is done here through a combination of background grids, sources and mean distance to neighbours attached to CAD-data (see [Löh96a, Löh97] for more details).
- An initial triangulation of the surface, with the face normals pointing towards the interior of the domain to be filled with points.

With reference to Figure 8, the complete advancing front point generation algorithm may be summarized as follows:

- Determine the required mean point distance for the points of the triangulation;
- `while`: there are active points in the front:
  - Remove the point `ipout` with the smallest specified mean distance to neighbours from the front;
  - With the specified mean point distance: determine the coordinates of `nposs` possible new neighbours. This is done using a stencil, some of which are shown in Figure 9;
  - Find all existing points in the neighborhood of `ipout`;
  - Find all boundary faces in the neighborhood of `ipout`;
  - `do`: For each one of the possible new neighbour points `ipnew`:
    - If there exists a point closer than a minimum distance `dminp` from `ipnew`:
      $\Rightarrow$ skip `ipnew`;
    - If the line connecting `ipout` and `ipnew` crosses existing faces:
      $\Rightarrow$ skip `ipnew`;
    - Determine the required mean point distance for `ipnew`;
    - Increment the number of points by one;
    - Introduce `ipnew` to the list of coordinates;
    - Introduce `ipnew` to the list of active front points;

    `enddo`
- `endwhile`

Details on point stencils, boundary consistency checks and efficient data structures for the search operations required may be found in [Löh98a].

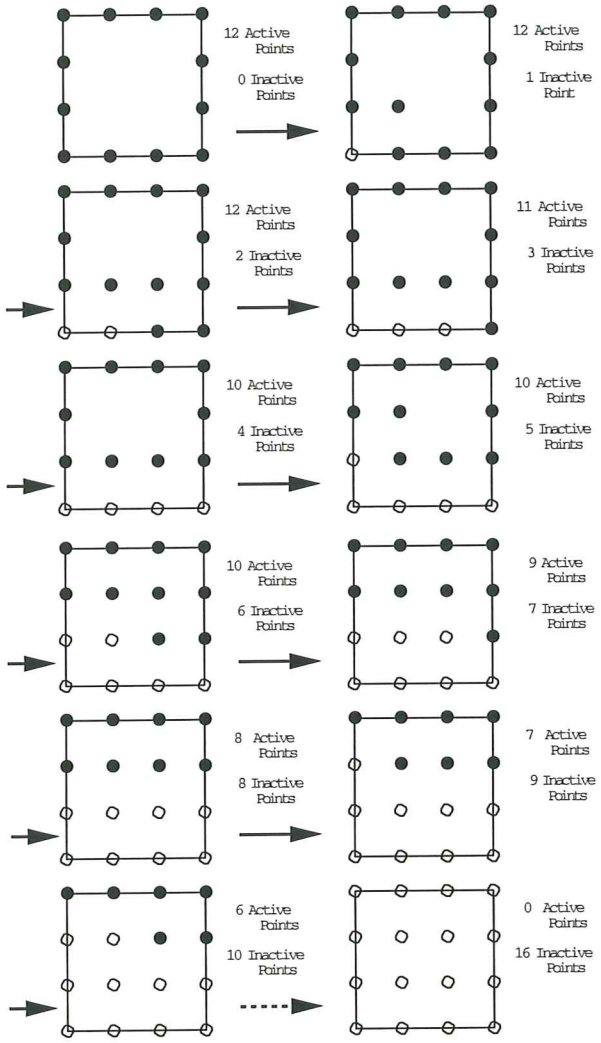


Figure 8 Advancing Front Point Generation

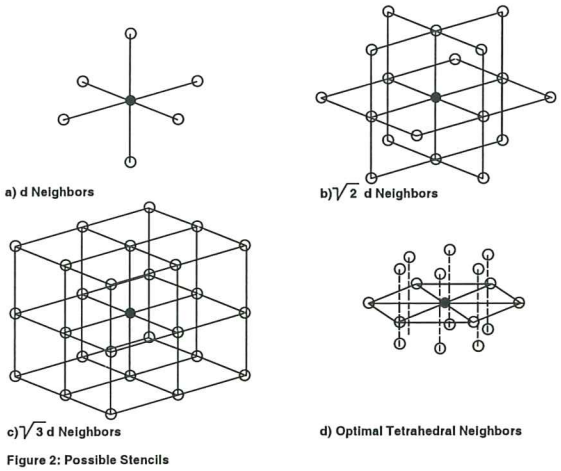


Figure 9 Stencils Used for Point Introduction

5.2 Tight Packing Option

The generation of clouds of **points** is less restrictive than the generation of clouds of **spheres**. In the latter, each sphere has a given radius, and the spheres may not interpenetrate each other. Several clouds of spheres generated with the technique described indicated that the packing obtained was below the expected maximum packing. This was particularly the case for clouds with a random variation of radius for the spheres. In order to increase the packing, the newly introduced sphere is moved as close as possible to the existing spheres (see Figure 10). An attempt is made to move the new sphere to the three closest spheres. Should this fail, the two closest spheres are considered. And should this also fail, the newly introduced sphere is moved as close as possible to the originating sphere. The spheres in question are assumed to form a spring system with contact forces. This spring system is relaxed until convergence to the closest possible position is achieved.

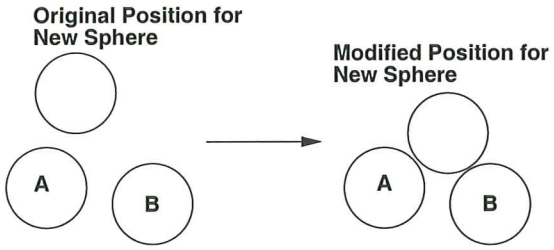


Figure 10 Movement of Spheres to Achieve Maximum Packing

It was found that this 'movement to close spheres' greatly increases the packing obtained for the final cloud of spheres. Figure 11 shows a typical discrete particle model for a concrete specimen. The domain is a cube, and the requirement was to generate spheres with a Gaussian size distribution and a diameter variation of 20%. The number of spheres generated was `npoin=6,950`, of which `nmovp=6,889` were moved during the generation process. The generated cloud of spheres had a volume fill fraction of $v_f = 0.485$. The average number of contacts was $n_c = 5.8$ per sphere.

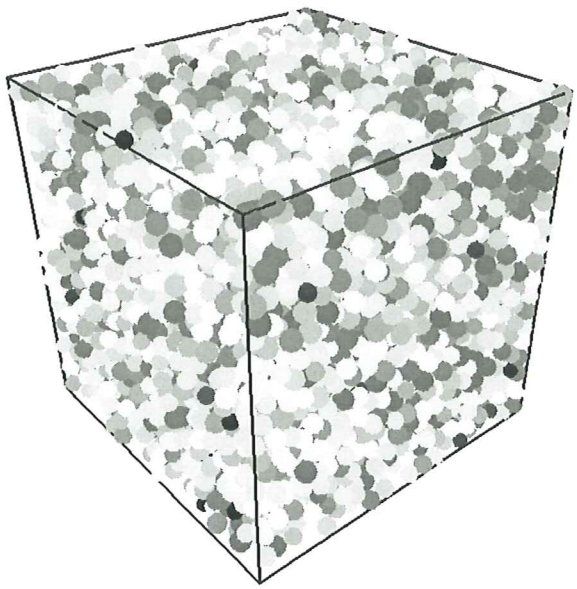


Figure 11 Cloud of Spheres for DMP Modeling of Concrete Specimen

Figure 12 shows snapshots of a sand-filling simulation with approximately $2{\cdot}10^5$ spheres. The sand was modelled by particles of approximately 2mm diameter. This is obviously larger than the actual sand, but nevertheless the material behaviour is captured. The figure shows the particles in a particular cut plane at different times during the filling and compaction process.

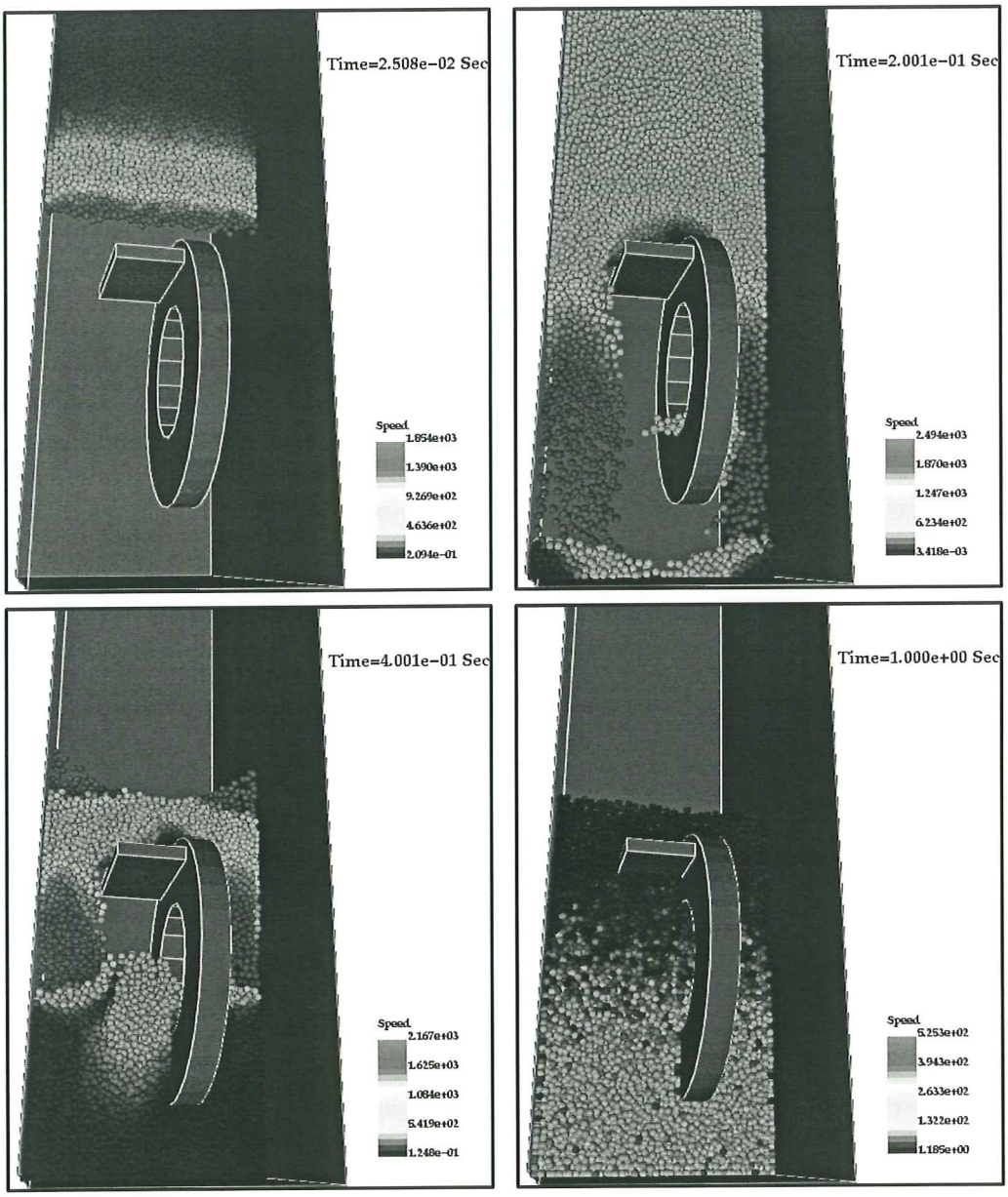


Figure 12 Sand-Filling Simulation With $2 \cdot 10^5$ Spheres

CONCLUSIONS AND OUTLOOK

Over the last decade, unstructured grid generation has become a focal point of research. The present paper has reviewed some recent developments in discrete surface gridding, parallel mesh generation, RANS gridding and the filling of space with a desired point density. Each one of these areas has seen notable advances.

As with any development of technology, the list of possible improvements is never ending. Among the areas of research in unstructured grid generation requiring immediate attention we mention:

- Automatic detection and proper meshing of 'stiff' geometric features, such as thin gaps;
- Scaling of parallel gridding to large numbers of processors (> 100);
- Improved RANS gridding, especially for separated, transient wake regions;
- Faster sphere generation to a prescribed volume fraction; and
- 'Smoothing' procedures for clouds of points.

REFERENCES

[Bak89] T.J. Baker - Developments and Trends in Three-Dimensional Mesh Generation. *Appl. Num. Math.* **5**, 275-304 (1989).

[Bat93] J. Batina - A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Aircraft Configurations; *AIAA*-93-0333 (1993).

[Bau93] J.D. Baum. H. Luo and R. Löhner - Numerical Simulation of a Blast Inside a Boeing 747; *AIAA*--93-3091 (1993).

[Bau94] J.D. Baum, H. Luo and R. Löhner - A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies; *AIAA*-94-0414 (1994).

[Bau95] J.D. Baum, H. Luo and R. Löhner - Validation of a New ALE, Adaptive Unstructured Moving Body Methodology for Multi-Store Ejection Simulations; *AIAA*-95-1792 (1995).

[Bau96] J.D. Baum, H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman - A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; *AIAA*-96-0795 (1996).

[Bau98] J.D. Baum, H. Luo and R. Löhner - The Numerical Simulation of Strongly Unsteady Flows With Hundreds of Moving Bodies; *AIAA*-98-0788 (1998).

[Bau99] J.D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone and C. Charman - A Coupled CFD/CSD Methodology for Modeling Weapon Detonation and Fragmentation; *AIAA*-99-0794 (1999).

[Bel94] T. Belytschko, Y. Lu and L. Gu - Element Free Galerkin Methods; *Int. J. Num. Meth. Eng.* 37, 229-256 (1994).

[Che97] L.P. Chew, N. Chrisochoides and F. Sukup - Parallel Constrained Delaunay Meshing; *Proc. 1997 Workshop on Trends in Unstructured Mesh Generation*, June (1997).

[Ceb01] J.R. Cebral and R. Löhner - From Medical Images to Anatomically Accurate Finite Element Grids; *Int. J. Num. Meth. Eng.* 51, 985-1008 (2001).

[dCo94] H.L. de Cougny, M.S. Shephard and C. Ozturan - Parallel Three-Dimensional Mesh Generation; *Computing Systems in Engineering* 5, 311-323 (1994).

[dCo95] H.L. de Cougny, M.S. Shephard and C. Ozturan - Parallel Three-Dimensional Mesh Generation on Distributed Memory MIMD Computers; *Tech. Rep. SCOREC Rep. # 7*, Rensselaer Polytechnic Institute (1995).

[Fry94] J. Frykestig - Advancing Front Mesh Generation Techniques with Application to the Finite Element Method; *Pub. 94:10*, Chalmers University of Technology; Göteborg, Sweden (1994).

[Geo91] P.L. George, F. Hecht and E. Saltel - Automatic Mesh Generator With Specified Boundary; *Comp. Meth. Appl. Mech. Eng.* 92, 269-288 (1991).

[Has98] O. Hassan, L.B. Bayne, K. Morgan and N. P. Weatherill - An Adaptive Unstructured Mesh Method for Transient Flows Involving Moving Boundaries; pp. 662-674 in *Computational Fluid Dynamics '98* (K.D. Papailiou, D. Tsahalis, J. Périaux and D. Knörzer eds.) Wiley (1998).

[Jin93] H. Jin and R.I. Tanner - Generation of Unstructured Tetrahedral Meshes by the Advancing Front Technique; *Int. J. Num. Meth. Eng.* 36, 1805-1823 (1993).

[Kam96] A. Kamoulakos, V. Chen, E. Mestreau and R. Löhner - Finite Element Modelling of Fluid/ Structure Interaction in Explosively Loaded Aircraft Fuselage Panels Using PAMSHOCK/ PAMFLOW Coupling; *Conf. on Spacecraft Structures, Materials and Mechanical Testing*, Noordwijk, The Netherlands, March (1996).

[Liu96] W.K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belytschko, C. Pan, R.A. Uras and C.T. Chang - Overview and Applications of the Reproducing Kernel Particle Methods; *Archives Comp. Meth. Eng.* 3(1), 3-80 (1996).

[Löh88a] R. Löhner - Some Useful Data Structures for the Generation of Unstructured Grids; *Comm. Appl. Num. Meth.* 4, 123-135 (1988).

[Löh88b] R. Löhner and P. Parikh - Three-Dimensional Grid Generation by the Advancing Front Method; *Int. J. Num. Meth. Fluids* 8, 1135-1149 (1988).

[Löh90] R. Löhner - Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing; *Computer Systems in Engineering* 1, 2-4, 257-272 (1990).

[Löh92] R. Löhner and J.D. Baum - Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems; *Int. J. Num. Meth. Fluids* 14, 1407-1419 (1992).

[Löh93] R. Löhner - Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations; *AIAA*-93-3348-CP (1993).

[Löh96a] R. Löhner - Extending the Range of Applicability and Automation of the Advancing Front Grid Generation Technique; *AIAA*-96-0033 (1996).

[Löh96b] R. Löhner - Regridding Surface Triangulations; *J. Comp. Phys.* 126, 1-10 (1996).

[Löh97] R. Löhner - Automatic Unstructured Grid Generators; *Finite Elements in Analysis and Design* 25, 111-134 (1997).

[Löh98a] R. Löhner and E. Oñate - An Advancing Point Grid Generation Technique; *Comm. Num. Meth. Eng.* 14, 1097-1108 (1998).

[Löh98b] R. Löhner, C. Yang, J. Cebral, J.D. Baum, H. Luo, D. Pelessone and C. Charman - Fluid-Structure-Thermal Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids; *AIAA*-98-2419 [Invited] (1998).

[Löh99] R. Löhner - Generation of Unstructured Grids Suitable for RANS Calculations; *AIAA*-99-0662 (1999).

[Löh01] R. Löhner - A Parallel Advancing Front Grid Generation Scheme; *Int. J. Num. Meth. Eng.* 51, 663-678 (2001).

[Mar95a] D.L. Marcum and N.P. Weatherill - Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection; *AIAA J.* 33, 9, 1619-1625 (1995).

[Mar95b] D.L. Marcum - Generation of Unstructured Grids for Viscous Flow Applications; *AIAA*-95-0212 (1995).

[Mes93] E. Mestreau, R. Löhner and S. Aita - TGV Tunnel-Entry Simulations Using a Finite Element Code with Automatic Remeshing; *AIAA*-93-0890 (1993).

[Mes96] E. Mestreau and R. Löhner - Airbag Simulation Using Fluid/Structure Coupling; *AIAA*-96-0798 (1996).

[Nay72] R.A. Nay and S. Utku - An Alternative for the Finite Element Method; *Variational Methods Eng.* 1 (1972).

[Oku96] T. Okusanya and J. Peraire - Parallel Unstructured Mesh Generation; *Proc. 5th Int. Conf. Num. Grid Generation in CFD and Related Fields*, Mississippi, April (1996).

[Oku97] T. Okusanya and J. Peraire - 3-D Parallel Unstructured Mesh Generation; *Proc. Joint ASME/ASCE/SES Summer Meeting* (1997).

[Oña96a] E. Oñate, S. Idelsohn, O.C. Zienkiewicz and R.L. Taylor - A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow; *Int. J. Num. Meth. Eng.* 39,3839-3866 (1996).

[Oña96b] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor and C. Sacco - A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems; *Comp. Meth. Appl. Mech. Eng.* 139, 315-346 (1996).

[Per87] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz - Adaptive Remeshing for Compressible Flow Computations; *J. Comp. Phys.* 72, 449-466 (1987).

[Per88] J. Peraire, J. Peiro, L. Formaggia K. Morgan and O.C. Zienkiewicz - Finite Element Euler Calculations in Three Dimensions; *Int. J. Num. Meth. Eng.* 26, 2135-2159 (1988).

[Per90] J. Peraire, K. Morgan and J. Peiro - Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD; *AGARD-CP-464*, 18 (1990).

[Per92] J. Peraire, K. Morgan, and J. Peiro - Adaptive Remeshing in 3-D; *J. Comp. Phys.* (1992).

[Per96] J. Peraire and K. Morgan - Unstructured Mesh Generation Including Directional Refinement for Aerodynamic Flow Simulation; *Proc. 5th Int. Conf. Num. Grid Generation in CFD and Related Fields*, Mississippi, April (1996).

[Pir94] S. Pirzadeh - Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method; *AIAA*-94-0417 (1994).

[Pir96] S. Pirzadeh - Progress Towards a User-Oriented Unstructured Viscous Grid Generator; *AIAA*-96-0031 (1996).

[Til02] R. Tilch and R. Löhner - Advances in Discrete Surface Grid Generation: Towards a Reliable Industrial Tool for CFD; *AIAA*-02-0862 (2002).

[Wea92] N.P. Weatherill - Delaunay Triangulation in Computational Fluid Dynamics; *Comp. Math. Appl.* 24, 5/6, 129-150 (1992).

[Wea94] N.P. Weatherill and O. Hassan - Efficient Three-Dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints; *Int. J. Num. Meth. Eng.* 37, 2005-2039 (1994).

Virtual Reality Techniques for Prototyping and Simulation in the Design Process

Pere Brunet
Universitat Politecnica de Catalunya, UPC
Barcelona, Spain

Abstract

Virtual Reality is emerging as a new tool for the visual simulation and inspection of complex geometric models. Virtual reality systems are powerful tools for the analysis of complex geometric 3D models. They support interactive navigation and walkthroughs, sensorial immersion and a transparent, implicit user interface. This talk presents the main features of Virtual Reality (VR) systems [1], the basic components and architectures and the best known applications
while focusing in virtual prototyping for industrial design applications.

The best known VR systems (Cave [2], StereoWall, PowerWall, workbench or VR theaters) are semi-immersive systems based on image projection. Their basic components include a tracking device, a simulation engine (hardware, 3D virtual model, navigation software and a graphics board), a projection system generating a different image for each eye of the user, and stereo glasses. Immersion is based on the automatic placement of the virtual cameras as a function of tracking information, and on the stereo effect. Haptic interaction requires real-time collision detection [3] and haptic output devices that give the user a perception of touch.

Virtual reality applications include training, planification and non-invasive diagnose in medical applications, cooperative inspection of complex or unreachable environments. In the Barcelona Virtual Reality Center (the center is a joint iniitative between UPC and the company Gedas Iberia [4]) we are mainly focusing on the development of new VR systems [5] and low-cost applications for industrial design, medical applications and architecture and urban planning. Focusing on industrial design, VR techniques allow the simulation, experimentation and inspection on virtual prototypes. Virtual prototypes are 3D models that can be inspected in a VR system with 3D immersion and real-time realistic visualization and interaction. Inspection of virtual prototypes facilitates the collaborative design analysis by teams including designers and end customers. The consequence is that design errors and problems can be detected in an early phase of the design workflow, being possible to make re-design decissions before the construction of the first phisical prototypes.

The future of Virtual Reality systems offers a huge number of possibilities. In the next few years, VR systems will become more portable and immersive, and will be usable in a number of new applications like guidance or interactive basic education. New and better implicit and haptic interaction devices, light and portable semi-immersive glasses and advanced cooperation tools among distributed VR systems will lead to dramatic changes in the design workflow. The consequence is a quality increase with reduced design time and better adaptation to the end user requirements.

References:

[1] A. Rowell, "Virtual Reality", Computer Graphics World, pp 21-28, February 1997.

[2] C. Cruz-Neira, D.J. Sandin, T.A. De Fanti, "Surround-screen Projection-based Virtual Reality: the Design and Implementation of the Cave", ACM Computer Graphics, Proc. of Siggraph'93, pp 135-142 (1993)

[3] D. Bowman, L. Hodges, "Formalizing the Design, Evaluation and Application of Interaction Techniques for Immersive Virtual Environments", The Journal of Visual Languages and Computing, Vol. 10 (1), pp 37-53 (1999).

[4] CRV, UPC-Gedas Virtual Reality Center: www.lsi.upc.es/dept/crv

[5] C. Andujar, M. Fairen, P. Brunet, V. Cebollada, R. Trueba, "Sistema de proyeccion inmersivo para la interaccion con modelos 3D", Proc. of IPO'02, Madrid (2002).

THE GENERATION OF UNSTRUCTURED BOUNDARY-CONFORMING MESHES OF HIGH-ORDER ELEMENTS

Joaquim Peiró, Spencer J. Sherwin and Sergio Giordana

Department of Aeronautics, Imperial College of Science, Technology and Medicine, Prince Consort Road, London SW7 2BY, United Kingdom

SUMMARY: The ability to construct suitable computational meshes is currently a significant limiting factor in the development of compact high-order algorithms, such as spectral elements or *p*-type finite elements, in very complex geometries. This article discusses the generation of 3D unstructured meshes of high-order elements that conform to the boundary of the computational domain and presents strategies to alleviate the problem of generation of distorted elements with singular elemental mappings.

KEYWORDS: High-order mesh generation, hybrid meshing, curvature based adaption, high-order elements.

INTRODUCTION

High-order unstructured algorithms, such as spectral elements [8] or *p*-type finite elements [9], offer the potential of high accuracy if the solution is smooth and a well behaved mapping exists between the elements in the mesh and a standard region. The extension of high-order algorithms to three-dimensional problems has already been addressed in these references, but their implementation within three-dimensional mesh generation and CAD surface representation techniques has received little attention. The development of high-order algorithms in very complex geometries is currently limited by progress in methods for high-order mesh generation.

However the extension of standard unstructured mesh generation technology to high-order algorithms is not a trivial exercise. Complications arise due to the conflicting requirements to generate coarse meshes whilst maintaining good elemental properties in regions of high curvature. For instance, Fig. 1 shows a valid discretization of the computational domain using linear elements, but the corresponding high-order discretization is unacceptable since it contains elements with singular local to global mappings.

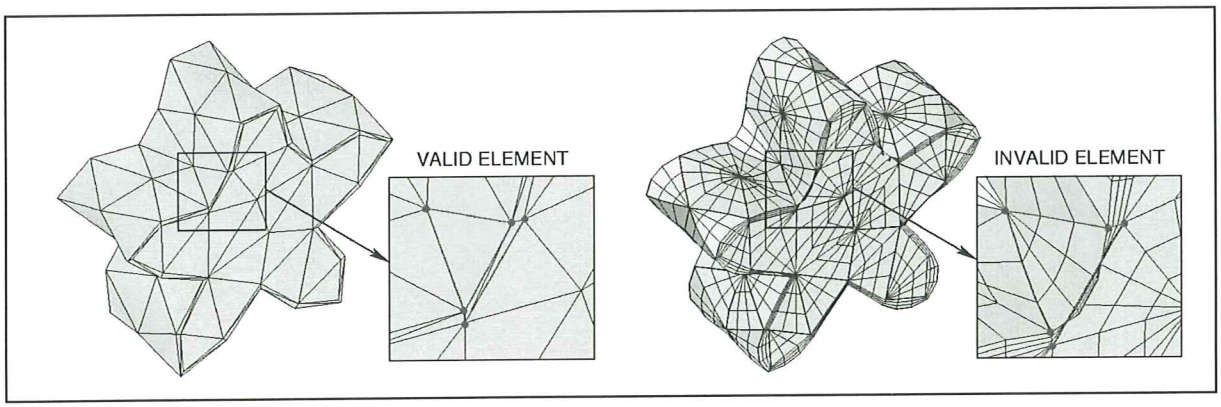


Fig. 1: The introduction of extra degrees of freedom in a valid mesh of linear elements (left) might lead to the appearance of invalid high-order elements (right).

In this paper we will discuss the method adopted for the generation of meshes of high-order elements with particular emphasis on the optimisation of the surface mesh generation, and will present techniques for avoiding the appearance of singular elements by refining the mesh according to the surface curvature. We will illustrate these techniques using the geometry depicted in Fig. 1. This is a cylinder of unit depth with a cross section generated as a sinusoidal perturbation of amplitude 0.8 to a circle of unit radius.

MESH GENERATION OF HIGH-ORDER ELEMENTS

The method adopted to generate an unstructured mesh of spectral/*hp* elements, as originally presented in [5], is based on the post-processing of an unstructured mesh of linear elements and proceeds in three steps. The first step is to generate a discretisation of the boundary of the domain into triangular surface elements. The second step involves the transformation of this surface triangulation into high-order elements suitable for a spectral/*hp* computation. Finally the interior volume is constructed using standard low-order mesh technology although more advanced techniques may generally be necessary as discussed in Dey et al. [1]. These steps are schematically represented in Fig. 2.

The geometry of the computational domain is defined through a boundary representation (B-Rep) where the domain is viewed as the interior region to a boundary composed by a set of faces on surfaces intersecting along curves where the edges of the boundary lie. This is shown in Fig. 2a. These curves and surfaces are described using standard techniques of computer-aided design (CAD) representation in terms of parametric curves and surfaces. Alternative representations such as implicit surfaces are currently being used in geometry reconstruction from images, particularly in medical applications. The application of the techniques presented here to implicit surfaces is discussed in reference [7].

The first stage of the high-order meshing is to generate a coarse linear surface triangulation, depicted in Fig. 2b, and then to modify these sub-domains into a *boundary conforming* mesh of high-order elements, shown in Fig. 2c, by splitting the sides and faces in a bottom-up fashion consistent with a B-Rep of the computational domain.

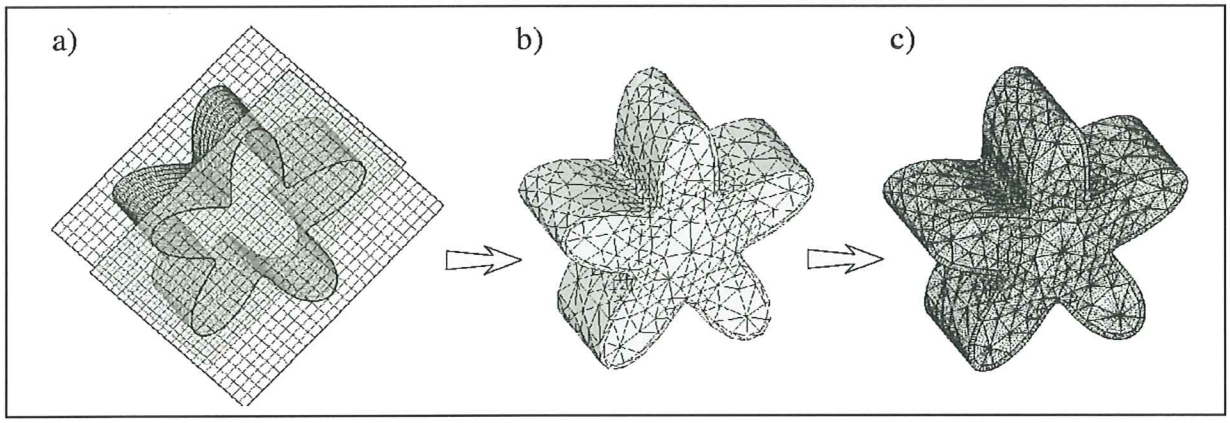


Fig. 2: Generation of a mesh of high-order elements: a) CAD boundary representation (B-Rep) of the domain, b) unstructured mesh of linear elements, and c) unstructured high-order mesh.

The generation of linear elements is achieved by using a modified advancing layers method in the near-wall regions and an advancing front technique in the rest of the domain as described in [6]. The meshes generated in this manner are employed in the modelling of viscous flows containing a boundary layer region near solid boundaries where the flow solution varies very rapidly. An unstructured boundary layer mesh can be constructed by extruding the surface triangulation in the wall normal direction. This produces a series of prismatic regions that can be subdivided into three tetrahedral elements. Such an approach is valid for meshes of linear elements and for high-order elements provided that the local surface is concave. However, in a locally convex region, the division of the prismatic regions into tetrahedral elements can lead to self intersecting elements as shown in Fig. 1. The obvious alternative is not to subdivide the prismatic region but apply a hybrid expansion. The computational domain then consists of an outer shell of prismatic regions surrounding an inner core of straight-sided tetrahedral elements as shown in Fig. 2b and 2c. Since the inner tetrahedral domain contains only straight-sided elements, the sub-domains are guaranteed to be valid if a valid low-order mesh has been constructed.

The generation of high-order elements therefore starts with a discretisation of the sides of the triangular surface elements into P points as required by a polynomial interpolation of degree $P-1$. There are two possible cases to consider. If the side belongs to a curve of the B-Rep, the intermediate $P-2$ points are placed along the length of the CAD curve where the edge lies. If the side is on a face of the B-Rep, then the intermediate points are positioned on the corresponding CAD surface. In both of these cases a point placement algorithm is required either within a parametric curve or surface. It is the generation of these points which we refer to as high-order mesh generation since this information is used to reconstruct a high-order numerical representation of the computational domain.

At present we complete the volume generation of linear elements using the modified advancing layer combined with an advancing front technique. The higher order surface information and the ability to deform element internally could be utilized during the process, as proposed in [1], but this has not been necessary here.

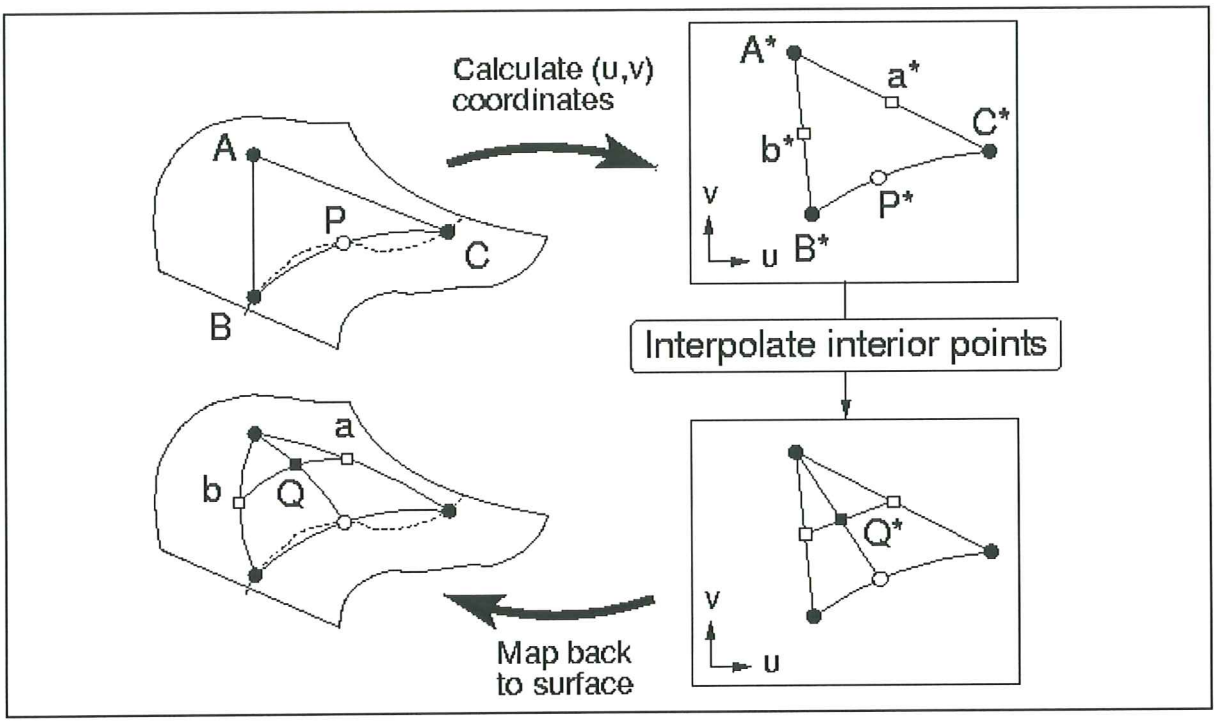


Fig. 3: Surface mesh subdivision procedure.

An important part of the high-order mesh generation process is the construction of high-order edges and faces on the CAD curves and surfaces of the B-Rep of the computational domain. To illustrate this process we consider the diagrammatic representation shown in Fig. 3. The starting point for the generation of a high-order representation of an elemental face belonging to a boundary is the definition of three vertices on a CAD surface. If an edge of the face lies on a CAD curve, its discretisation requires the placement of points along the CAD curve. Typically, a distribution based on a Gaussian quadrature is used. The shape of the other edges can be freely defined within the CAD surface. A computationally convenient approach is to use a linear interpolation between the parametric coordinates of the end points of the edge. The points in the interior of the face can be determined in a similar fashion by using, for example, a transfinite interpolation within the parametric plane. Finally the three-dimensional coordinates of the boundary points of the elemental face are obtained by mapping the calculated parametric coordinates onto the CAD surface.

The use of a transfinite interpolation between the discretised edges of triangles in the parametric space followed by a projection onto the surface does not take into account the metric information of the mapping between the physical and parametric space. For CAD surfaces represented by isometric mappings (i.e. those that preserve lengths) the above procedure produces reasonably good quality elements. In regions where the mapping is highly anisometric, such an approach can lead to very deformed mappings or, at worst, invalid elements. In the following section we propose a method to obtain an optimal distribution of mesh points through the minimisation of an appropriate function that represents the energy of the system.

OPTIMISING THE SURFACE APPROXIMATION

To address the problem of obtaining an optimal distribution of points, let us consider a quadrature with P integration points and associated normalised weights z_i; $i=1,\ldots,P$ $(-1 \le z_i \le 1)$ in a one-dimensional interval $a \le x \le b$. It is known that the optimal positions x_i; $i=1,\ldots,P$ of the points are given by

$$x_i = a\left(\frac{1-z}{2}\right) + b\left(\frac{1+z}{2}\right); \qquad i=1,\ldots,P \tag{1}$$

which lead to an isometric mapping and therefore a constant Jacobian. An optimal distribution of integration points for curves is easily obtained by interpreting the coordinate x as the arc length of the curve. The implementation of this criterion using parametric CAD curves is relatively straightforward and amounts to finding the coordinates of a point in the curve that corresponds to the required length.

This procedure can be extended to elements with straight sides and faces by using a tensor-product of optimal distributions along coordinate lines. However, a different strategy is required to achieve such an optimal distribution for *curved* edges and faces as we need to account for the distortion introduced by the presence of curvature.

A parametric surface is defined as a transformation between a two-dimensional parametric space (u,v) and a three-dimensional space $\mathbf{r}(u,v) = [x(u,v),\ y(u,v),\ z(u,v)]$. Mesh generation is considerably simplified if performed in a parametric space. Therefore, as previously discussed, a computationally straightforward approach, illustrated in Fig. 3, is to approximate the element edge by a straight line in the parametric plane and to distribute the points along that line according to equation (1). A similar approach has been adopted in the work of Dey et al. [1]. However, as discussed previously, if the transformation $\mathbf{r}(u,v)$ is anisometric, this approach can lead to highly deformed or invalid elemental regions.

Optimal distribution of edge points

A more general procedure can be obtained by reformulating the problem of finding the optimal distribution of points as that of minimising the potential energy of a set of springs linking adjacent points. If the stiffness of a spring joining nodes i and $i+1$ is taken to be inversely proportional to the weight increment $z_{i+1} - z_i$, it is easily shown that the optimal distribution (1) is a minimum of the potential energy, denoted here by $\Im$, of such system of springs given by

$$\Im(x_2,\ldots,x_{P-1}) = \sum_{i=1}^{P-1} \frac{(x_{i+1} - x_i)^2}{z_{i+1} - z_i}. \tag{2}$$

This approach, unlike equation (1), is directly applicable to curved edges and faces on surfaces. The implementation of this approach to edges on a parametric surface is slightly more involved. Since our parametric surface is defined in terms of the parametric coordinates $\mathbf{u} = (u,v)$, we recast the cost function as

$$\Im_e(\mathbf{u}_2,\ldots,\mathbf{u}_{P-1}) = \sum_{i=1}^{P-1} \frac{\left\| \mathbf{r}(u_{i+1},v_{i+1}) - \mathbf{r}(u_i,v_i) \right\|^2}{z_{i+1} - z_i}. \tag{3}$$

In this two-dimensional minimisation procedure, the end points of the edge are fixed and the curve defining the edge is free to move in the parametric surface. This minimisation procedure is also *non-linear* since it involves the mapping $\mathbf{r}(u,v)$ between the physical coordinates and the parametric plane. The main advantage of this approach with respect to the transfinite interpolation is that the optimal solution will tend to a geodesic of the surface between the two end points and the distribution of points along the geodesic will also tend towards the optimal quadrature distribution.

The cost function (3) can also be easily adapted to deal with the discretization of curved edges defined as a function of a single parameter u. The minimisation clearly does not alter the length of the curve and it will position the points in accordance with the quadrature distribution (1).

Optimal distribution of face points

The spring network analogy can also be applied to find the location of interior points within a curved elemental face. Assuming that a good distribution of points is known for a planar straight sided triangle, we can ask what is the set of stiffnesses k in a network of springs that will produce the same distribution of points as its equilibrium position?

Here we assume a tensor-product distribution of quadrature points with P points in each direction. The positions of the quadrature points in the parametric space are denoted by $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})$; $i,j = 1,\ldots,P$ and the optimal distribution of the interior points is obtained by minimizing the potential energy of the system defined as

$$\Im_f(\mathbf{u}_{2,2},\ldots,\mathbf{u}_{i,j},\ldots,\mathbf{u}_{P-1,P-1}) = \sum_{j=2}^{P-1}\sum_{i=1}^{P-1}\frac{\left\| \mathbf{r}(u_{i+1,j},v_{i+1,j}) - \mathbf{r}(u_{i,j},v_{i,j}) \right\|^2}{z^{(1)}_{i+1} - z^{(1)}_{i}} + \sum_{i=2}^{P-1}\sum_{j=1}^{P-1}\frac{\left\| \mathbf{r}(u_{i,j+1},v_{i,j+1}) - \mathbf{r}(u_{i,j},v_{i,j}) \right\|^2}{z^{(2)}_{j+1} - z^{(2)}_{j}} . \quad (4)$$

The symbols $z^{(1)}_i$ and $z^{(2)}_j$ denote the quadrature weights in the i and j directions respectively. A slightly modified version of this method could also be applied to the problem of finding the optimal position of the interior points within a curved element. Although we have assumed a tensor-product distribution of the points defining the polynomial interpolation within the element, any distribution of points can be minimised in this fashion by an appropriate choice of the spring network. The reader is referred to reference [8] for further details.

To illustrate the surface mesh generation methodology, we consider the generation of a tetrahedral p-type mesh of fifth order polynomial within a simple cubic computational domain $0 \le x,y,z \le 10$. The faces of the cube are located on tensor-product surfaces defined by an anisometric mapping shown in Fig. 4a). The spacing varies linearly with a value of $\delta = 0.1$ at the boundary and $\delta = 2.1$ at the centre of the faces. The anisometry of the mapping is due to the fact that the unevenly spaced network of lines depicted in Fig. 4a) is obtained as the image of a network of coordinate lines $u = const.$ and $v = const.$ in the parametric plane which are uniformly spaced with $\Delta u = \Delta v = 1$ as depicted in Fig. 4d. The curve definition of the edges representing the intersection between each of the faces was taken to be isometric.

A standard *h*-type unstructured mesh generation process was used to construct a coarse mesh with 66 elements. Although the *p*-type elements can easily be constructed on a planar surface by a linear interpolation between the vertices, we have chosen to reconstruct the surface elements using the parametric definition of the surface as required for a non-planar surface. Therefore we know that the optimal solution for this geometry is a linear distribution of points between the vertices according to the quadrature distribution given by equation (1).

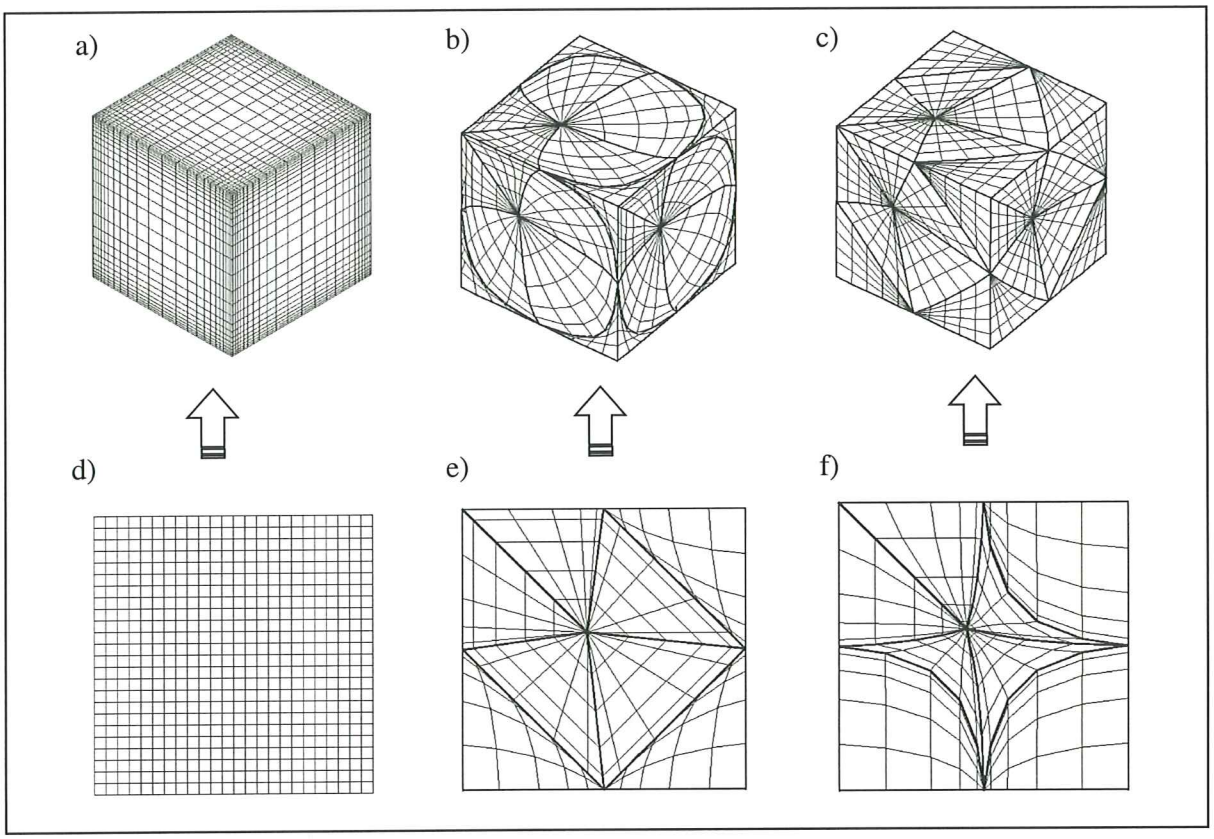


Fig. 4: Influence of surface mapping on the *p*-type generation procedure. a) Anisometric CAD description. The corresponding representation in the parametric space is depicted in Fig. 4d. b) High-order mesh using a transfinite interpolation in the parametric space represented in Fig. 4e. c) High-order mesh using an optimised point placement in the parametric space as shown in Fig. 4f.

However if we apply the optimisation procedure based on the spring analogy, we obtain the surface mesh shown in Fig. 4c which produces a fit very close to a linear mapping between the vertices. A natural consequence of obtaining a good physical surface distribution is that the parametric distribution becomes very distorted as shown in figure Fig. 4f. Mesh distortion in the physical space has a strong influence in the accuracy of the numerical solution. Results presented in reference [8] for an elliptic problem have shown that the optimized method leads to improved rates of convergence, as the mesh size is decreased, with respect to the transfinite interpolation in the parametric space.

CURVATURE BASED DISCRETIZATION

Curvature based refinement in which the mesh size is obtained as a function of the curvature has been proposed, see for instance [3], as a way to obtain an accurate piecewise linear approximation of a curved surface. Following the notation of Fig. 5a, the curve is locally

approximated by a circle of radius R, the radius of curvature. We assume that the mesh spacing can be represented by a chord of length c in the circle. Denoting the maximum distance between the chord and the circle by t, we restrict its value, as suggested in [3], to $t \le \varepsilon R$, where ε is a user defined tolerance. The mesh spacing, δ, can now be obtained as a function of R and ε as

$$\delta \approx c \le 2R\sqrt{\varepsilon(2-\varepsilon)} \tag{5}$$

As a guide, a value of $\varepsilon = 0.01$ results in the division of a circle into 26 segments.

The use of this technique enhances the quality of the high-order meshes generated from linear tetrahedral and prismatic meshes [8]. However, this criterion on its own is not sufficient to guarantee validity of all high-order elements as it does not account for the possible intersection of the boundary sides and faces with those on the interior. Here we propose an alternative method more suitable for the discretization of boundary layers.

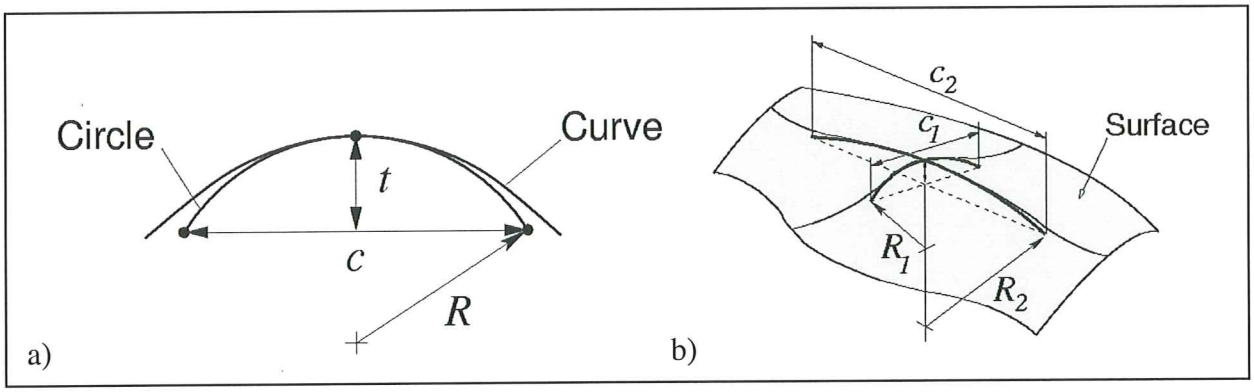


Fig. 5: Notation for curvature calculation: a) curves, and b) surfaces.

In the modelling of viscous flows, the value of δ is usually prescribed to achieve a certain boundary layer resolution. Following the notation in Fig. 6, the value of c is therefore chosen to guarantee that the osculating circle representing the curve does not intersect the interior sides of the elements, i.e. $\theta \le 90^o$ for the triangular element in Fig. 6a. The value of c, which should be considered as a maximum mesh spacing, can now be obtained as a function of R and δ. Its value c_t for triangular elements is

$$c_t \le R\sqrt{\frac{2\delta}{R+\delta}} \tag{6}$$

and the corresponding value c_q for quadrilateral elements is

$$c_q \le \frac{2R\delta}{R+\delta}\sqrt{1+\frac{2R}{\delta}}, \tag{7}$$

where the boundary displacement is assumed to be the same on either side of the rectangle. It is interesting to notice that, for a given δ, the quadratic element allows for a mesh spacing c_q which is about twice the value of spacing c_t for the triangular element.

The extension of the curvature refinement method to surfaces is straightforward. The refinement criteria given by formulas (5), (6) and (7) is used for the two principal directions and the corresponding mesh spacings, c_1 and c_2 in figure Fig. 5b, are calculated from the values of the principal curvatures R_1 and R_2. Expressions for the curvature of curves and surfaces can be found in reference [2].

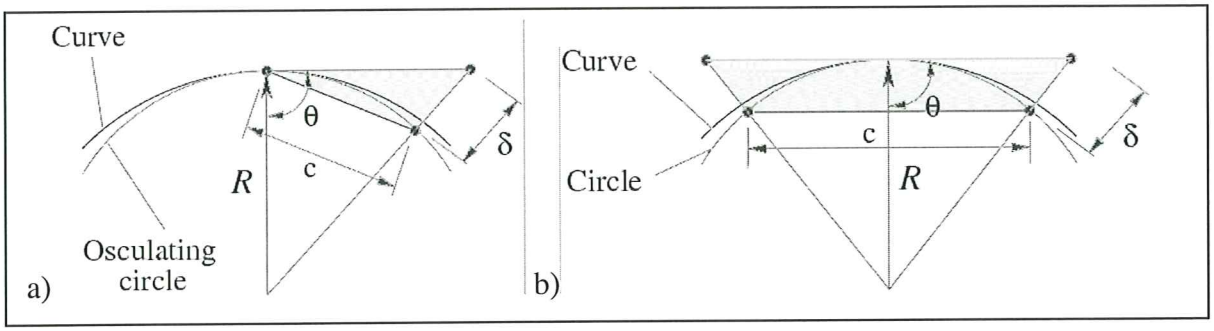


Fig. 6: Notation for mesh spacing calculation on curves: a) triangular element and b) quadrilateral element.

An example of a hybrid mesh generated for the geometry previously considered in Fig.1 and using the criterion given by equation (5), is shown in Fig. 7a. The value of ε in equation (5) has been chosen to be the largest one that produces a high-order mesh without singular elements. The final mesh contains 2388 high-order elements with fifth order ($P=5$) polynomial interpolation. Fig. 7b shows the surface plot variation of the quality of the mesh represented by the index Q_{3D} defined as the ratio between the global minimum and maximum values of the Jacobian determinant of the mapping, J_q, over the discrete set of quadrature points $q=1,\ldots,N_q$, this is

$$Q_{3D}=\frac{\min(J_q)}{\max(J_q)}, \qquad q=1,\ldots,N_q. \tag{8}$$

This mesh has a minimum value of $Q_{3D}=0.15$ corresponding to a factor of 7 variation in the elemental Jacobian.

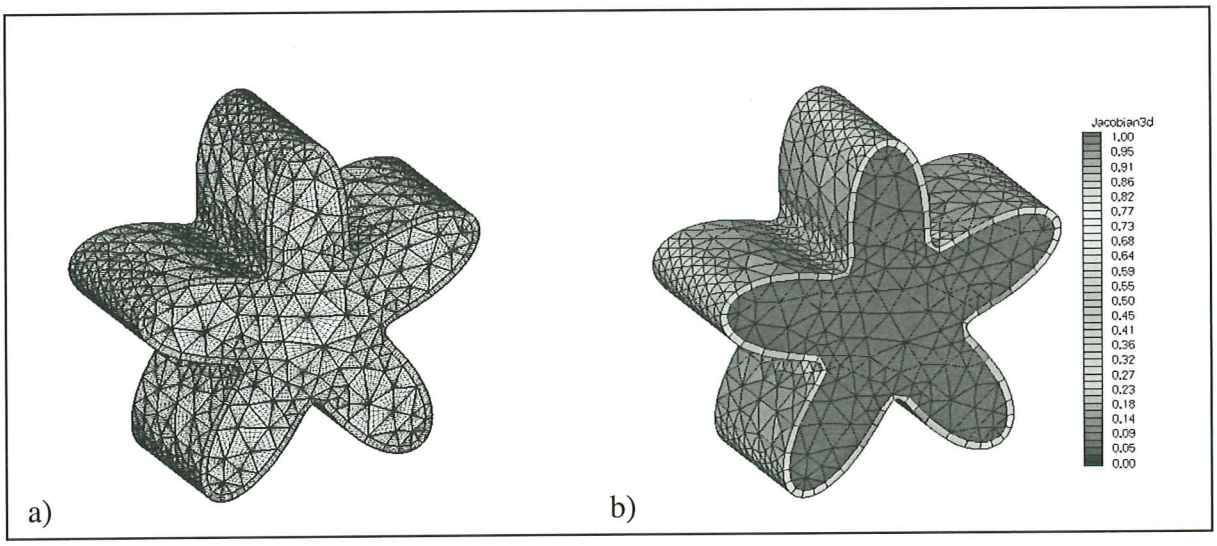


Fig. 7: Curvature based mesh refinement for prismatic elements according to equation (5): a) high-order mesh, and b) distribution of the mesh quality index Q_{3D}.

The application of the refinement criterion (7) to the same geometry leads to the mesh and distribution of quality index depicted in Fig. 8. The mesh contains now 1692 elements with a larger number of elements with an index Q_{3D} close to one than the previous mesh (Fig. 7).

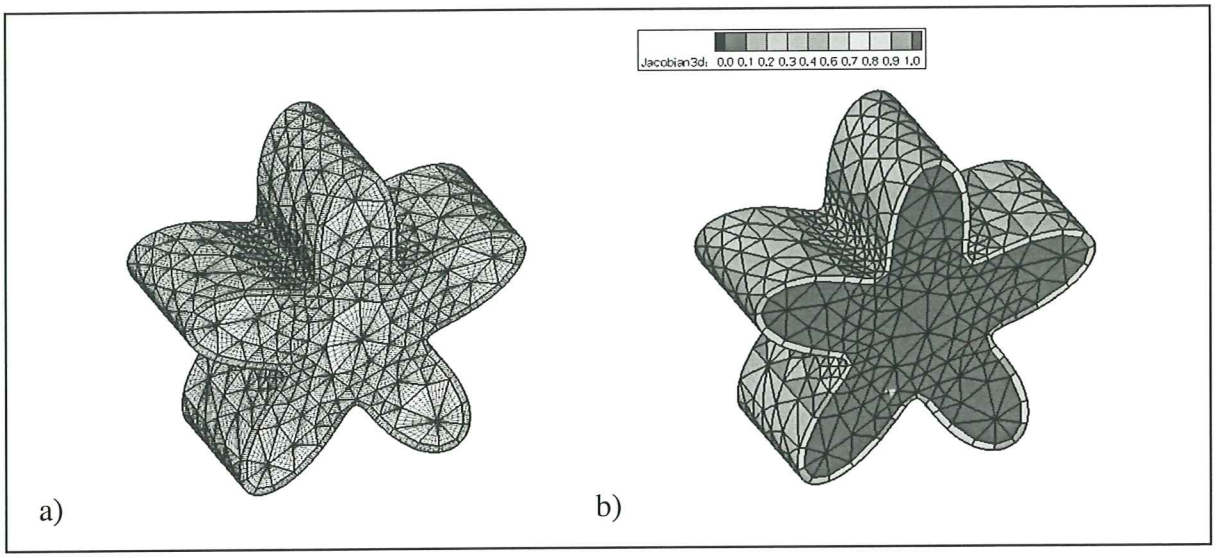


Fig. 8: Curvature based mesh refinement for prismatic elements according to equation (7): a) high-order mesh, and b) distribution of mesh quality index Q_{3D}.

However, the refinement applied here does not account for the sign of the surface curvature and these curvature based criteria are too restrictive to ensure element validity in those regions where the domain is locally convex. For a *convex* region, the less restrictive criterion

$$\delta \leq R \tag{9}$$

suffices to guarantee element validity. This is highlighted in Fig. 9 where the refinement criterion (7) has been selectively applied to concave regions only. The result is a valid mesh with fewer elements (1424) and a similar distribution of the quality index Q_{3D}.

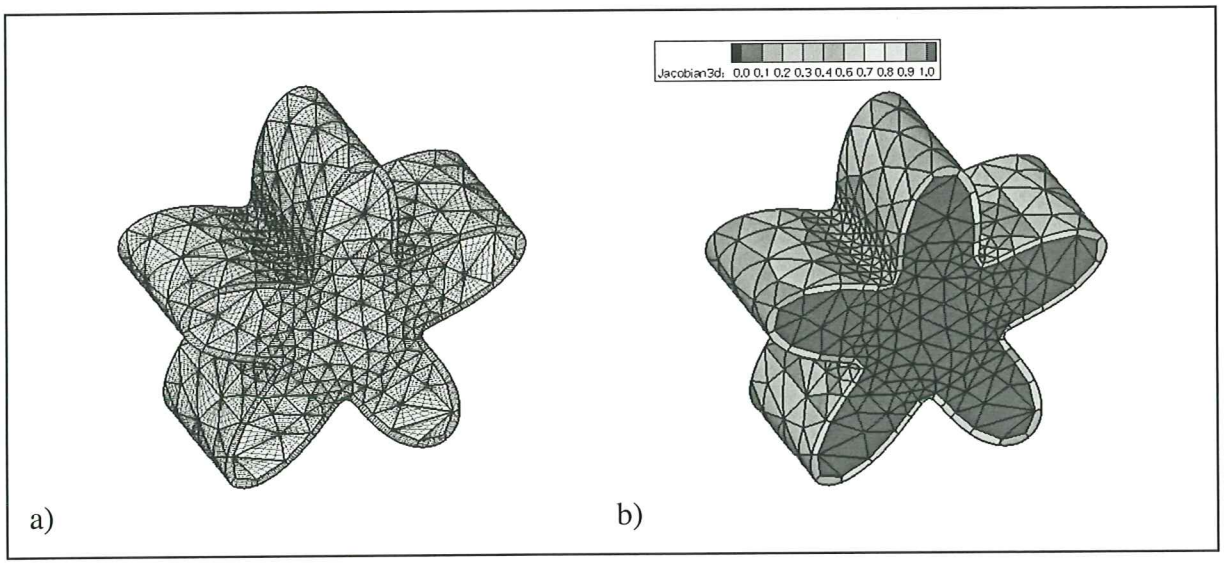


Fig. 9: Selective curvature refinement: The criteria (7) is applied to convex regions. Concave regions are applied the less restrictive requirement (9).

CONCLUSIONS

In this paper we have presented a set of strategies to generate unstructured meshes of valid high-order elements. These represent a step forward towards the elusive target of consistently generating valid computational meshes for complex geometries without user intervention. The starting point is a coarse mesh of linear elements that can be generated by traditional unstructured mesh generation techniques. However, the subdivision of this mesh into a boundary conforming mesh of high-order elements requires careful handling due to the distortion that non-planar surfaces might introduce.

We have presented methods for generating a hybrid discretisation using prismatic and tetrahedral elements, together with an approach for generating optimal surface discretisations for high-order elements that accounts for surface curvature. This generation process minimises the element distortion due to the surface curvature based refinement by searching for the shortest geodesic line between two points whilst maintaining a Gaussian quadrature distribution. Finally the use of curvature-based refinement has proven to be a very useful tool to produce valid high-order meshes with fewer elements.

ACKNOWLEDGEMENTS

The authors would like to acknowledge support from the Smiths' Charity, the Bupa foundation, the Heartbeat charity and the Clothworkers' Foundation. The Imperial College centres for Biomedical Visualization and Parallel Computing provided computational resources.

REFERENCES

1. Dey, S., Shephard, M.S. and Flaherty, J.E., "Geometry representation issues associated with *p*-version finite element computations", *Comp. Meth. Appl. Mech. Engng.*, **150**, 1997, pp. 39-55.
2. Do Carmo, M.P., "Differential Geometry of Curves and Surfaces", Prentice-Hall, 1976.
3. P. Frey and P-L. George, "Maillages", Editions Hermes, 1999.
4. Karniadakis, G.E. and Sherwin, S.J., "Spectral/*hp* Element Methods for CFD", Oxford University Press, 1999.
5. Peiró, J., Shah, O., Doorly, D. J., Sherwin, S. J. and Caro C.G., "Unstructured Viscous Mesh Generation for Haemodynamic Flow Simulation Using High-Order Spectral Elements", *Proc. 6th International Conference on Numerical Grid Generation in Computational Field Simulations,* Greenwich, U.K., 1998, pp. 457-466.
6. Peiró, J. and Sayma, A.I., "A 3-D Unstructured Multigrid Navier-Stokes Solver", *Numerical Methods for Fluid Dynamics V*, Morton, K.W. and Baines, M.J., Eds., Oxford University Press, 1995.
7. Peiró, J., Giordana, S., Griffith, C. and Sherwin, S.J., "High-Order Algorithms for Vascular Flow Modelling", Accepted for publication in *Int. J. Numer. Meth. Fluids*, 2002.
8. Sherwin, S.J., and Peiró, J., "Mesh Generation in Curvilinear Domains Using High-Order Elements", *Int. J. Numer. Meth. Engng.*, **53**, 2002, pp. 207-223.
9. Szabó, B. and Babuška, I., "Finite Element Analysis", Wiley, 1991.

Web computing: new generation of remote analysis

Ramon Ribó Rodríguez
Compass Ingeniería y Sistemas
Barcelona, Spain
info@compassis.com
www.compassis.com

Abstract

One computer simulation analysis can be subdivided in three parts: preprocessing, analysis and postprocessing. We call Web computing to the approach of making the pre and postprocessing part in the local computer and making the analysis part in a remote computer. This separation between the two parts must be totally or semi-transparent to the user.
The main advantages that web computing can provide are:

- Give a service to company costumers by providing them analysis capabilities in a rental basis
- Permmits to run models that require a lot of computer resources in a dedicated server
- It is possible to centralize the analysis in one server inside a Company intranet

It is possible to use GiD [1], to perform remote analysis. One example of remote analysis server can be found in [2].

References

[1] *GiD website*: http://gid.cimne.upc.es. GiD program is available to download. Reference manuals and technical information of this program are also available.

[2] *Compass website*: http://www.compassis.com. Tdyn and RamSeries programs are available to download. Reference manuals and technical information of these programs are also available.

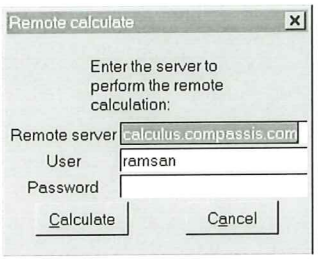
Introduction

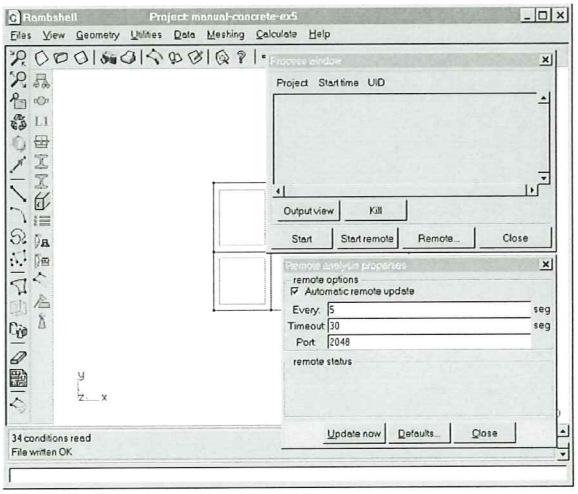
- Web computing in the 90's

  The concept of executing the analysis in a different computer than the pre/postprocessing is not new. The difference com in the fact of making this transparent to the user.
- Hardware & software requirements
- Commercial issues
  - License price
  - Software maintenance and upgrade
- Intranet solutions

The user interface

Remote analysis must be as simple for the user as running in the local computer.



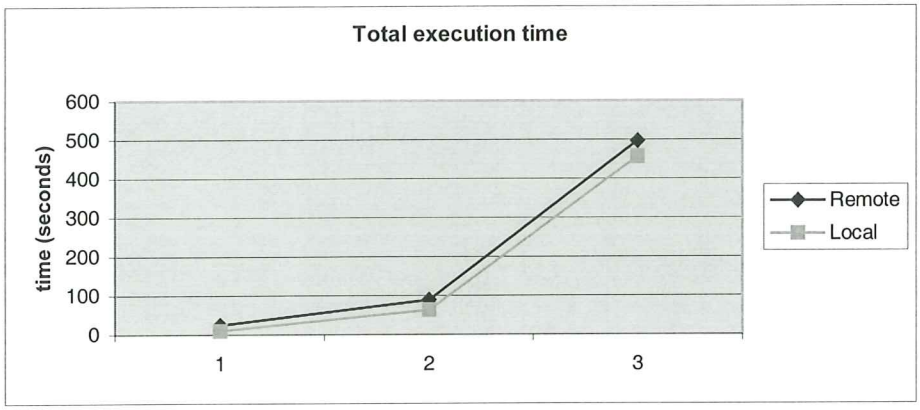


The communication

- Standard TCP-IP over Internet
- Can be used from:
  - Modem connection
  - ADSL connection
  - Higher speeds
- Connection issues
  - Speed
  - Data compression
  - Timeouts

Execution time

If we consider as execution time the total time since we send the process until it finishes, the time spend in sending the data is small compared to the total.



Difference of execution times using an ADSL telephone line

Security

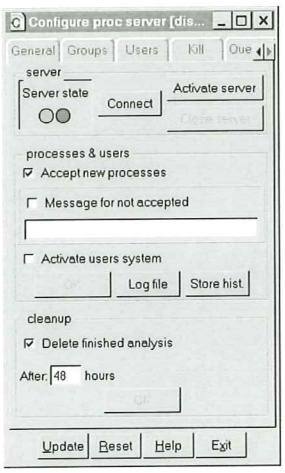
- Security of the server agains attacks
- Encryption
- Tunneling
- SSH
- Different security levels
    - No users
    - Users & passwords

UNIX advantages for the server

- Better load balance
- Daemon concept (versus Winnt services)
- Takes advantages of UNIX queues
- Additional security given by the general UNIX users & permissions
- Higher reliability
- Disadvantages
    - It is necessary to have an executable version of the analysis codes for that UNIX & hardware platform
    - Programs are not binary compatible between different hardware platforms in UNIX

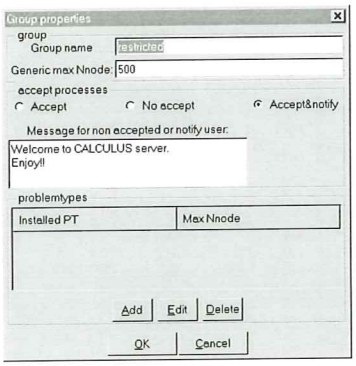
General configuration

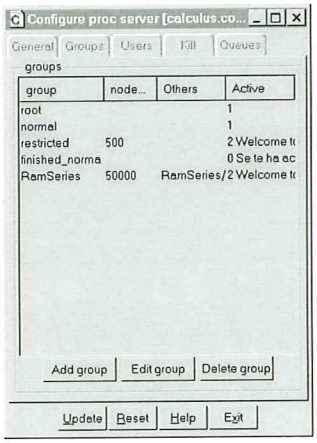
- The server can be configured from any computer
- Also from a web interface



General configuration: Groups

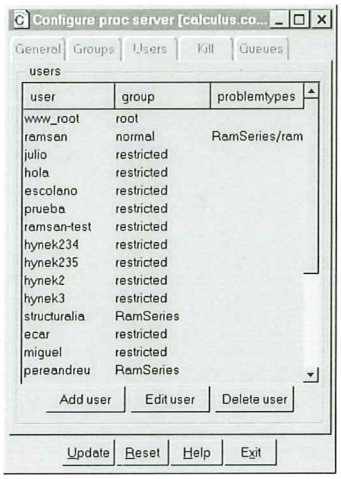
- Allows to have different type of users with different access permissions

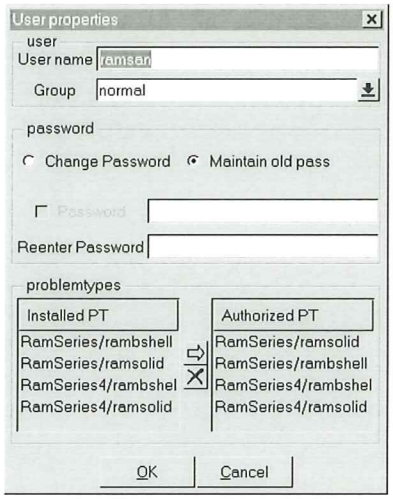




General configuration: Users

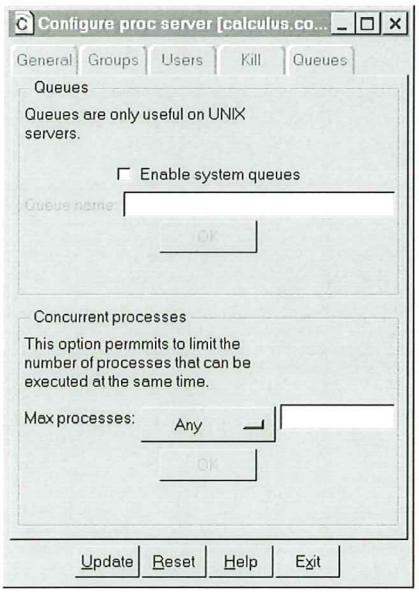
- User control access system
- Similar to OS level acces system
- It is possible to control permissions on a per user basis



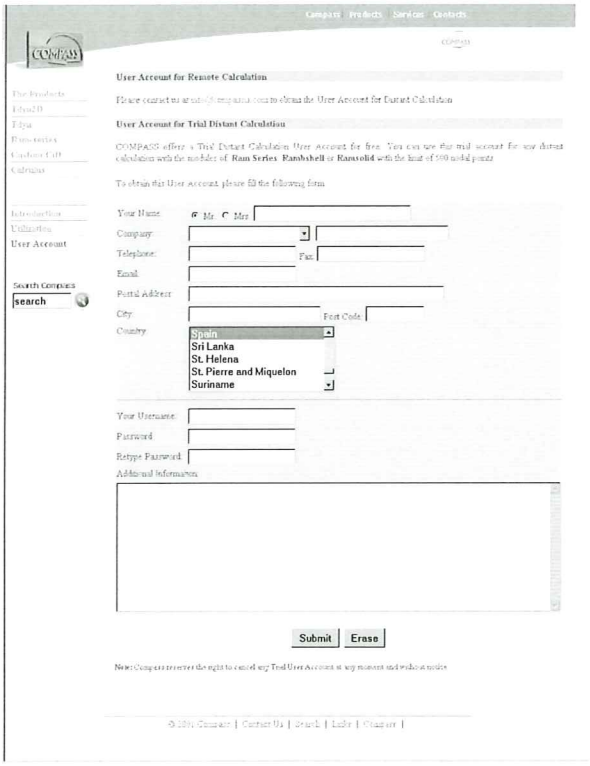


General configuration: Queues

- Takes advantage of UNIX queues, if present
- Alternatively, permmits to limit the number of concurrent processes



Web configuration interfase



Conclusions

- It is technically feasible with the current network capabilities
- Security issues should be more elaborated
- Advantages
  - Concept of "pay per use"
  - Avoids problems of version updates
  - Constant communication costumer-provider
  - Very convenient central intranet services for big companies
  - Specially suited to execute processes that demand huge computer resources
- Disadvantages
  - Common personal computers are every day more powerful
  - A new commercial service, like this, needs time to get accepted by the costumers

Application of GID to microwave printed circuits and antennas design

Ferdinando Tiezzi and Juan R. Mosig

École Polytechnique Fédérale de Lausanne (EPFL),
Laboratoire d'Électromagnétisme et Acoustique (LEMA),
EPFL, DE-ELB, LEMA, CH-1015 Lausanne, Switzerland.
Web Address: http://lemawww.epfl.ch/

Abstract *– In this paper, we present the application of GID to the design of multilayered planar printed circuits and antennas for microwave applications. In this context GID is used as main preprocessing tool, i.e. to build the geometry, to assign condition to some elements, to mesh the geometry and to generate the input for the analysis tools. To tailor GID to the special requirements of the printed circuit world, a new problem type has been developed that make the design of printed circuits more user-friendly and closer to the existing commercial tools dedicated to this task. This new environment has also been integrated inside the framework MADS (Multipurpose Antenna Design Simulator). All the developments have been realised in the frame of an European EU-ESPRIT project.*

INTRODUCTION

While microwave printed circuits are difficult to analyse, their geometrical structure is relatively simple. Microwave printed circuits are made by one, or more, dielectric layers (often called "substrates"). These substrates have basically the same structure of the standard boards used in electronic applications: a large, but thin dielectric slab with one or both faces covered by a thin film of copper where the circuit will be printed out. This explains why a lot of sophisticated methods for numerical analysis have been developed in the past and new ones are still under development, but also why there are so few developments concerning the pre-processing, design and meshing. The situation is very far from what happens for example in the mechanical field where a lot of commercial general-purpose tools, like GID, were developed to help the engineers to design and mesh the structures in a very efficient way. In fact standard CAD environments are often too heavy and not well suited for microwave printed circuits design and the present situation is that the few existing commercial microwave CAD softwares have developed their own dedicated GUIs and meshers, while at research level people still continue to use home made user interfaces and semi-automatic meshing.

This happens because the design of microwave circuits presents some particularities that it is worth to explain. While at low frequency the substrate acts only as hard support for the printed strips and for the electronic components, when the frequency increase the circuit performances are strongly affected by the substrate characteristics. In particular the response is affected by the electric permittivity, the magnetic permeability and the substrate thickness, but the effects due to the transversal size of the substrate are negligible, at least in typical applications.

From the previous observations, some methods of numerical analysis have been developed (essentially equivalent to Boundary Element Methods), which consider the substrates as infinite and take into account their effect analytically. In particular the method used in our application is based on a surface integral equation formulation of the electromagnetic problem [1, 2]. This equation contains special Green's Functions that take into account all the effects due to the presence of the substrates [3] and the only unknown is the current density J in the metallic strips. The Method of Moments (MoM) is applied to this equation, and in particular to the unknown current density, to obtain a system of equation that will be solved numerically. This means that with respect to methods almost purely numerical, like finite differences, where the entire 3D space must be discretised, in this case only the copper strips (that can be considered practically 2D) must be meshed. On the other side the dimensions of the strips affect barely the performance of the circuit and their design must be realised with an extreme precision. Moreover, the shapes of the circuits can look sometimes complicate, but most of the time a big circuit can be reduced to a series of basic shapes connected between them. Such shapes can be described very easily and effectively through a parametric representation.

All what said until now shows clearly how the design of printed circuits can be a quite easy task whenever a dedicated drawing program is available, but that at the same time the requirements are often hard to be satisfied by standard CAD environments. To make an example, a 2D drawing tool can be enough to design the strips of the circuit and, in the case of more dielectric layers, the capability to manage more design layers (sheets), option that is almost ever included in the modern drawing tools and satisfies completely the requirements. But at the same time the designer must be able to describe the vertical structure of the circuit, assign its characteristic to each dielectric material and associate each metallization level to a specific dielectric. Such a description requires either a full 3D representation, which is complete but onerous, or a

schematic representation of the vertical structure of the circuit and just a link between the vertical structure and the design level.
This second option is what we realised inside GID and what will be described in the following of this paper. In the first part of the next section we will describe the part of the problem type dedicated to the creation of the vertical structure. In the second part we will show how insert circuits elements at the right level through a parametric description and in the last part we will describe briefly the generation of the input files for the electromagnetic analysis and the integration of GID inside MADS [4, 5].

A PROBLEM TYPE FOR MICROWAVE PRINTED CIRCUITS

A new problem type has been developed to answer to the main requirements of the printed circuit design. All the special features developed are contained both in a toolbar and in a menu (see Fig. 1).

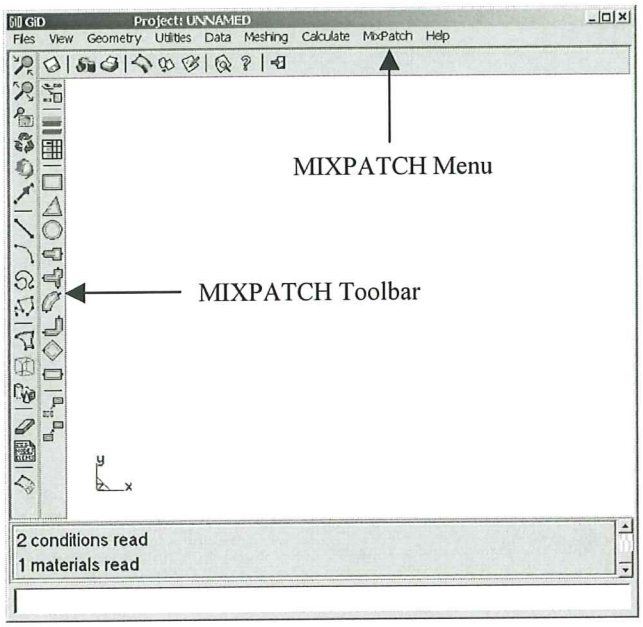


Fig. 1 – GID-MIXPACTH environment

The Toolbar contains four main classes of tools: Exportation of geometry and mesh for the Electromagnetic analysis, Design and management of the antenna substrate, Insertion of printed circuit components and two shortcuts for MOVE-SURFACES and COPY-SURFACES functions.

The first step in the design of a printed circuit is the definition of the dielectric layers where the circuit will be printed out. A special frame, shown in Fig. 2-a, and several procedures have been developed to give to the user an effective way to build this structure.

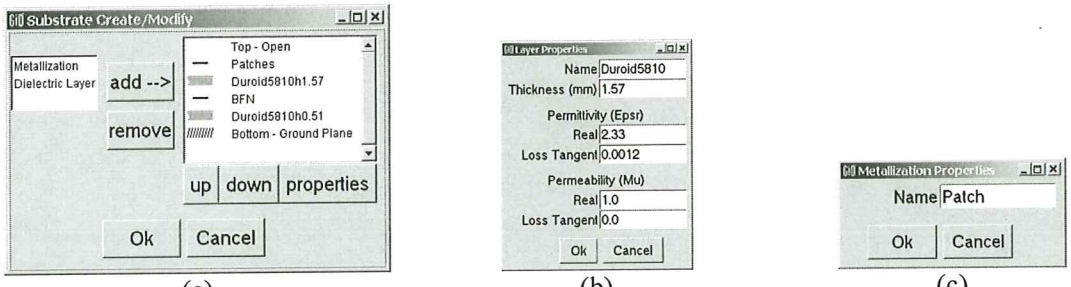


Fig. 2 – Frame for the design of the circuit substrate

The list on the left contains the type of materials that can be added to the antenna substrate. In the current situation only metallizations and dielectric layers can be added to the circuit substrate, but other types, like slots and via holes, could be easily added in a further version. The list on the right of the frame shows the substrate structure of the circuit as has been built by the user. The user can use the buttons in the frame to add/delete, move up/down an element of the list and assign some properties to it (see Figures 2-b and 2-c).

Once the substrate setup finished and the "OK" button in Fig. 2-a pressed, a number of GID layers, equivalent to the number of metallizations present in the substrate structure, will be automatically created. Then the "Substrate" frame, shown in Fig. 3, will pop up. This frame shows a schematic view of the vertical section of the antenna substrate and allows the management of the layers during the drawing phase. The same actions can be done also in the original "layers frame" from GID, but this schematic view helps more the user during the design phase.
The second step is the drawing of the circuit. The circuit geometry can be drawn using both the standard GID tools and the predefined elements in the dedicated toolbar.
The predefined elements are at the moment: rectangle, triangle, circle, line-step, tee-junction, curve, chamfered bend, corner-fed patch and line, but other typical shapes could be easily added in the future. Fig. 4 shows the frame used for the parametric insertion of a microstrip line (which is basically a rectangle, but where the reference point for insertion can be the middle of an edge). The user can set the dimension of the line just inserting length and width in the appropriate fields and he can eventually also impose a rotation and an offset with respect to the point where the element will be inserted. This point can be inserted manually or picked up in the screen with the mouse.

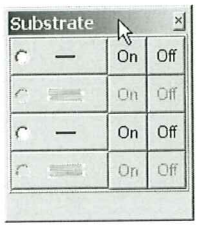


Fig. 3 – Frame for the management of the layer in the drawing phase

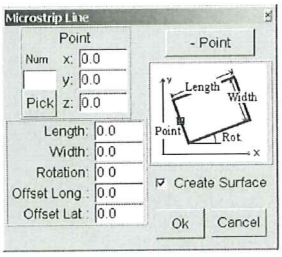
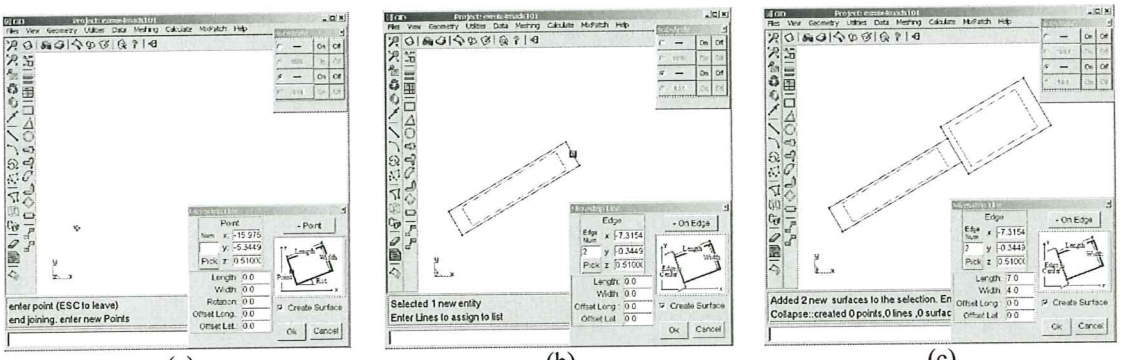


Fig. 4 – Example of frame for the parametric insertion of new components

To help the user, when the point is picked with the mouse the Z-coordinate of the point is set automatically to the z level of the metallization that is active at that time (see Fig. 5-a).
Another useful feature for printed circuit design is that a new element can be attached directly to the edge of a previous element. In fact the user can choose to add the element on an edge instead that in a point. In this case the user can pick on an existing edge and automatically its number is retrieved and the coordinate of its center are computed and displayed in the corresponding field (see Fig. 5-b). At the same time also the rotation of the element is automatically computed so that the new element is perfectly matched to the previous one (see Fig. 5-c).

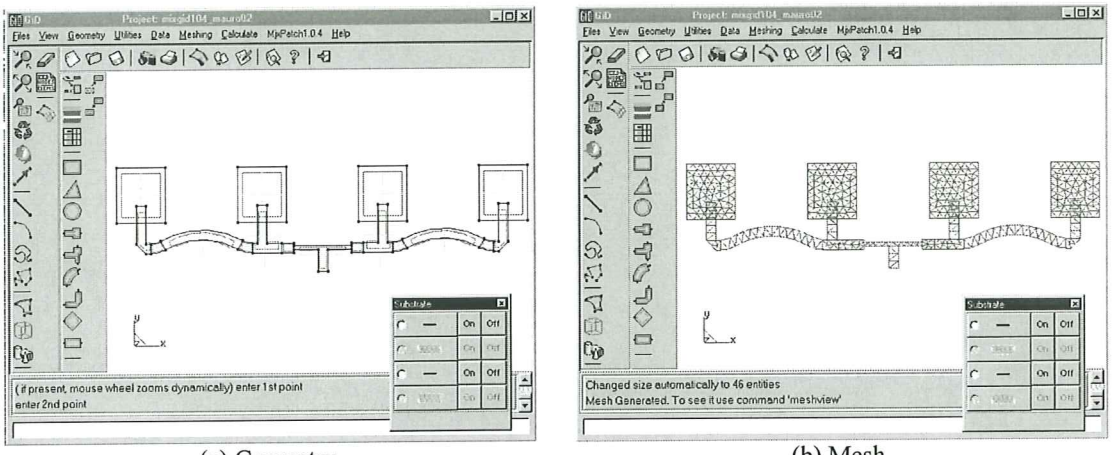


(a) (b) (c)

Fig. 5 – Example of insertion of two cascaded microstrip lines with different section.

After the substrate and the circuit have been designed it remains still to identify the points where to put the generators that will feed the circuit. This has been done using the GID feature to create new conditions that can be applied to any of the geometrical entities. More in details a new condition called "Port" which apply to "lines" has been generated inside the problem type. This condition enables also the insertion of the characteristic of the generator, like the type, the internal impedance and the value of the magnitude and phase.
In Fig. 6 we show an example of geometry and mesh of a printed antenna built on a two layers two metallizations substrate.
GID together with the problem type described above is used in our laboratory as pre-processing for the design of printed circuits and antennas. Moreover the same tool, with some minor changes, has been also integrated inside the system MADS (Multi-purpose Antenna Design Simulator), a framework developed

within a European ESPRIT Project. In this case GID is not used as the main user interface, but is just a pre-processing tool inside a bigger system. The MADS environment is in charge of the launch of GID with a new or an existing project and all the file management is also controlled by the MADS system. Thus to avoid possible conflicts the menu item and the main toolbar of GID have been customized in order to remove all the operation requiring the opening of a file browser like "open", "save as.." etc. and also the possibility to create a new project.



(a) Geometry. (b) Mesh.

Fig. 6 – Example of design. Printed antenna array of 4 electromagnetically coupled patches.

CONCLUSIONS

As explained before, microwave printed circuit design is a delicate task. In this paper we have presented the special features we added to GID in order to tailor it for our purposes. Nevertheless the world of electromagnetic analysis is very wide and each technique can have some special requirements in the pre-processing phase and for this reason we think that, as it happened in the past for mechanical or building engineering, it would be useful to have a CAD environment specially designed for electromagnetic engineering which could help microwave engineers and researchers in their work. In this optic we think that GID could be a good starting point because is a powerful and flexible approach for the treatment of the geometry and of the mesh, and it can be quite easily customised and coupled to a specific method of analysis.

Acknowledgements

This work was supported by the European Union within the Esprit-MADS Project, project no. 28363.

References

[1] J.R. Mosig, R.C. Hall and F.E. Gardiol, "Numerical analysis of microstrip patch antennas", Handbook of microstrip antennas, Peter Peregrinus, London, 1989, ISBN 0-86341-150-9, pp. 393-452.

[2] J.R. Mosig, "Integral equation Technique", chapter 3 in Numerical Techniques for Microwave and Millimeter Wave Passive Structures, pp. 133-214, T. Itoh ed., John Wiley, New York, 1989.

[3] J.R. Mosig and F.E. Gardiol, "Analytical and numerical techniques in the Green's function treatment of microstrip antennas and scatterers", IEE PROC., Vol. 130. Pt. H, No. 2, Mar 1983, pp. 175-182.

[4] Esprit-MADS Project 28363 (1998-2001), Multipurpose Antenna Design System, Final Qualification Test Document.

[5] Esprit-MADS Project 28363 (1998-2001), Multipurpose Antenna Design System, Software User Manual.

FRACTALCOMS PROJECT: MESHING FRACTAL GEOMETRIES WITH GiD

Josep Parrón[(1)], Juan Manuel Rius[(1)], Jordi Romeu[(1)], Alex Heldring[(1)] and Gabriel Bugeda[(2)]

[(1)]*Department of Signal Theory and Communications (TSC)*
Universitat Politècnica de Catalunya (UPC)
Jordi Girona 1-3, mòdul D3, Campus Nord UPC, 08034 Barcelona, Spain

[(2)] *Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE)*
Universitat Politecnica de Catalunya (UPC)
C/ Gran Capita s/n, mòdul C1,Campus Nord UPC, 08034 Barcelona, Spain

SUMMARY: Antennas based on fractal structures present some particular properties as self-similarity and miniaturization that overcome some of the main limitations of classical antennas. Defining and meshing the fractal geometry are the two first stages in the numerical analysis of their electromagnetic behavior. Fractal structures are highly convoluted with very small details, nevertheless, some of these fractals can be easily defined with very few parameters using the concept of an iterated function system (IFS). Some other important points about the meshing of fractal geometries are also described in this communication. GiD provides a very suitable environment to create arbitrary 3D meshes. One of the objectives of Fractalcoms project is to extend the capabilities of GiD in order to generate automatically the mesh of fractal geometries taking as input data the IFS parameters.

KEYWORDS: Fractals, multiband antennas, small antennas, iterated function system, adaptive meshing, numerical methods.

INTRODUCTION

The emergence of antennas with fractal geometries has given an answer to two of the main limitations of the classical antennas: the single band performance and the dependence between size and operating frequency [1]. The self-similar properties of certain fractals result in a multiband behavior of the antennas built after these fractals [2][3]. On the other hand, the highly convoluted shape of these fractals makes possible the reduction in size, and consequently in mass and volume, of certain antennas [4]. These reductions can make possible to combine multimedia, communication and teledetection funcionalities in a reduced space like a handy phone or even a wristwatch or a credit card. For instance, it has been demonstrated that a fractal antenna can provide GPS services within a conventional mobile cellular phone. The aim of Fractalcoms project is to explore the performance limits of these fractal shaped devices.

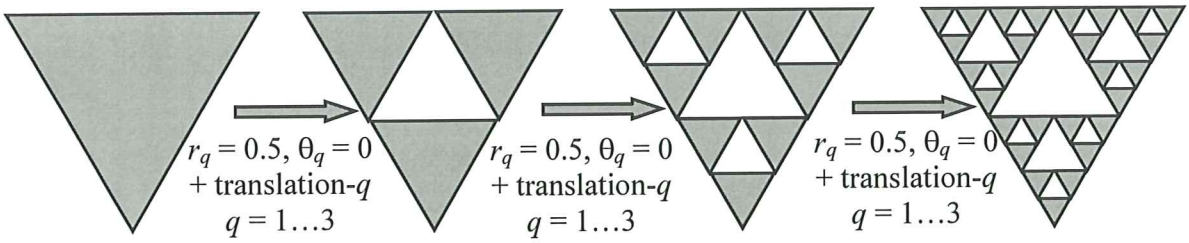


Fig. 1: Four iteration Sierpinski fractal obtainted after a set of affine transformations

Fractal structures can be analyzed using the integral equation methods (IE), in conjunction with the well-known Method of Moments (MoM) [5]. The MoM splits the fractal geometry in basis functions. Rao, Wilton and Glisson linear triangles with common vertices (RWG) [6] are usually chosen for their flexibility to model 3D arbitrary surfaces.

Since fractal structures are highly convoluted with very small details, many basis functions (unknowns) can be required in the solution of the problem. In order not to increase excessively the computational requirements a simple and smart definition and meshing of the geometry must be done.

GEOMETRY GENERATION: ITERATED FUNCTION SYSTEM

A simple way to build most fractal structures is using the concept of iterated function system (IFS) [7]. An IFS is defined by a set of Q affine transformations in the plane $\{\omega_q\}_{q=1}^{Q} : R^2 \rightarrow R^2$ which can be written as:

$$\omega_q(x) = A_q x + t_q = \begin{pmatrix} r_{q1}\cos\theta_{q1} & -r_{q2}\sin\theta_{q2} \\ r_{q1}\sin\theta_{q1} & r_{q2}\cos\theta_{q2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_{q1} \\ t_{q2} \end{pmatrix} \tag{1}$$

where x_1 and x_2 are the coordinates of point x. If $r_{q1}=r_{q2}=r_q$ with $0<r_q<1$, and $\theta_{q1}=\theta_{q2}=\theta_q$, the IFS transformation is a contractive similarity (angles are preserved) where r_q is the scale factor and θ_q is the rotation angle. The column matrix t_q is just a translation on the plane.

Applying several of these transformations in a recursive way, the self-similar fractal is obtained. In fact, self-similarity can be also understood as the property by which the fractal is found inside the fractal itself but at a smaller scale. In Fig. 1 and Fig. 2 can be found two examples of the recursive procedure followed to obtain the fractal shape. It must be noted that

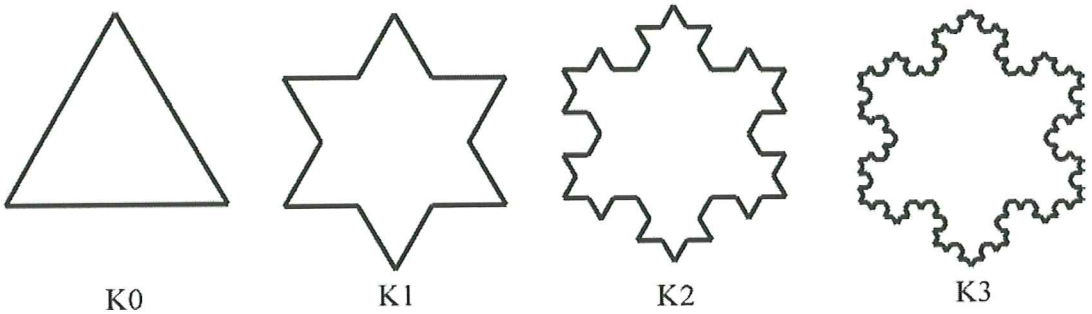


Fig 3. Koch island after three iterations

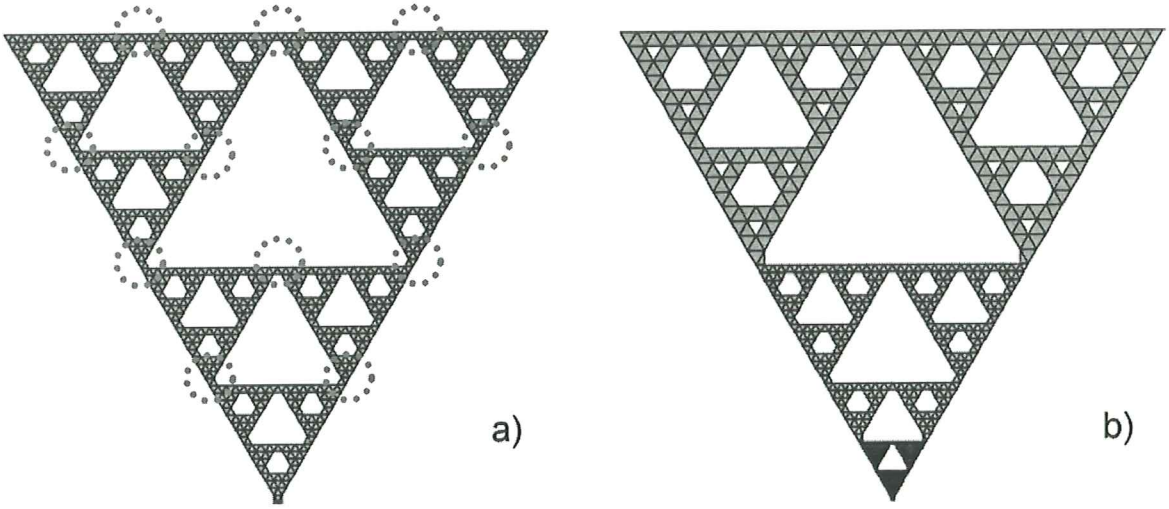


Fig. 3: a) Sierpinski gasket monopole uniformly discretized in 1700 RWG triangles.Dotted lines mark some of the small triangles joining the bigger triangles.
b) The same monopole with an adaptive meshing of 1364 RWG triangles.

the input arguments are only the initial geometry (initiator), the number of iterations and some parameters about the transformations to perform.

MESHING FRACTAL GEOMETRIES

Once the geometry has been generated, it must be descretized in RWG triangles, Fig. 3a shows a uniform discretization of a Sierpinski fractal. In Fig. 3a, it must be also noted some small additional triangles joining the larger ones. These small triangles may be required for certain initiators to be able to represent the electric density of current flowing along the whole antenna.

It is important to point out that, when applying MoM, memory storage is proportional to N^2, while computation time is proportional to N^3 where N is the number of unknowns. Then, a large number of basis functions (unknowns) will result in a large linear system that can easily overcome the capabilities of our computer.

A way to reduce these computational requirements is to take advantage of the fact that the electric density of current is concentrated in certain regions to perform an adaptive meshing. Fig. 3b shows as a finer mesh can be used in those regions with high density of current while the rest of the structure is represented with a coarser mesh. The loss of accuracy in the final results is not significant but the saving of time and memory can be very important.

CHALLENGES

GiD has been developed by CIMNE, one of the partners of Fractalcoms project, and provides a very suitable environment to create general 3D meshes. One of the objectives of Fractalcoms project is to add new features to GiD in order to create the fractal geometry and the adaptive mesh taking as input data the IFS parameters and the initiator. An appropiate mesh will decrease the number of unknowns of the problem and thus the computational requierements will be also reduced.

ACKNOWLEDGEMENTS

This work is supported by the European Commission through FET project IST-2001-33055 (FRACTALCOMS), the "Departament d'Universitats Recerca i Societat de la Informació (DURSI)" of the "Generalitat de Catalunya" under "Distinció de la Generalitat de Catalunya per a la Promoció de la Recerca Universitaria", and the Spanish "Comisión Interministerial de Ciencia y Tecnología (CICYT)" through grant TIC 2001-2364-C01-01.

REFERENCES

1. D.H. Werner and R. Mittra eds., *Frontiers in electromagnetics*, IEEE Press, 1999, chapters 1-3.
2. C. Puente, J. Romeu, R. Pous, X. García, F. Benitez, "Fractal multiband antenna based on the Sierpinski gasket", *IEE Electronic Letters,* vol. 32, nº 1, pp. 1-2, January 1996.
3. C. Puente, J. Romeu, R. Pous, A. Cardama, "On the behavior of the Sirpinski multiband antenna", *IEEE Transactions on Antennas and Propagation,* vol. 46, nº 4, pp. 517-524, April 1998.
4. C. Puente, J. Romeu, R. Pous, J. Ramis, A. Hijazo, "Small but long Koch fractal monopole", *IEE Electronic Letters,* vol. 34, nº 1, pp. 9-10, January 1998.
5. R. F. Harington, *Field Computation by Moment Methods*, MacMillan, 1968.
6. S. M. Rao, D. R. Wilton, A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape", *IEEE Transactions on Antennas and Propagation*, vol 30, pp. 409-419, 1982.
7. H. O. Pietgen, H. Jurgerns, D. Saupe, *Chaos and Fractals, New Frontiers in Science*, Springer Verlag, New York, 1992.

A GiD-FEAP Interface

A. Ghisi

Department of Structural Engineering, Technical University (Politecnico) of Milan, Italy

SUMMARY: FEAP is a Finite Element Analysis Program. It has been implemented by R.L. Taylor and it is widely used in academic environments. In this paper an interface between GiD and FEAP 7.3, implemented by the author both for pre and post-processing purposes is described. The pre-processing phase is performed in GiD. The developed user interface allows for the interactive assignment of data for most of the constitutive laws already present in the original version of FEAP 7.3. At the moment, only small displacement problems with solid elements can be dealt with.

KEYWORDS: FEAP, customization, post-processing, solid elements, small displacements

INTRODUCTION

At the Department of Structural Engineering (DIS) of the Politecnico di Milano (Italy) the professional version of FEAP has been chosen as the basic platform for development of new methodologies in finite element structural mechanics. The release 7.* of FEAP added many new capabilities to the program, which now appears to be an effective tool for structural analysis: it offers a fair choice of elements, formulations and constitutive laws and its modular structure permits a rather easy introduction of new features. It may solve static and dynamic problems, in linear and non-linear structural mechanics. New versions are released on a regular basis adding more options (for example, version 7.4 added other 3D elements). To enhance the use of the program for our purposes, it was decided to develop interfaces between FEAP and an automatic pre-processor and also one or more post-processors. GiD was selected for its simplicity and powerful capabilities.
To accomplish this task, it was necessary:
1) to define two new problem types in GiD 6.2.0.b, one for two-dimensional problems (feap_7_3_2D) and one for three-dimensional problems (feap_7_3_3D). In particular, a *.bas files had to be written in order to create automatically the input files for FEAP.
2) to introduce into FEAP the possibility to create an output database in a format comprehensible for the post-processing module of GiD. This was obtained by defining a new macro (i.e. a new keyword recognized by FEAP) called WRIO (acronym from WRite Input/Output). This macro manages the data flux and creates the new ASCII files for post-processing, named filename.flavia.msh and filename.flavia.res, i the format required by GiD.
3) since in our department data can be post-processed also using TECPLOT™, a second interface between FEAP and TECPLOT™ has been developed. Incidentally, GiD can also post-process TECPLOT™ files. This second interface makes use of a separate module, called DISFEAPPOST: it accesses a database in binary format (a direct access file) and translates the data in a format useful for TECPLOT™ or also for GiD.

FEATURES OF THE PROBLEM TYPES

The customization of the problem types in GiD includes 9 conditions and 11 materials.

Condition	FEAP_7_3_2D	FEAP_7_3_3D
Point_constraint	X	X
Point_load	X	X
Point_plot	X	X
Line_constraint	X	X
Face_load	X	
Point_mass_damper_and_ spring	X	X
Surface_constraint		X
Surface_load		X
Hydrostatic		X

Table 1: condition types

Material	FEAP_7_3_2D	FEAP_7_3_3D
Solid elastic isotropic	X	X
Solid elastic orthotropic	X	X
Solid elastic transversally isotropic	X	X
Solid plastic Von Mises with isotropic/kinematic hardening	X	X
Solid viscoelastic	X	X
Fourier analysis isotropic	X	
Frame elastic isotropic		X
Shell elastic isotropic		X
Membrane elastic isotropic		X
Plate elastic isotropic		X

Table 2: "materials"

In our concept a "material" is a combination of a constitutive law and a finite element family (solid elements/shells/plates, etc.). "Materials" are organized in two books: one for solid elastic "materials" and one for other "materials".

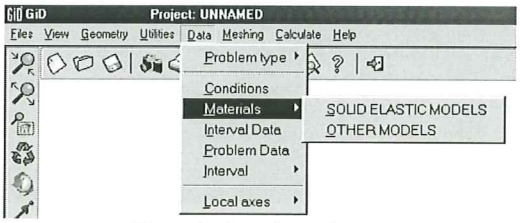


Fig. 1: books of "materials"

Each "material" requires different input for the more commonly used parameters (i.e. Young modulus, Poisson's coefficient) and other, more rarely used parameters, like body forces. It is possible to activate or deactivate the input for these less common parameters by a switch; in this way, if they are not needed, they disappear or are grayed, so the user cannot enter them by mistake in the analysis.

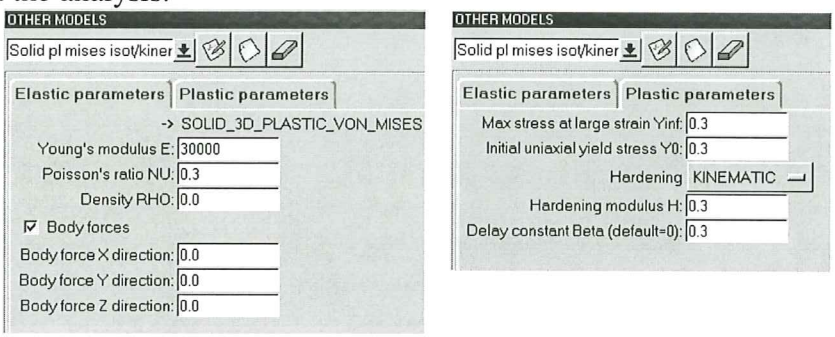


Fig. 2: a "material": Von Mises elastic-plastic law with isotropic or kinematic hardening

An useful feature has proven to be the possibility to add an image to remind the user about the meaning of the symbols appearing in the window, as shown in Fig. 3.

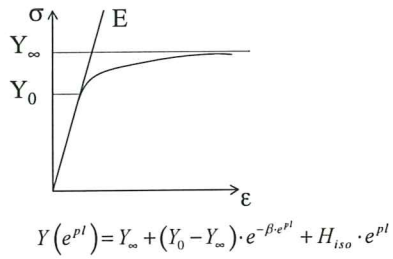


Fig. 3: this image appears in the same window of the parameters and it helps the user to remember what the symbols mean

A condition can be imposed over the geometry or over the nodes, like in the standard version of GiD. The feap_7_3*.bas file translates the model in a FEAP input file named *ifilename* (the prefix "i" is needed for standard input in FEAP): the default is a standard static linear elastic analysis in small displacements. At present it is responsibility of the user to change the input for a more complex analysis, for example a time dependent analysis, but other improvements are currently in progress.
Using the .bat file it is possible to run the analysis directly from GiD (the program automatically creates the FEAP input file) and check the results with the post-processing facilities of GiD (at present only for solid elements).
The element choice in input includes 2D and 3D solid elements (linear and quadratic), Fourier analysis solid element, plate, frame and truss elements. Input for new constitutive laws will be added in the future.

THE FEAP CUSTOMIZATION

A new FEAP macro, named WRIO (WRite Input/Output), has been added to the FEAP vocabulary: it can produce directly either the .flavia.msh and the .flavia.res files (suitable for GiD) or some binary databases. The latter will be processed after the end of execution using an apposite program (named DISFEAPPOST) which creates the files for post-processing, according to one of two different formats, summarized in Fig. 4, below.

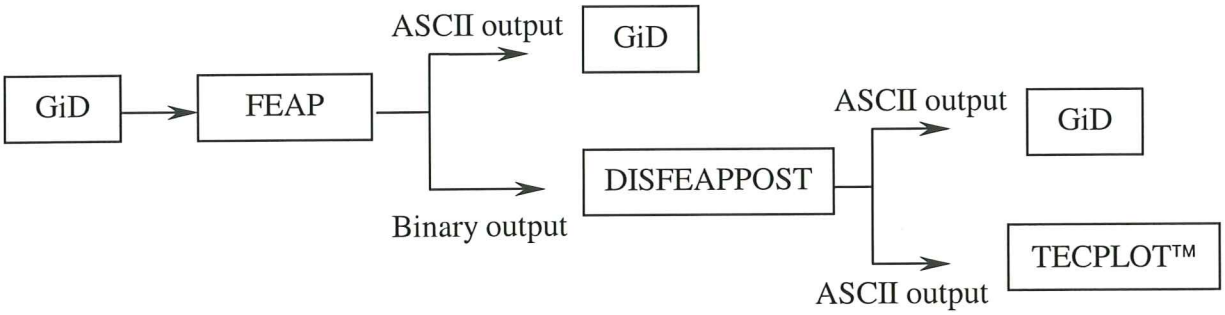


Fig. 4: two different ways to postprocess the results, the first directly from the solver to GiD with an ASCII file, the second through a binary file to two postprocessors.

The macro WRIO has two options, one for creating an ASCII output file for GiD (this is the default) and one for binary file output. The latter facility is completed by another program (DISFEAPPOST) to translate binary data into ASCII file for GiD or TECPLOT™. With this approach it is possible to select after the analysis is finished which time-step or which type of result is needed for visualization. This second methodology is time-saving in presence of large and/or non-linear analyses, where writing ASCII files becomes a very time-consuming

task. The program uses the new post-processing format made available in the new beta release (at present, GiD 6.2.0b) which is needed to visualize the results.
It is possible to combine different element types in GiD, even though not in a straightforward way. For example it is possible to use both quadrilateral and triangular elements in a mesh for two-dimensional solid mechanics problems. This important feature has been kept in the output from FEAP to GiD, with some programming. A new macro was added in FEAP, to manage the output of displacements, stresses, strains and internal variables. FEAP automatically distinguishes between the different element types, each one with its number of nodes and of Gauss points. Now the program builds dynamically allocated lists, where it puts the element numbers corresponding to each different element type. Each one of these is labeled accordingly. For example Quad4_4_GP means quadrilateral elements with four Gauss points, Tetra4_1_GP means tetrahedra with one integration point, and so on. Every different list (in fact, this is a list array, using the facilities of FORTRAN 90) is treated as a different mesh_name in the post-processing file; for each list FEAP automatically puts in the post-processing file the correct definition for the Gauss point type with the gauss_points_name corresponding to the mesh_name.

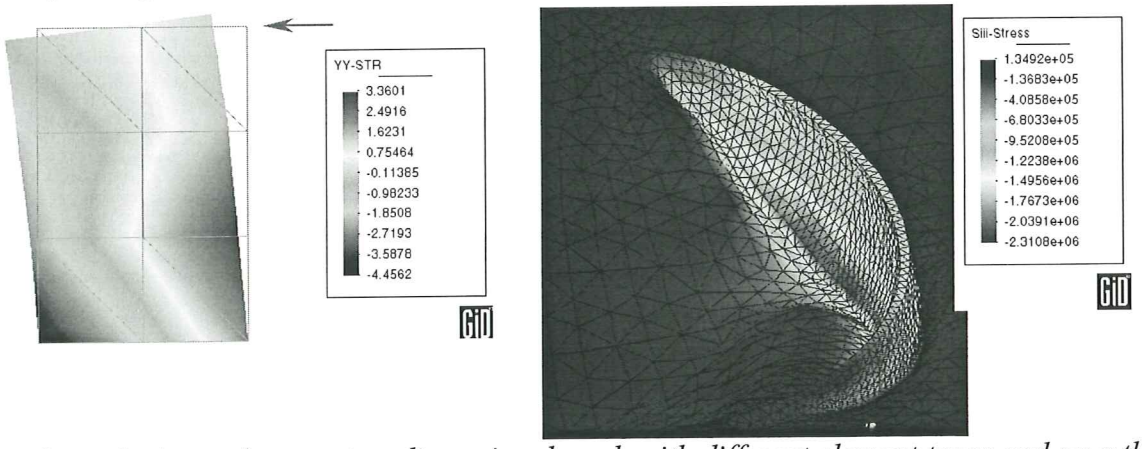


Fig. 5: analysis results on a two dimensional mesh with different element types and on a three dimensional tetrahedral mesh of an arch-gravity dam under self weight and hydrostatic load. Examples obtained with GiD and FEAP.

So the results are obtained from FEAP at nodes or at Gauss points and for different element types in the same mesh, even if visualization at Gauss points at present is limited to one element type only in GiD.

FUTURE DEVELOPMENTS

The developed enhancements represent a necessary preliminary step to allow for the introduction in GiD-FEAP of new capabilities for analyses with nonlocal damage models and possible transition to propagating discrete cohesive cracks. The availability of the source code in FEAP and easiness of problem customization in GiD represents a very promising combination that we hope will greatly contribute to help our research in structural mechanics.

REFERENCES

1. R. Ribó, M.A. de Riera Pasenau, E. Escolano "GiD Reference manual", (release 6) International Center for Numerical Methods in Engineering (CIMNE), 2001
2. R.L. Taylor, "FEAP 7.3 manuals" (User manual, Programmer Manual, Theory Manual, Example Manual), 2001

Customization of GiD & FEAP for Scientific Applications

T. Raible[1], D. Freßmann[1], R.L. Taylor[2], P. Wriggers[1]

[1] *Department of Civil Engineering and Surveying, Institute of Mechanics and Computational Mechanics, University of Hannover, Appelstr. 9A, 30167 Hannover, Germany*

[2] *Department of Civil and Environmental Engineering, 721 Davis Hall, University of California, Berkeley, CA 94720-1710, USA*

SUMMARY: The visualization of results from FEM calculations is an important step in the analysis of complex problems. Due to the fact that high performance graphic cards where only available to the limited area of workstations and graphic standards changed over long time periods, major development effort was set on the scientific simulation codes. Recent development of computer hardware made adequate computing and visualization power available to everybody and increased the necessity of easy to use pre- and post processing. For scientific use, it is important to provide the full user interface capabilities of the original finite element (FE) code due to code development and debugging reasons. While visualization aspects become important, the FE code has to provide all information to the post processor. The stage of development for an interface of GiD and FEAP will be demonstrated by examples of current research projects such as the process simulation of foam forming, model adaptivity for beams and an adaptive 3D implementation of hanging nodes.

KEYWORDS: visualization of FEM results, software customization.

INTRODUCTION

Visualization of numerical results has been a topic since the invention of sophisticated calculation tools such as the Finite Element Method (FEM). As high performance graphic cards were only available to the "workstation domain", development was mainly focused on the numerical analysis and algorithmic part of these FE codes. This situation changed after introducing consumer computers (PC's) in the early 1980's, although they were still quite expensive. With respect to recent computer hardware development concerning memory and visualization power, an exceptional development has been taking place in that field. On the application side, the awareness of the advantages of FEM solutions established these methods as standard analysis tools. Industrial requirements toward an integrated development process in terms of Computer Aided Engineering (CAE) accelerated the progress of pre- and post processing tools. Due to that coupling of FE codes like FEAP and graphical oriented pre- and post processing tools like GiD, integration of both is required. This process has also consequences for the education and the scientific work at universities. Students have to be trained in the philosophy and practice of such tools and scientists use them to improve their daily work. Another aspect is the direct application of new developments to more complex problems than standard test examples.

Customization requirements

Application of pre- and post processing tools in our context needs flexibility on both sides, visualization and calculation codes. This will be explained by the normal development in terms of the calculation code development process:

1. Significant effort is usually set on the development and implementation of new theories and algorithms into the FE code. This is, unfortunately, still a time consuming work and debugging and classical code development tools are used to solve these problems. Mesh generation capabilities of the FE codes are sufficient at this stage and the post processing capabilities can be used to verify the results. A powerful interface in the FE code provides the data for the post processing.

2. After functionality is guaranteed, the pre-processor interface has to be adjusted to new specifications of the FE code input. Now, standard problems defined by the PPPT from a data bank can be used to perform further tests on the reliability, potential and performance of the development.

3. As industry is often interested in the transfer of new developments, extended pre-processing capabilities are necessary while complex geometries are usually delivered in CAD data formats. Now, geometry and mesh generation combined with an intuitive working surface are important to deliver results in limited time.

Here GiD was linked to the FE code FEAP as meshing and visualization tool. Within that integration process, certain difficulties appear because of different philosophies of the programs. In the case of GiD and FEAP, the interface with regard to geometry like nodal coordinates can be implemented easily whereas the classification of the elements has to be done carefully. The connection between the element classes of FEAP and the geometric objects if GiD depend on the dimension of the problem. This is shown in Fig. 1.

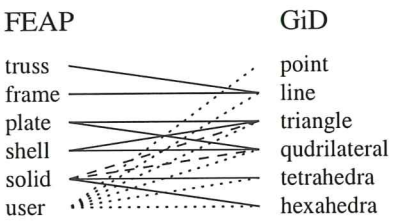


Fig. 1: Element definition connections

Dashed lines indicate connections for 2d cases. For user defined elements, connections can be established directly to all GiD objects as indicated by the dotted lines. For output of e.g. stresses, strains or error indicators, the GiD point visualization option is used to process raw gauss point data from all FEAP elements.

Examples

The selected examples demonstrate particular details for the pre-/post processing process and the interpretation of the results. As the integration process of FEAP and GiD is still going on, the results taken from current research project give an outline of today's functionality.

Model adaptivity

In structural engineering, model adaptivity is an attractive method to improve the accuracy in parts of the structure that are of special interest. This has been realized for structural parts like profiled beams using frame- and shell-elements. To connect the two parts, special transition elements that preserve displacements, internal forces and heat conduction in the transition zone are used. In the example shown in Fig. 2, a uniform

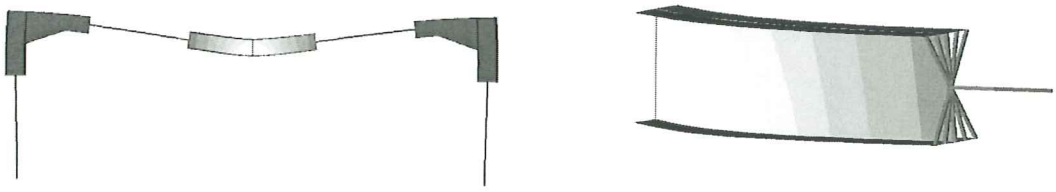


Fig. 2: Temperature distribution, detail of transition elements

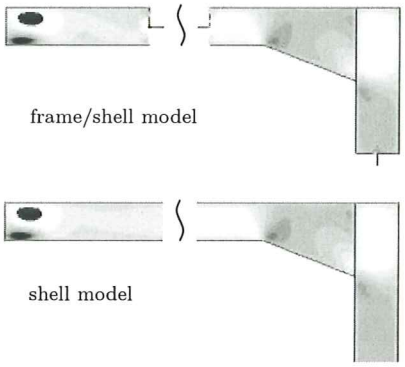


Fig. 3: Comparison of S_{xy} stresses

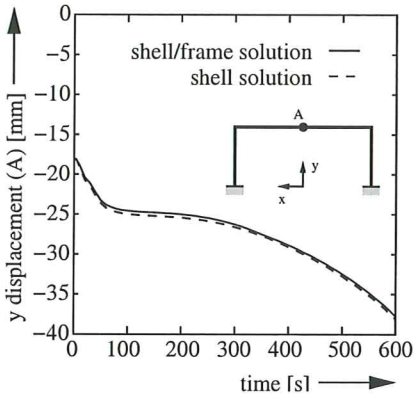


Fig. 4: Displacement in point A

loaded frame is heated in the center (point A, Fig. 4). As a detail, the realization of the transition elements is also described. Good agreement is achieved for the σ_{xy} stresses shown in Fig. 3 and the vertical displacements shown in Fig. 4.

Adaptive 3d meshing using hanging nodes

Adaptive meshing techniques for mesh re- and de-refinement are still under development. One possibility is to use so called "hanging nodes", which are located in faces or on edges of neighboring elements like indicated in Fig. 5. Elements can be easily concentrated by subdividing from a coarse starting grid.

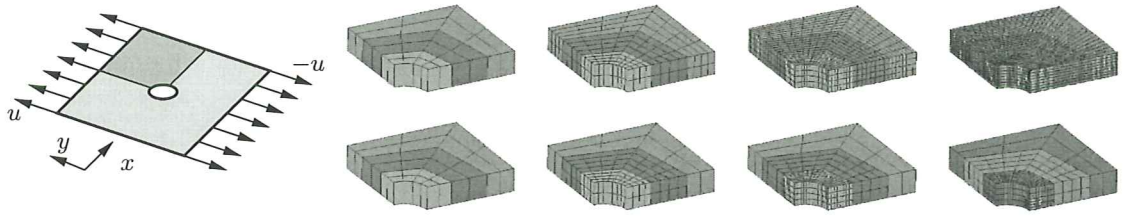


Fig. 5: Problem definition and comparison of uniform / hanging-node adapted meshes

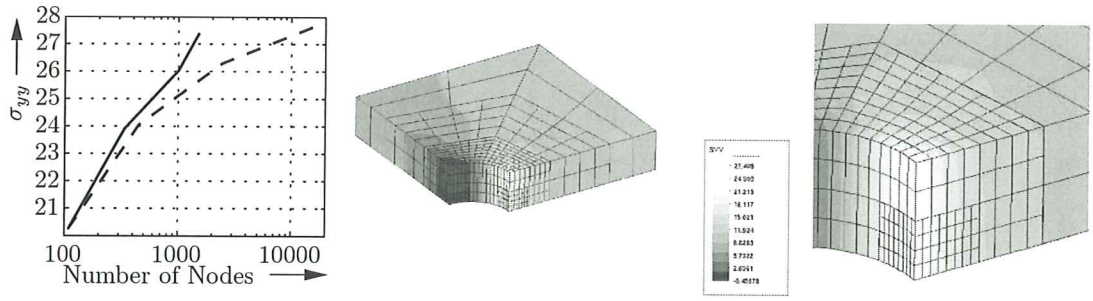


Fig. 6: Quality of calculation (line hanging nodes, dashed uniform adaption) and σ_{yy} stresses [N/mm^2]

A comparison of the quality of the calculation concerning the stresses σ_{yy} is given in Fig. 6. The hanging node calculation was done using approx. 1.600 nodes, where as the uniform refined mesh contained 16.000 nodes which shows the potential of saving computing resources.

Visualization of special computation techniques

Another example is the calculation of particulate materials at finite strains. In these simulations, the particles are described by spheres that are embedded in a matrix. A representative volume element (RVE) is shown in Fig. 7 without the matrix (particles only). This RVE is regularly meshed using hexahedra elements. The particles are introduced by

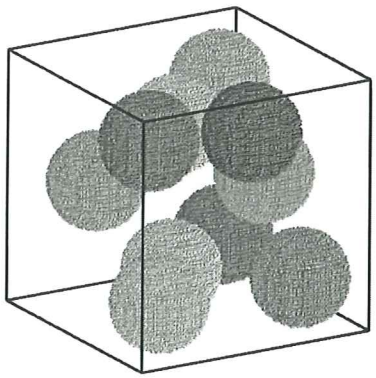


Fig. 7: RVE geometry

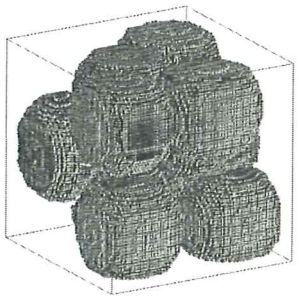


Fig. 8: Main stresses (min. -10 max.-400 N/mm^2)

applying the different material properties on Gauss point level. To improve the geometrical description, the number of Gauss points used in the elements containing both, matrix and particles, is increased. The first main stress computed on discrete Gauss points after homogenious compression (50 % of initial RVE volume) is shown in Fig. 8 (approx. 90.000 Gauss points on the surfaces of the spheres).

Conclusions

Modern engineering has generated powerful FE codes like FEAP and pre- and post processing tools like GiD. The efficient customization of both leads to an integrated tool for many applications in engineering. Scientific development with respect to post processing capabilities is supported and transfer to industrial applications is improved by the use of standard CAD interfaces.

APPLICATION OF GID TO 3D BIOMECHANICS PROBLEMS, INTERFACING WITH ABAQUS AND FEAP

Juan Carlos García*, Javier Rodríguez* and José María Goicolea*

**E.T.S.I. Caminos Canales y Puertos, Universidad politécnica de Madrid, Ciudad Universitaria, Madrid 28040, Spain*

SUMMARY: In the context of a biomechanic research work for modelling arterial wall soft tissue, we have been employing GiD for the development of finite element models. The object is, starting from complex 3D geometries obtained from in-vivo images, to develop F.E. models, analyse them with ABAQUS and FEAP (for which purpose the appropiate interfaces with GiD have been developed), and finally visualize the results. In order to determine reliable models for the development of the atherosclerosis, a description of the hemodynamic conditions in the places of interest is detailed, being necessary a numerical tridimensional simulation that allows to calculate the distribution of shear stress in the interface and the stress state in the arterial wall. For that, initial geometry data are obtained by angiographies, determining the lumen, and IVUS (IntraVascular Ultrasound System), providing information about the morfology and structure of the arterial wall. The present paper focuses on the treatment of real geometric data of an arterial bifurcation with GiD (imported as IGES), and presents the results (also using GiD) of stresses in the arterial walls.

KEYWORDS: GiD, ABAQUS, FEAP, in-vivo images, IGES, atherosclerosis, angiographies, IVUS.

INTRODUCTION

Nowadays in Spain cadiovascular diseases are the first cause of dead from 75 years old and the second between 15 and 74 years old. This justifies the interest in offering computer assistance for the cardiac diagnostic and therapy. One of the most important research objectives in this field is to assess about the development of atherosclerosis from tridimensional numerical simmulations, based on the calculated hemodynamic representative data (such as pressure, shear stress in the wall, etc.) as well as the mechanical response of the wall, mainly wall stress.

This paper presents the modelling of an arterial wall bifurcation. Geometry data, corresponding to a patient with stenosis was obtained from angiographies and IVUS (In-

traVascular Ultrasound System), and imported by GiD (CIMNE, 1999) in IGES format. In the other hand, data about blood pressure and shear stress on the inner surface were calculated with the computer fluid mechanics program FLUENT (Fluent, 2001). The model was analysed by the finite element analysis program for solids ABAQUS (Hibbit et al., 1998), and the results were finally postprocessed with GiD.

In the last part of the paper, it is also presented the way as GiD interfaces with FEAP (Taylor, 2000) using its customization capabilities.

MODELLING OF AN ARTERIAL WALL BIFURCATION

Mesh generation. First of all, an IGES file which define the inner and outer surfaces of the arterial wall was obtained after the analysis of angiographies and IVUS (in-vivo images), (see Fig. 1).

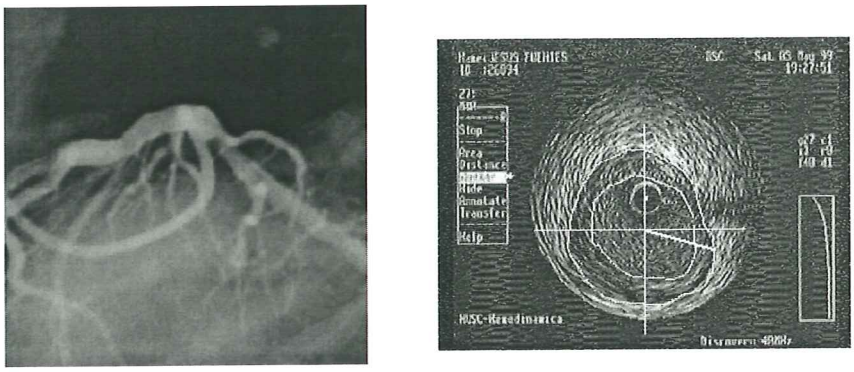


Fig. 1: Images from an angiography (left) and IVUS (right)

The meshing of the surfaces was made using triangular elements. The size of the elements has been determined according to the following criteria:

- It is intended that every element has at least one load point. It has been found out that this requirement is fullfilled with a characteristic element size which is double of the average distance among load points.
- It is necessary to refine the mesh in those places where high gradients of stresses or high variations in the thickness are expected.

Definition of loads and boundary conditions. A semiautomatic calculus has been made based on the files containing the generated mesh and the blood loads previously calculated with FLUENT.In this process both the thickness distribution and the nodal loads of the inner mesh are calculated.

It is supposed that all movements in the upstream and downstream sections are restricted. This is easy to force by creating a nodal set (`*NSET`) with the nodes selected in GiD.

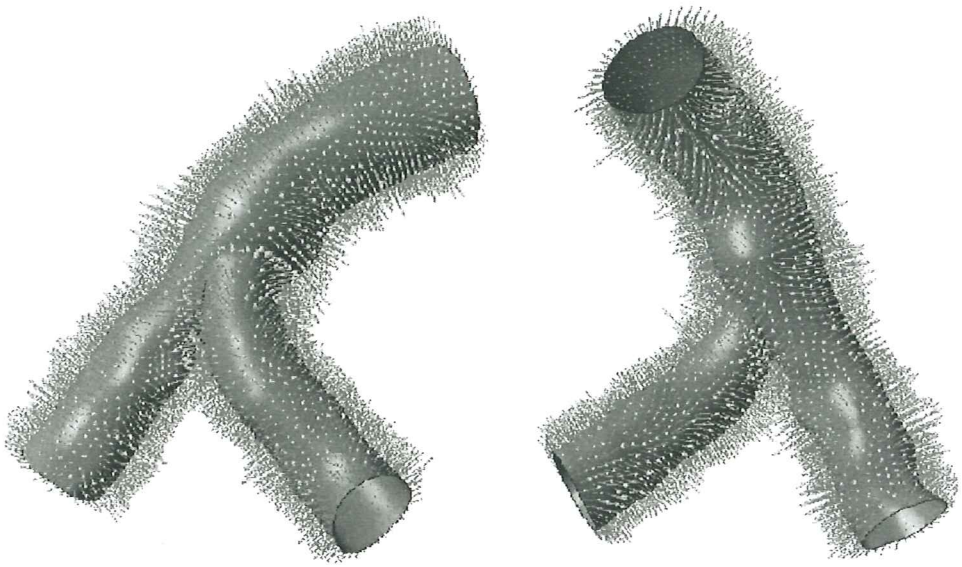


Fig. 2. Nodal forces on the inner wall

Wall stress computation. The model was processed by ABAQUS using triangular elements with three nodes `S3R`, producing a constant interpolation of the membrain and bending strains. Abaqus allows to consider variable element thickness using the option `*NODAL THICKNESS`, where every node can have different thickness. A hyperelelastic quasi-incompressible material (neohookean) was considered in the computation. See for more information (Bonet y Wood, 1997) and (Holzapfel, 2000).

Postprocess. Using GiD, it has been drawn the contour of scalar magnitudes as the pressure on the inner surface, the modulus of shear stress on the inner surface, the thickness of the arterial wall, the Von Mises stress on the inner surface, the Von Mises stress on the outer surface, the medium Von Mises stress in the thickness (Fig. 3) and the homogenity Von Mises stress in the thickness. Also, it was drawn vectorial fields as the shear efforts on the inner surface and the nodal forces on the inner surface (Fig. 2).

INTERFACING GID WITH FEAP

We have developed some configuration files in order to interface GiD and FEAP, in order to export calculation files from FEAP which are directly interpretated by FEAP in 2D and 3D problems. It has been defined:

Global settings. Title, number of load steps, maximum number of iterations, drawing scale, loads and displacements scale.

Materials. Type of materials and a set of mechanical parameters.

Conditions. Nodes and directions where displacements are imposed, displacement and force values. These conditions may be define over points, lines, surfaces and volumes (being this the preferred order). It is also possible to define the entities where nodal displacements (over points) and element stresses (over volumes) are to be controlled.

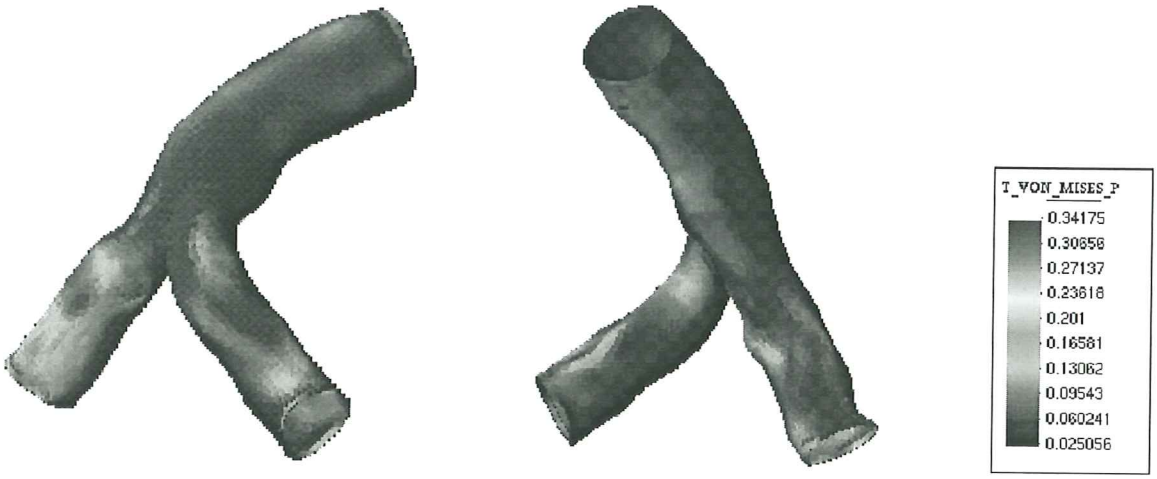


Fig 3. Contours of average Von Mises stress.

CONCLUSIONS

A biomechanics problem calculated by FLUENT and ABAQUS has been presented, and preprocessed and postprocessed by GiD (facilitating the task). In order to perform the analysis, it has been necessary to generate some configuration files for GiD in order to interface with ABAQUS. An interface of GiD with FEAP has also been indicated, making possible the automatic generation of the data file, and minimizing the introduction of mistakes.

ACKNOWLEDGEMENTS

The authors of this paper would like to thank *Grupo de Mecánica de Fluidos de la E.T.S.I. Industriales de la Universidad Politécnica de Madrid, Unidad de Hemodinámica del Hospital Clínico San Carlos de Madrid* and *Unidad de Cardiología Intervencionista del Hospital Meixoeiro de Vigo.* The authors would also like to thank the project *Modelado tridimensional y simulación de las arterias coronarias (MOTRICO), Programa Nacional de Tecnologías de la Información y Comunicaciones, Ministerio de Educación y Cultura.*

REFERENCES

Bonet, J. y Wood, D. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, 1997.

CIMNE. *GiD. Manual de utilización*, 1999.

Fluent, Inc. *FLUENT 5. Manual de usuario*, 2001.

Hibbit, Karlsson, y Sorensen. *ABAQUS User's manual*, 1998.

Holzapfel, G.A. *Non linear solid mechanics.* Wiley, 2000.

Taylor, R.L. *FEAP, a Finite Element Analysis Program*, 2000.

FINITEPOINT2D: A MESHLESS GiD CODE FOR LINEAR ELASTICITY

Enrique Escolano and Franco Perazzo
International Center for Numerical Methods in Engineering
Universidad Politécnica de Cataluña, Edificio C1
Gran Capitán s/n, 08034 Barcelona, Spain

Key words: Finite Point Method, Meshless Method, Elasticity, GiD

Abstract

The Finite Point method, a meshless formulation, is applied to the classic linear elastic 2D problem. This new method is implemented as a GiD problem type to automatically generate the clouds and visualize the results.

1. INTRODUCTION

The meshless methods are developed to solve the difficulties to generate the mesh for finite element methods, especially in 3D cases. The finite point method is based on a approximation of the unknown function, on a cloud around a star node, by least weighted squares. The differential equation to solve is sampled in each point to construct a linear system of equations, similarly to finite difference techniques. This method is fully meshless (it not requires a local auxiliary mesh for integration as other methods).

2. FUNCTION APPROXIMATION

The unknown function u is locally approach near a star node with a linear combination of base functions $[f_1,f_2,...,f_m]$. For this 2D case, the program use the monomial base with m=6, $[1,x,y,x^2,xy,y^2]$

$$u(x,y) \cong \hat{u}(x,y) = \sum_{i=1}^{m} \alpha_i \cdot f_i(x,y) \qquad (1)$$

The coefficients α_i are determined minimizing a summatory of square errors on the local nodes $\{p_1,..p_n\}$ of the cloud. The error is weighted by a function $\varphi(x,y)$ to enhance the precision near the star node.
The functional J to minimize respect α_i is

$$J = \sum_{j=1}^{n} \varphi(p_j) \cdot (u(p_j) - \hat{u}(p_j))^2 = \sum_{j=1}^{n} \varphi(p_j) \cdot (u(p_j) - \sum_{i=1}^{m} \alpha_i \cdot f_i(p_j))^2 \qquad (2)$$

Solving the normal equations linear system, the coefficients obtained are:
Notation: $\bar{\alpha} = [\alpha_1,\cdots,\alpha_m]^T$, $\bar{f} = [f_1,\cdots,f_m]^T$, $\bar{u} = [u(p_1),\cdots,u(p_n)]^T$

$$\bar{\alpha} = \underline{\underline{C}}^{-1} \cdot \bar{u} = \underline{\underline{A}}^{-1} \cdot \underline{\underline{B}} \cdot \bar{u} \qquad (3)$$

with

$$\underline{\underline{A}} = \sum_{j=1}^{n} \varphi(p_j) \cdot \bar{f}(p_j) \cdot \bar{f}(p_j)^T$$

$$\underline{\underline{B}} = [\varphi(p_1) \cdot f(p_1),\cdots,\varphi(p_n) \cdot f(p_n)]$$

3. CLOUDS

For each node, the approximation is local, only depends of the values on the cloud nodes (collection of n near nodes).
A cloud requires a minimum number of nodes n•m (dimension of the space base functions), and also a maximum for computational pourposes.
Another requirement is the non-singularity of the "A" matrix of the equation (3) (for example, a collection of aligned nodes is invalid).
A simple selection of the points inside a sphere centered at the star node is not generally a good choose.
Near the boundary, a cloud must not contain nodes physically unconnected with the star node. A strategy to construct the clouds from the list of all points can be consulted at [2].
In order to apply the boundary conditions for the EDP's it is also required the boundary normals.

The FPM is fully meshless, it only requires clouds and boundary normals, but in this problem type implementation, it can read a 2D mesh from GiD to build the clouds and normals.
To select nodes for the cloud, the edge-connected nodes are used: first the directly connected with the star node, and then the next level and so on until the desired number of nodes is reached.
For a proper version, GiD creates directly only nodes (and a boundary mesh for normals). It is a contradiction to develop a meshless method to bypass the effort of generate elements, and create the clouds/normals from an auxiliary mesh.

4. LINEAR ELASTICITY

In the Finite Point method, the differential equations are sampled by puntual collocation at each node, using for the displacement the approximation (1)

In this program a stabilized formulation is used to avoid ill-conditioned system of equations typical of puntual collocation. If the differential equations are:

$A(u_j)=0$ inside the domain Ω (4)

The stabilized formulation appends terms based in the real finite dimensions of the cloud. The stabilized form is:

$$A - \frac{1}{2} h_k \frac{\partial A}{\partial x_k} = 0 \qquad (5)$$

with h_k characteristic length parameter dimension ($k=1,2$ for 2D case)

For elasticity problems the stabilized equilibrium equations are:

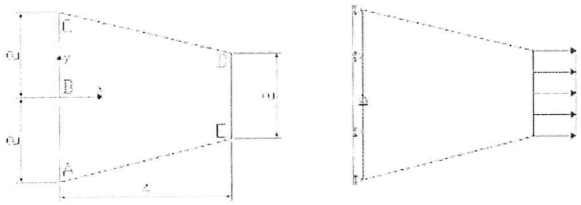
$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i - \frac{1}{2} h_k \frac{\partial}{\partial x_k}\left(\frac{\partial \sigma_{ij}}{\partial x_j} + b_i \right) = 0 \qquad \text{in } \Omega \qquad (6a)$$

$$u_j - \hat{u}_j = 0 \qquad \text{in } \Gamma_u \text{ (dirichlet)} \qquad (6b)$$

$$\sigma_{ij} n_j + t_i - \frac{1}{2} h_k n_k \left(\frac{\partial \sigma_{ij}}{\partial x_j} + b_i \right) = 0 \qquad \text{in } \Gamma_t \text{ (neumann)} \qquad (6c)$$

5. EXAMPLES

Source of the test: NAFEMS, linear statics benchmarks vol.1, October of 1987, test IC1.
This sample is a plane stress case, geometry dimensions are in meters and the boundary conditions are: AC locked in *x* direction and B locked in *y*.

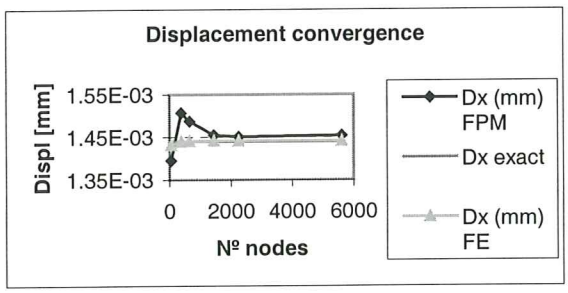


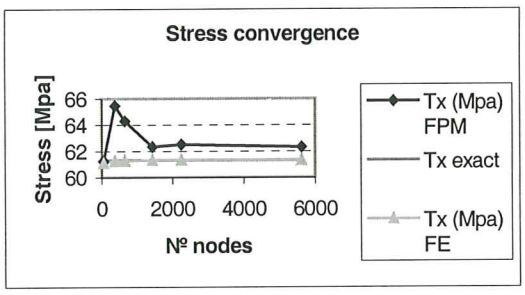
Material: Isotropic,
E=210103MPa, v=0.3,
thickness=0.1m
Aplicated charge: load of 10MN/m applied over DE border

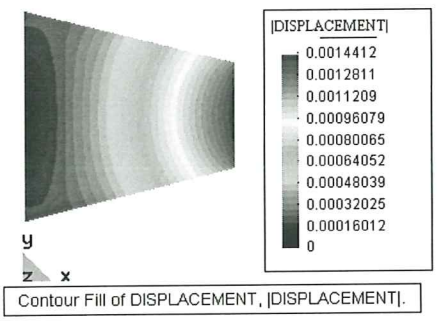
Wished result: U=1,44 mm in the mid of DE border - Sxx=61,3MPa in B

Obtained results:

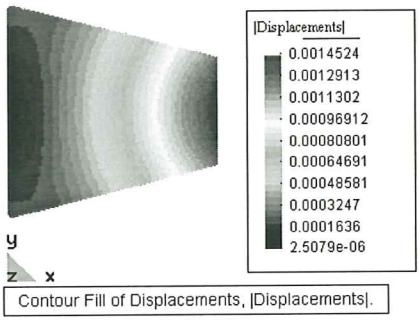
Nº nodes	Dx (mm) FPM	Dx (mm) FE	Tx (Mpa) FPM	Tx (Mpa) FE
54	0.001395	0.0014316	61.266	61.070
377	0.001507	0.0014398	65.471	61.240
652	0.001487	0.0014406	64.307	61.262
1432	0.001455	0.0014411	62.336	61.318
2246	0.001451	0.0014410	62.513	61.320
5614	0.001452	0.0014412	62.306	61.333







Contour Fill of DISPLACEMENT, |DISPLACEMENT|.



Contour Fill of Displacements, |Displacements|.

6. CONCLUSIONS

To evaluate the FPM, it has to be compared with the classical Finite Element Method (FEM), not only in accuracy of results but also in the practical and computational aspects.

The disadvantages of the FPM are:
- The generation of clouds from the nodes is complicated, and the "standard" preprocess programs actually are not ready for this requirement.
- Unsymmetrical stiffness matrix (the memory occupation increase, and some classical algorithms are disabled, for example, a direct cholesky decomposition can not be used to solve the linear equations system)
- ill-conditioned linear systems (convergence problems for iterative solvers as biconjuqate gradients and bad precision of some results)
- Construction of "Shape functions" more expensive (inversion of a $_{mxm}$ matrix).
- Cannot apply punctual forces (the formulation works directly with stresses) and cannot obtain reactions (the domain is diffuse without a mesh).
- Problems to sample the EDP if exist more than one material.
- Undefinition of required boundary normals in some edges or vertex (bad results near this boundary).
- The graphical postprocess is more limited without elements (cannot represent well defined contour fill of colours, integration on cuts, etc)

The main advantages are:
- Savings in the mesh generation (the main advantage of the method)
- Easy increment of the interpolation degree (only increasing the number of monomials of the base functions)
- Can work with high order derivatives of the unknown function (of interest when solving EDP's involving these derivatives)

Concluding, for a 2D elastic case the cost of the mesh generation is low, and the finite FEM is actually more competitive than the FPM, but it is of interest for a complicated 3D volume, or evolutive shape (big displacements, shape optimization, damage models, fluid dynamics, etc.).

7. REFERENCES

[1] E. Oñate, F. Perazzo and J. Miquel. A finite point method for elasticity problems. Computer and Structures 79 (2001).
[2] R. Löhner, C. Sacco, E. Oñate and S. Idelsohn - A Finite Point Method for Compressible Flow. *ECCOMAS 2000*, Barcelona, September (2000).
[3] M. Vericat. Análisis de estructuras en el campo elástico mediante métodos sin malla. Tesina de especialidad. UPC. Septiembre 2000.
[4] A. Huerta and S. Fernández. Coupling element free Galerkin and finite element methods. *ECCOMAS 2000*, Barcelona, September (2000).
[5] GiD Reference Manual. International Center for Numerical Methods in Engineering (CIMNE).

Modelización Numérica y Virtualización de Edificios Históricos. Interface GiD-Sap2000-VRML

Muñoz Salinas, Francisco. Maristany i Carreras Jordi

Abstract:

La modelización numérica de edificios históricos representa, actualmente, una herramienta prácticamente imprescindible para entender su comportamiento estructural, más si de lo que se trata es de asumir une estudio para su conservación y/o rehabilitación. Actualmente, la mayoría de los programas informáticos de simulación numérica de edificios arquitectónicos se dividen en tres partes: El Preproceso, el cual prepara toda la información necesaria (geometría, condiciones de borde, cargas, etc.); el Análisis, donde se desarrollan todos los cálculos y el Postproceso donde se analizan y toman decisiones en base a los resultados obtenidos en los pasos anteriores.
Es en este último punto donde el usuario de los programas de cálculo se encuentra con las mayores dificultados de interpretación de resultados. Unos resultados demasiado numéricos, con la interminable lista de números y valores no es lógicamente la mejor manera de abordar el problema y una salida gráfica de los mismos no está actualmente excesivamente desarrollada debido a la falta de memoria que tradicionalmente han venido sufriendo los ordenadores que se utilizan habitualmente para resolver esta clase de problemas. Por ello, es importante contar con herramientas tanto de hard como de soft que permitan visualizar los datos obtenidos. Actualmente, empieza a aparecer cantidad de programas de análisis estructural que tienen sus postprocesos muy avanzados, pero siguen teniendo limitaciones de visualización en el espacio: Operan solo en dos dimensiones. Por éste motivo, este trabajo pretende integrar los diferentes programas de cálculo que existen con los de representación gráfica en el espacio. De esta manera, la visualización de los resultados numéricos (valores tensionales, de esfuerzos y de deformación) puede ser mucho más interesante. Se trata de permitir a los usuarios navegar virtualmente a través de cualquier estructura en 3D y poder comprender su comportamiento. Resumiendo, esta investigación tiene como finalidad el desarrollar una interface entre los programas comerciales de cálculo de estructuras y los de representación gráfica y virtual en 3D, con aplicación práctica a edificios históricos. Al final se desarrolla un ejemplo practica particularizado en el Sagrario de la Catedral de la ciudad de México.

1. Introducción.

En la actualidad el estudiante de arquitectura se encuentra notablemente desconcertado para analizar la gran cantidad de información, numérica y gráfica, que lo proporciona los programas de cálculo.

Uno de los problemas que comporta el análisis de resultados a través de su lectura en formato de datos numéricos es de que se le hace muy difícil entender fácilmente las dimensiones del problema, aparte de tener que interpretar naturalmente cada uno de los valores que se presentan. Debido a esto, cada vez más investigadores apuestan por un sistema basado en los resultados gráficos.

Su representación, en la mayoría de los programas comerciales, pero, aun es muy poco interactiva y con limitadas herramientas de visualización.

De ello se deduce de que es importante que el universitario pueda contar con programas informáticos que le permitan visualizar, de manera ágil y sencilla, los resultados gráficos de cualquier análisis de estructuras arquitectónicas, y así, poder hacer una interpretación rápida y correcta de ellos.

2. Metodología para el análisis estructural de un edificio arquitectónico, a través de entornos gráficos.

En consecuencia, dentro de la construcción de un modelo existen tres procesos a desarrollar:

a. El primero, llamado *Preproceso*, en el cual se define todo lo relacionado a la geometría del edificio, y entrada de datos en general.

b. El *Análisis*, donde, dependiendo del tipo de problema, se aplican las distintas teorías de análisis estructural (sistema de barras, de placa, elasticidad bi y tridimensional, etc.).

c. Y por último el *Posproceso*, que es la manera de visualizar los resultados numéricos a través de gráficos.

Analicemos cada uno de ellos :

2.1 El *Preproceso*:

Su problema fundamental radica en la generación de un modelo numérico de la estructura a estudiar. Por esta razón, la modelización, para su cálculo posterior, puede llegar a ser compleja y esto crea la necesidad de utilizar programas de dibujo (*CAD*) para su realización..

En resumen, el ***preproceso*** es: la generación de entidades de dibujo de una geometría primitiva de la estructura. Esto significa, dibujar la estructura que se quiere estudiar y a su vez realizar una ***discretización*** suficiente de ella (generación de la malla).

Una vez que se tiene esta discretización, se graban los datos en un archivo, en nuestro caso con la extensión ****.dxf***, el cual, proporcionará todos los datos que sean necesarios para su posterior análisis : Nodos, coordenadas de los mismos, elementos de la malla, así como la conectividad entre ellos.

Estos datos serán posteriormente interpretados por un programa de Elementos Finitos, junto con sus respectivos datos estructurales tales como: materiales, condiciones de contorno, etc.

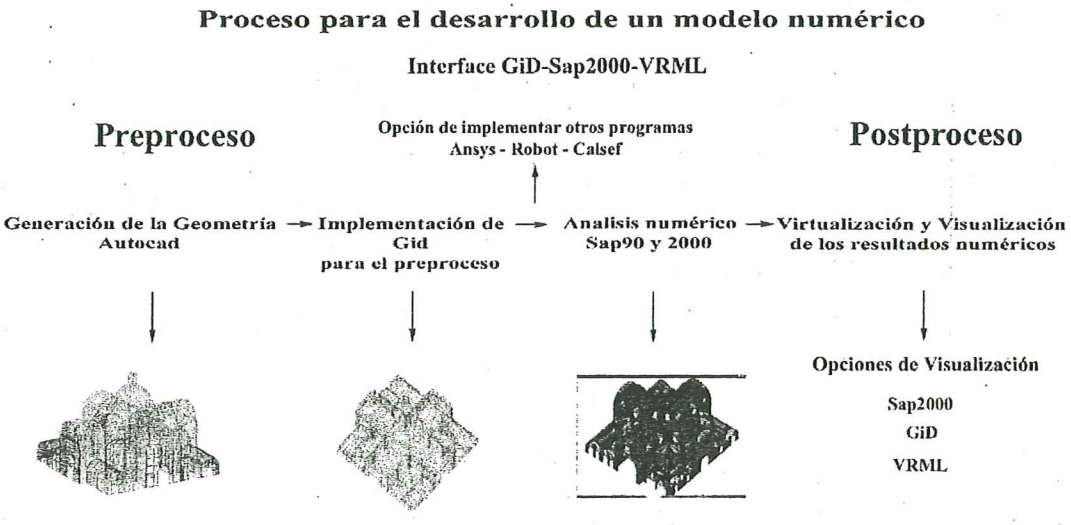


Fig. 1 -Metodología del Preproceso y Postproceso-

2.2 El Análisis:

En la actualidad, el Cálculo de Estructuras se realiza mediante programas informáticos cada vez más avanzados y sofisticados. Al mismo tiempo se implementan nuevos métodos de cálculo, que nos permiten conocer de forma ágil y fácil el comportamiento estructural de cualquier género de edificio.

Uno de ellos, el Método de los Elementos Finitos, es actualmente gracias a la informática, una de las herramientas más utilizadas para el cálculo de estructuras entre estudiantes, profesionales e investigadores.

Dentro del cálculo numérico, las estructuras de barras se pueden expresar a través de un número limitado de ecuaciones como seria el caso del conocido cálculo matricial. Desgraciadamente la mayoría de estructuras tienen un comportamiento continuo por lo que su representación partiendo de este número mínimo de estas ecuaciones obliga a simplificar notablemente el modelo con la consiguiente pérdida de exactitud de los resultados obtenidos.

Para poder realizar un análisis más aproximado de estas estructuras se precisa la integración de las ecuaciones diferenciales de equilibro de cada una de las partes que forman la estructura.

Dentro de las estructuras continuas más comunes en arquitectura, podemos encontrar muros de contención, depósitos de agua, cúpulas, forjados planos, etc. A pesar de que todas las estructuras continuas son tridimensionales existen algunos casos en los cuales se pueden describir sus comportamientos por medio de modelos unidimensionales (tal es el caso de las estructuras de barras), bidimensionales (elasticidad bidimensional, teoría de placas) o sólidos tridimensionales.

Actualmente, podemos afirmar que el Método de los Elementos Finitos, es una de las herramientas más potentes para el cálculo de estructuras uni, bi, o tridimensional.

Cualquier usuario puede entenderlo perfectamente , siempre y cuando tenga unos mínimos conocimientos básicos tanto de cálculo matricial de barras, informática y programación, así como de resistencia de materiales.

2.3 El *postproceso*

Una vez realizado el análisis numérico de la estructura a estudiar, el siguiente paso es la interpretación de los resultados. Para ello, los programas comerciales más utilizados (Sap2000, GiD) permiten una visualización gráfica de los valores tensionales, esfuerzos y deformaciones.

En esta investigación se ha desarrollado la interface *GiD-Sap2000-VRML*, que permite al usuario realizar un análisis numérico de cualquier estructura de forma sencilla y con una inter-relación con los programas facil y directa.

3. Programas comerciales de pre, cálculo y postproceso dentro del cálculo estructural.

3.1 El programa Sap2000

Sap2000 es un programa de análisis, elástico lineal y de segundo orden, de estructuras, por medio del método de los elementos finitos, que incluye un postprocesador gráfico para la presentación de resultados.

La preparación de datos para el desarrollo de un problema comprende básicamente:

1. *La descripción de la geometría estructural y de los materiales, así como sus condiciones de borde y datos generales.*

2. *La definición de los estados de carga para los cuales la estructura precisa ser analizada.*

El programa cuenta con un *pre* y *postproceso* mucho más completo que su anterior versión Sap90, además de implementar el cálculo sísmico. Sap2000 es, actualmente, el uno de los más utilizados entre los calculistas, investigadores y estudiantes.

3.2 El programa GiD

GiD es un programa de interface gráfico, que incide en el *pre y postproceso* y que se utiliza para definir, preparar e incluso visualizar todos los datos relacionados con la simulación numérica. Estos datos, permiten definir la geometría, los materiales, las condiciones de contorno, y toda la información que se precise para la etapa posterior del cálculo. De la misma manera, el programa puede generar mallas de Elementos Finitos y definir toda la información de simulación numérica en un formato. También es posible, dentro del programa, ejecutar la simulación numérica y visualizar la información de los resultados. Finalmente cabe recordar que implementa programas de dibujo bastante similares al Autocad, lo que aumenta significativamente su versatilidad.

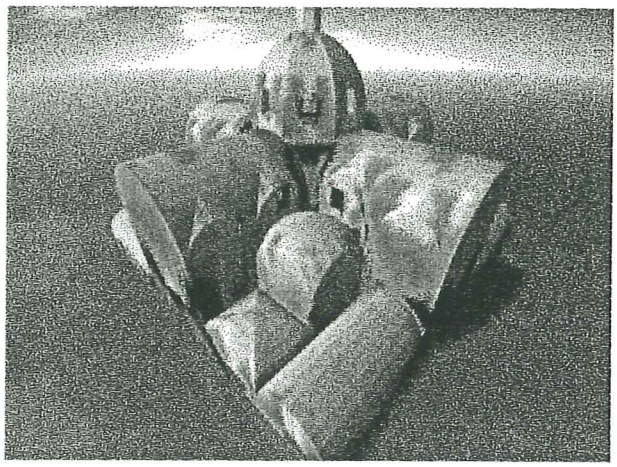


Fig 2 -Modelo numérico del Sagrario de la Catedral de México-

4. Interface GiD-Sap2000-VRML.

Gracias al avance de la informática, los programas comerciales para el cálculo de estructuras son cada vez más potentes y permiten, por tanto, analizar estructuras cada vez más sofisticadas.

Su versatilidad con programas estándar de dibujo (tanto Sap2000 como GiD pueden operar con formatos **.dxf* , por ejemplo) permite un ahorro considerable de tiempo, pero aun no está suficientemente desarrollada la compatibilidad entre los distintos programas de cálculo.

Por otra parte, por lo que respecta al postproceso, Sap2000 genera una representación gráfica muy completa, pero, con limitación de visualizaciones. GiD, por el contrario incluye herramientas de rotación dinámica, cortes de malla, así como la generación de archivos de imágenes en formatos distintos como los **.tif.* Pero ninguno de ellos, contempla la posibilidad de generar modelos virtuales de resultados gráficos.

De ello se deduce que es muy importante que los usuarios de estos programas puedan visualizar estos modelos.

Es en este sentido que aquí se presenta una interface el VRML, que permite generar modelos virtuales válida para cualquier cálculo numérico y que funciona integrando los programas comerciales más utilizados por las universidades e intentando asumir los últimos avances tecnológicos.

4.1 El VRML.

El *VRML*, es una interface virtual que toma los resultados numéricos generados por *Sap2000 i GiD* y lo traduce a un modelo, desde el cual se puede navegar gráficamente dentro y fuera de la estructura analizada.

2.1.1 información requerida por la interface VRML:

* Datos numéricos de la geometría del modelo.
* Definición de los distintos elementos y su conectividad entre ellos (malla)
* Desplazamientos.
* Resultados numéricos de Tensiones y Esfuerzos.

Toda esta información es procesada por un programa y traducida a un formato vrml.

El usuario, solo necesita, para visualizar el modelo, una versión actualizada de Netscape y de Cosmoplayer. A partir de los cuales podrá navegar dentro la de la estructura y visualizar los resultados numérico-gráficos.

La interface *VRML* se presenta como una herramienta muy potente en el ámbito de la percepción e interpretación de los resultados numérico-gráficos.

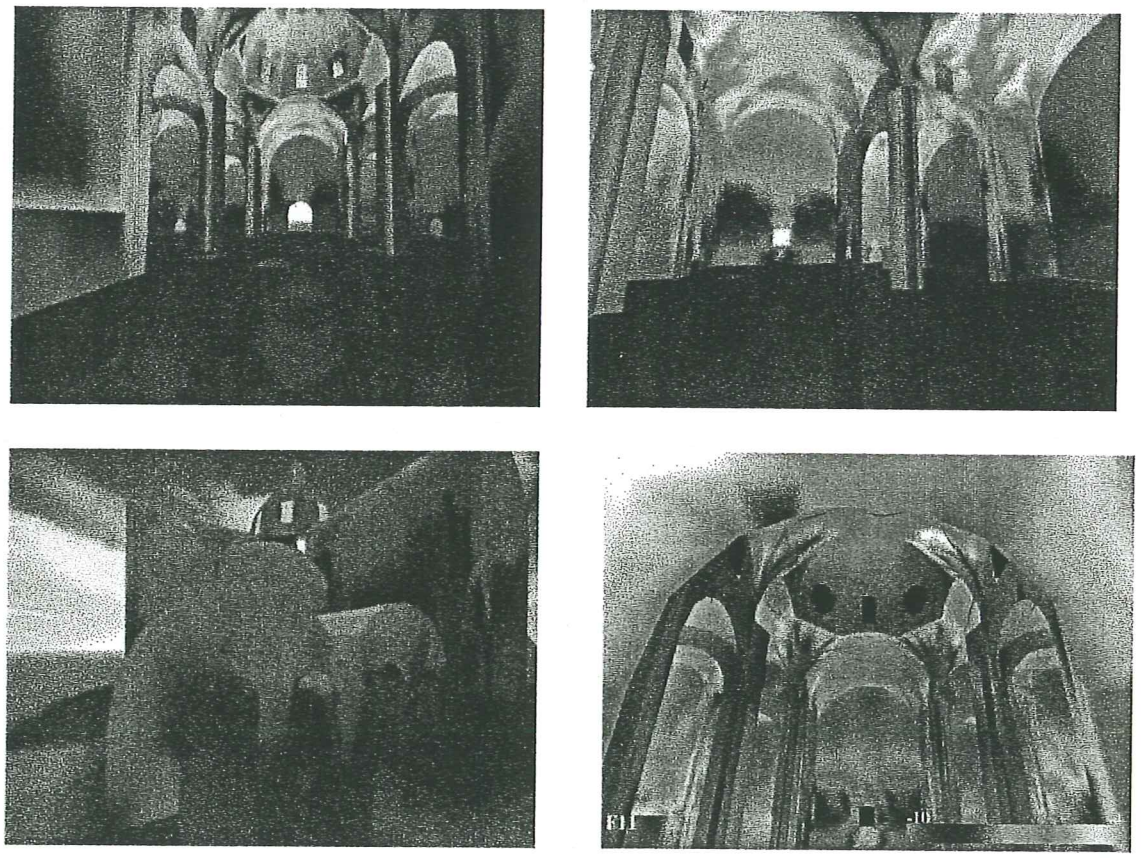


Fig. 3 -Virtualización numérica del Sagrario de la Catedral México-

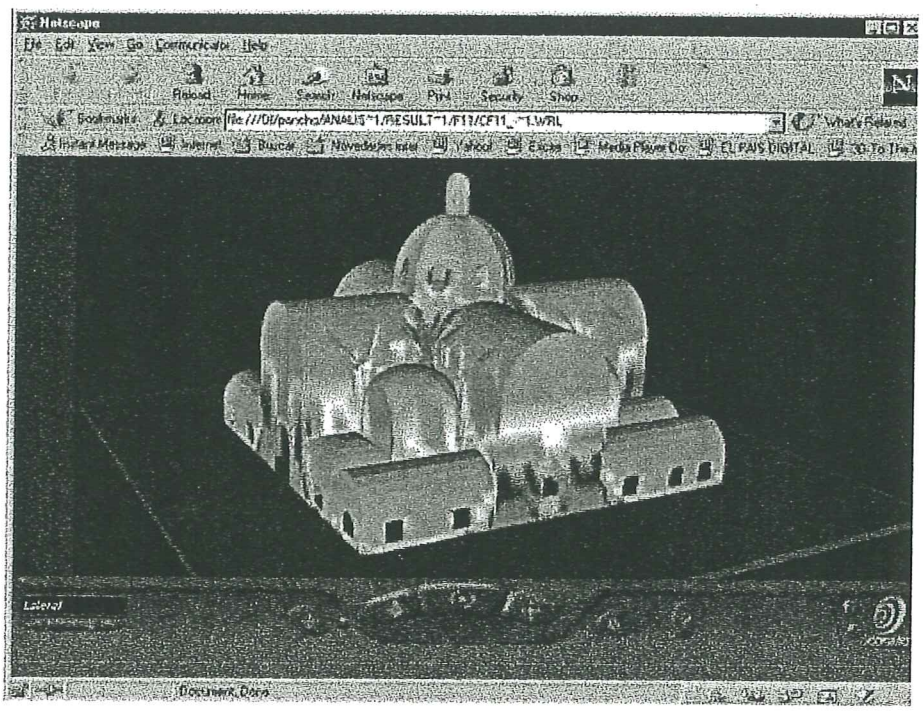


Fig. 4 -Virtualización numérica del Sagrario de la Catedral México por Internet-

Conclusión:

El espacio virtual muchas veces tiene más peso que el real ya que genera percepciones más complejas. Una interpretación adecuada de los resultados numéricos permite, al usuario, un conocimiento más profundo del comportamiento estructural de edificios arquitectónicos en general, así como el de los históricos en particular. Es por ello, que esta interface integra algunos programas comerciales, cubriendo la ausencia de herramientas más potentes en la visualización gráfica de los resultados numéricos, fundamentalmente en el campo de 3D.

Bibliografía:

GARCÍA DE ARANGOÁ, ANTONIO.
*Elasticidad
(Teoría y Experimentación), Fundamentos aplicaciones.
Editorial DOSSAT, S. A.
Madrid 1945.

HUGHES, THOMAS J. R.
*The finite element method linear static and Dynamic
Finite element analysis Thomas J: R: Hughes.
Publicacion : Englewood Cliffs, N. J: Prentice-Hall International.

LOVE. A. E. H.
*A Treatise on the Mathematical theory of Elasticity, Inc. New York, 1944

OÑATE, E., HANGANU, A. BARBAT, A., OLLER, S. , VITALIANI, R. Y SAETTA, A.
*Structural analysis and durability assesment of historical construccions
using a finete element damage model, structural analysis of historical
construccion, Editor: P. Roca, CIMNE, BARCELONA.

MUÑOZ SALINAS, FRANCISCO Y MARISTANY i CARRERAS, JORDI
Pre y Postprocesadores para el Cálculo de Estructuras.
Interface GID-Sap90.
CIMNE, 1999
Primera Edición

SAP 2000.
*Computer Software for
Structural & Earthquake Engineering.
Developed and written in U. S. A.

SEMINARIO INTERNACIONAL SOBRE ANÁLISIS ESTRUCTURAL DE CONSTRUCCIONES HISTÓRICAS.
*Posibilidades de las técnicas numéricas y experimentales
E.T.S.E.C.C.P., U.P.C.
Barcelona, 8, 9, 10 de Noviembre 1995.

TREATING THE GEOMETRY OF A FLOATING CAISSON TO OBTAIN A STRUCTURED MESH IN GiD

Benjamín Suárez[1], Antonio Foces[2], José Pereda[2] & Mariano Cacho[2]

[1] *International Center for Numerical Methods in Engineering (CIMNE), Edificio C-1, Campus Norte UPC, C/ Gran Capitán S/N, 08034 Barcelona, Spain.*
[2] *Departamento de Resistencia de Materiales de la Universidad de Valladolid, C/ Paseo del Cauce S/N, 47011 Valladolid, Spain.*

SUMMARY: An algorithm for the generation of the geometry of floating multicellular caissons used in ports and harbours facilities is shown. Data input to analyze these caissons with finite elements using structured meshing in the GiD program is generated. Once the basic dimensions that define the geometry of any caisson (length, breadth, depth, slab extents, diameter and degree of bevelled edges of the cells, thickness of throat and walls) are known, the algorithm divides the whole structure in appropriate small hexahedral volumes and generates the input data file. Afterwards, the GiD program makes the structured meshing of these volumes, obtaining therefore an accurate mesh of elements. Also a considerable saving of time in the generation of the mesh within the stage of analysis in the design process of this type of structures is attained.

KEYWORDS: Floating multicellular caisson, structured meshing, java-algorithm, object-oriented programming.

INTRODUCTION

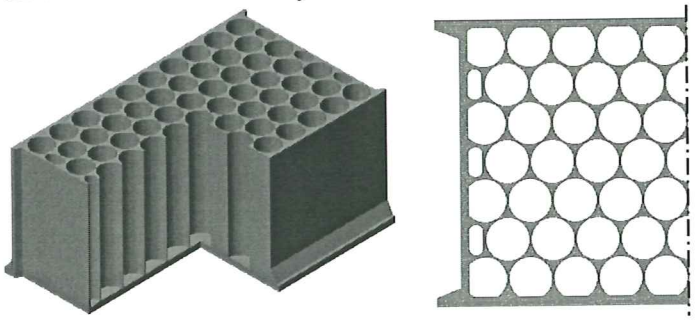
A fast construction of docks may be achieved with the use of floating caissons, which are prefabricated concrete box-like elements drilled with cylinder cavities or cells. Floating box caissons are manufactured upon a floating platform equipped with synchro lift for partially submerging the concrete structure while at the same time erecting it. Then it can be towed out to the final location, where it is then sunk by filling it with sand, gravel, or concrete. The different marine works and harbour constructions in which these caissons can be used include: ports, breakwaters, wharves, berthing facilities and docks, dry docks and slipways.

Description of the typical caisson geometry

In Figure 1.a a cut of a three-dimensional view of a typical floating caisson is shown. The caisson is usually formed by two basic elements. The first one, the slab, is a flat and massive element, with a depth of 0.6 to 1.5 meters that closes the structure bottom. The second one, the shaft, is a parallelepiped block strongly lightened by a series of circular cells arranged properly to gain the larger possible flotation, maintaining suitable values of resistance and stiffness. The usual

dimensions of the caisson oscillate between 20 and 30 m of length, 15 and 25 m of breadth (wide) and 12 to 20 m of draft (height).

In Figure 1.b a half of a generic cross-sectional section of the shaft is shown. The circular cells are disposed in a staggered formation to maximize the lightened surface. In the proximities of the lateral walls, where the use of circular cells would create problems of execution and excessive stress concentration in incoming angles, ovoidal cells are used (formed by a rectangle and two half circles). Also, when both the outer dimensions and the frontal and lateral walls thickness are fixed, sometimes it is necessary to bevel the cells next to these walls.

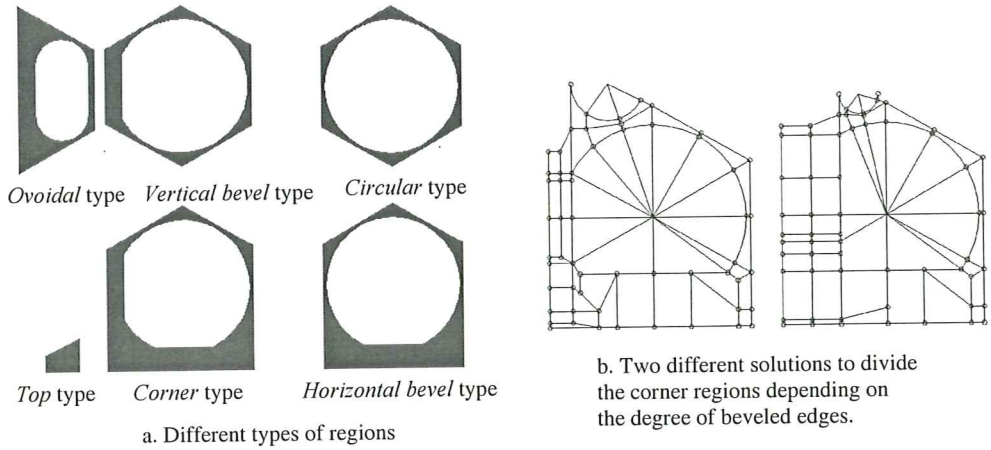


a. Cut from the caisson 3D view. b. Cross-sectional shaft section

Figure 1. Generic views of a typical floating caisson

Proposed algorithm and regions types that define the cross-sectional section

When trying to analyse the resistant behaviour of the caisson to diverse loads by means of the Finite Elements Method, it is necessary to make the meshing of its volume. Geometry is perfectly well-known, reason why a mesh generating program (as GiD) could easily achieve a mesh using tetrahedral elements. Nevertheless, given the peculiar characteristics of the shaft section, the resulting unstructured mesh would not be accurate and would contain a high number of elements.



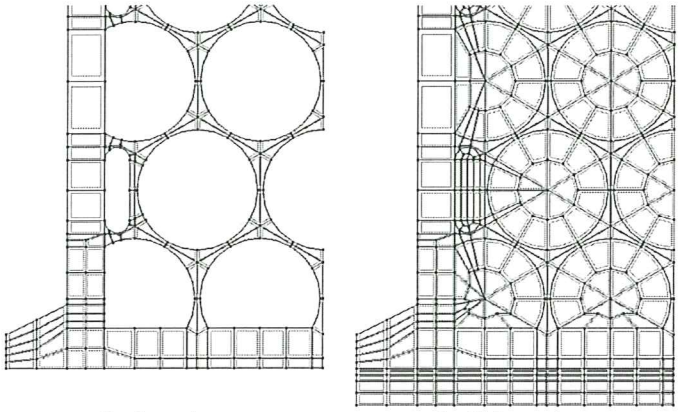
a. Different types of regions

b. Two different solutions to divide the corner regions depending on the degree of beveled edges.

Figure 3. Basic regions that define the cross-sectional shaft section

The proposed algorithm makes a previous geometry processing so that the structured meshing made in GiD gives a very accurate solution. Essentially, it divides the cross-sectional section in basic regions, grouping those of equal geometry in diverse typologies. Each type is treated separately, taking care of its peculiar characteristics. In Figure 3.a the basic regions in which the cross-sectional section of the shaft has been divided are shown.

Later, each region is divided in several zones of four sides (straight and/or curved), so when performing the structured meshing with hexahedra, GiD will perfectly locate these elements in these zones: it will only have to divide properly the zone to give a rough or fine mesh, as desired. In order to determine the zones, nodes must be located in the throats (zones of smaller thickness between adjacent cells), in which the stresses will be more critics. The possible variations of some parameters in each typology have been included in the algorithm, therefore allowing the treating of any degree of lateral and frontal bevelling (from 0 to 100%), symmetry, situation of the tops, etc. In Figure 3.b there is an example showing two possible situations, among others. Figure 4.a shows the result of applying the proposed algorithm to a shaft section.



a. Shaft section b. Slab section

Figure 4. Zonal division of the cross-sectional section

The algorithm allows to generate the points, lines, arcs, surfaces and volumes that define these zones, in such a way that the characteristic parameters that define these entities (coordinates, numeration, centres of gravity, etc) are written in a text file (with extension *.geo*) that can be read and processed directly by GiD.

In the massive prismatic slab, the meshing could be simpler, but to obtain the coupling of nodes and elements with those of the shaft, it is necessary to maintain the same meshing in the slab. Therefore, the same type of basic regions is kept, what implies the filling of the slab regions by making an inner division of the circles in diverse zones of four sides. The Figure 4.b shows the division made in the slab.

Figure 5 shows the meshed caisson by GiD with hexahedra elements. Lines have been divided in two, reason why eight hexahedra by each one of the volumes defined are generated.

Classes defined in the proposed algorithm

As the information that describes the caisson presents a clear hierarchic structure, the developed algorithm (programmed in the language JAVA) makes use of the philosophy of the object-oriented programming: An object is a container of the properties and the own variables as well as of the interrelations between the object and the others. In this way, diverse objects, which are instances of classes, are defined. Thus, the class Point contains variables like the numeration, coordinates and number of lines that connect it for a given point. The class Points_Collection groups a series of points that have something in common: they belong to the same basic region. In the same way, the class Line, contains the numeration and number of surfaces that connect it, and the classes initial Point and end Point that define a line. Following this scheme, Arc and

Surface classes are constructed with the respective collections of arcs and surfaces. The class Cell contains, for a given basic region, the different collections from points, lines, arcs and surfaces that define it. Finally, the Cells_Collection class contains all the basic regions that define a cross-sectional section. In the algorithm, two collections of cells are used, one for the shaft and one for the slab.

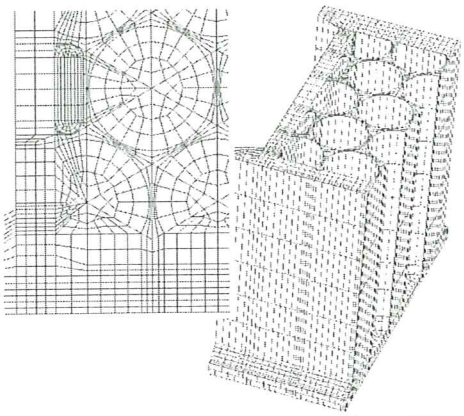


Figure 5. Meshed caisson by GiD

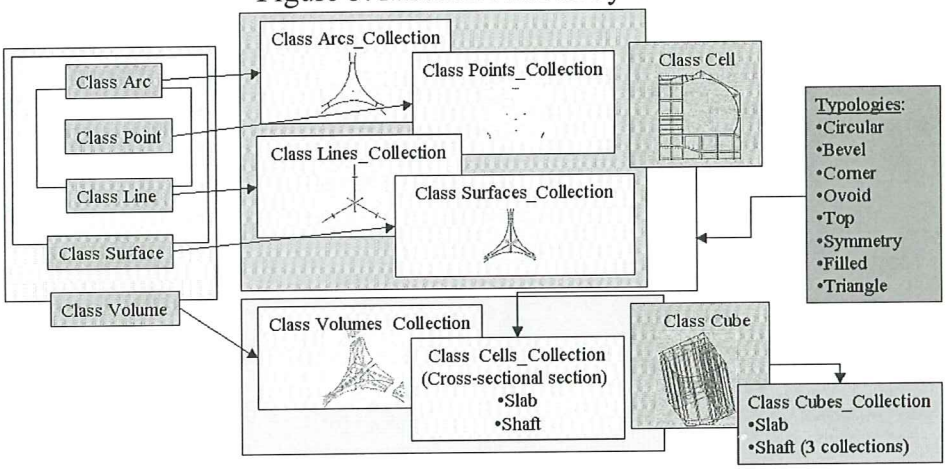


Figure 6. Scheme of classes used in the algorithm

Volume is the class defined from a surface of the collection of cells, the same one translated to a certain height, plus the four vertical resulting surfaces to connect both. The class Volumes_Collection contains all the volumes generated for a cross-sectional section. With the information stored in this collection and the Cells_Collection, the class Cube can be defined. It contains the points, lines, arcs, surfaces and volumes of a basic region. The class Cubes_Collection includes all the cubes between two cross-sectional sections. The algorithm uses one collection of cubes for the slab and up to four for the shaft, for the study of different sections at different heights, as well as to maintain a suitable factor of aspect in the element. Figure 6 shows schematically the relation between the different used classes.

References

1. Eric Comerma Piña, "Diseño automatizado de cajones multicelulares flotantes para obras porturarias", *Tesina de Especialidad*, Universidad Politécnica de Cataluña, Septiembre 1999.
2. Yoshimi Goda, "Random Seas and Design of Maritime Structures", World Scientific Publishing Co., 2000.

LA ASIGNACIÓN DE CONDICIONES SOBRE CAPAS EN LA INTERFAZ GRÁFICA DEL PROGRAMA STAMPACK

G. Socorro[1], E. Escolano[2], L. Neamtu[1],

[1]Quantech ATZ, Edificio NEXUS, Gran Capitán, 2-4, 08034, Barcelona, Spain
[2]CIMNE, International Center for Numerical Methods in Engineering, Barcelona, Spain.

SUMMARY: In order to optimise the functionality of the specialised friendly user interface, a new development in **GiD** programming has been introduced. The present paper is basically addressed to programmers interested in creating interfaces between **GiD** and a specialised numerical solver and deals with the particular aspect of how conditions can be assigned to layers from the graphical interface. Decrease in memory requirements and increase of the data file generation speed for industrial applications are expected. The developers of the commercial code for sheet stamping simulation processes **Stampack** are presently testing and improving the use of this new facility furthermore presented.

KEYWORDS: layer, condiciones, interfaz gráfica.

INTRODUCCIÓN

La asignación automática de condiciones (contactos, frenos, cargas, velocidades prescritas, etc.) sobre las entidades geométricas, es de vital importancia en los procesos de estampación de chapas metálicas.

La asignación de condiciones de la interfaz del programa **Stampack** se realiza sobre los elementos o nodos de las mallas, no sobre la geometría; por lo que se necesita que el sistema **GiD**, nos permita asignar condiciones solo a la capa "layer" en la que se encuentra dicha herramienta y no sobre todos los elementos o los nodos de una determinada herramienta. Esto permitirá aumentar la velocidad en la asignación de condiciones, a la vez que reducirá considerablemente la cantidad de memoria necesaria para asignar las condiciones, evitando con esto el desbordamiento de la memoria disponible.

En la simulación de los procesos de multi-estampación de chapas, se necesitan varias etapas (gravedad, movimiento del pisador por velocidad, estampación, recuperación elástica, corte, etc.) para obtener la forma final de la pieza, lo cual implica que hay varios intervalos con datos diferentes.

El archivo de datos que se genera es muy grande, con lo cual el programa tarda bastante para escribirlo; de lo anterior se infiere la necesidad de implementar nuevos comandos, que disminuyan el tiempo de escritura y permitan estimar el tiempo que tarda en escribir cada uno de los grupos de datos.

La asignación de condiciones específicas sobre capas debe poderse realizar desde la aplicación gráfica de forma automática.

IMPLEMENTACIÓN Y USO

En versiones de **GiD** anteriores a la 6.1 las condiciones sólo se podían aplicar sobre entidades geométricas del modelo (puntos, curvas, superficies, volúmenes, nodos y elementos).

Para incorporar la nueva condición aplicada sobre una capa, se han modificado algunas estructuras y funcionalidad internas del sistema, así como la caja de diálogo de asignación de condiciones.

Para que la nueva prestación sea totalmente funcional, debería tener las siguientes posibilidades: a) creación y asignación de nuevas condiciones; b) asignar una o más de una condición del mismo tipo a una capa (opción canrepeat del fichero *.cnd); eliminación de condiciones; d) listado de condiciones (capa asignada y las propiedades) ; e) visualización de las condiciones (se dibujarán todas las entidades de geometría o malla que estén contenidas en la capa); f) acceso a la información de las capas existentes, así como a las condiciones aplicadas sobre capas desde el fichero de plantilla *.bas (plantilla que interpreta **GiD** para escribir el fichero de datos para realizar la simulación); g) permitir bucles sobre nodos o elementos, pasando sólo por las entidades contenidas en las capas designadas previamente; h) estimar los tiempos de escritura en el archivo de plantilla.

La sintaxis en el fichero *.cnd, CONDTYPE debe especificarse "over layer", por ejemplo:

```
BOOK: Geometry
NUMBER: 1 CONDITION: Geom-Set
CONDTYPE: over layer
QUESTION: Number:
VALUE: 1
QUESTION: Set_Name:
VALUE: Blank
QUESTION: Layer_Name:
VALUE: Sheet
END CONDITION
```

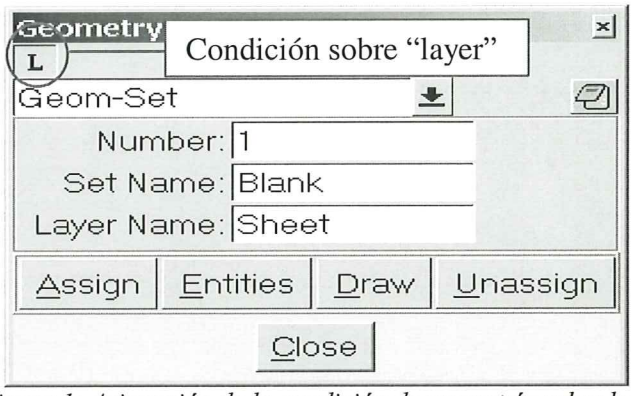


Figura 1: Asignación de la condición de geometría sobre la chapa "Sheet".

Al cargar el tipo de problema, en la caja de diálogo de condiciones, se mostrará un icono con una "L", indicando que la condición "Geom-Set" se aplica sobre una capa (véase la Figura 1).

En el fichero *.bas se han añadido las instrucciones siguientes: a) **loop layers* – permite hacer un bucle sobre las capas existentes; b) **LayerNum* – devuelve el nº ordinal de una capa; c) **LayerName* - devuelve el nombre de la capa. Se debe notar que *LayerNum* y *LayerName* deben estar dentro de un bucle sobre las capas. Dentro de un bucle sobre nodos se puede acceder a las propiedades de su capa mediante: a) **NodesLayerNum*; b) **NodesLayerName*. Análogamente, en un bucle de elementos se puede consultar: a) **ElemsLayerNum*; b) **ElemsLayerName*.

Se pueden recorrer las capas que tienen asignada una condición "over layer", y acceder al valor de sus campos, por ejemplo:

```
*set cond Geom-Set *layers
*loop layers *OnlyInCond
  Geometry set properties: *Cond(1)  *Cond(Set_Name)  *Cond(3)
```

*end layers

EJEMPLO DE APLICACIÓN

Para realizar la validación de las nuevas prestaciones se ha seleccionado el ejemplo del proyecto de la conferencia de NUMISHEET'99 (proceso de multi-estampación). Este proyecto presenta las siguientes etapas: 1) Blankholder velocity; 2) Blankholder force; 3) Forming; 4) Cutting; 5) Blankholder velocity; 6) Blankholder force; 7) Forming; 8) Blankholder velocity; 9) Blankholder force; 10) Forming; 11) Springback. En la Tabla 1, se presenta el tipo de problema Stampack_CondLayer.gid que escribe el bloque de los sets de geometría (las coordenadas de todos los nodos del modelo).

Fichero Stampack_CondLayer.bas

```
*Set var geooper=operation(IntvData(Geometry_SETs,int)/3)
*if(geooper>0)
*Set var geoadd=0
*Set var aux=0
*for(i=1;i<=geooper(int);i=i+1)
*Set var base=operation((i-1)*3)
*Set var p0=IntvData(Geometry_SETs,*operation(base+1),int)
*if(p0==1)
*if(aux==0)
GEOMETRY_DEFINITION
*set var aux=1
*endif (aux==0)
*Set var geoadd=operation(geoadd+1)
*endif (p0==1)
*end for(i=1;i<=geooper(int);i=i+1)
*if(geoadd>0)
   GENERAL:  GSCALE = 1.0 *\
*Set elems(All)
*Set Cond Geom-Set *layers *NoCanRepeat
*for(i=1;i<=geooper(int);i=i+1)
*Set var base=operation((i-1)*3)
*Set var p0=IntvData(Geometry_SETs,*operation(base+1),int)
*if(p0==1)
*Set var p1=IntvData(Geometry_SETs,*operation(base+2),int)
    $ *IntvData(Geometry_SETs,*operation(base+3))
   SET = *p1
*Loop layers *OnlyInCond
*set var NumLayer(int)=LayerNum
*if(p1==cond(1,int))
*Loop nodes
*format "%10i%+14.5e%+14.5e%+14.5e"
*if(NUMLAYER==NodesLayerNum)
*NodesNum *NodesCoord()
*endif (NUMLAYER==NodesLayerNum)
*end loop nodes
*endif (p1==cond(1,int))
*end Loop layers *OnlyInCond
   END_SET
*endif (p0==1)
*end for (i=1;i<=geooper(int);i=i+1)
END_GEOMETRY_DEFINITION
*endif (geoadd>0)
*endif (geooper>0)
```

Fichero Stampack_CondLayer.prb

```
PROBLEM DATA
END PROBLEM DATA
INTERVAL DATA
TITLE: Geomerty sets
QUESTION:
Geometry_SETs:(ACTION,Number,ToolName)
VALUE: #N# 3 1 1 Blank
END INTERVAL DATA
```

Fichero Stampack_CondLayer.mat

```
NUMBER: 1 MATERIAL: GenericName
QUESTION: Property
VALUE: 1.0
END MATERIAL
```

Fichero Stampack_CondLayer.cnd

```
BOOK: Geometry
NUMBER: 1 CONDITION: Geom-Set
CONDTYPE: over layer
QUESTION: Number:
VALUE: 1
QUESTION: Set_Name:
VALUE: Blank
QUESTION: Layer_Name:
VALUE: Sheet
END CONDITION
BOOK: Loads
NUMBER: 2 CONDITION: Point-Load
CONDTYPE: over points
CONDMESHTYPE: over nodes
QUESTION: Set_load_number:
VALUE: 1
QUESTION: X-Force:
VALUE: 0.0
END CONDITION
BOOK: Contacts
NUMBER: 3 CONDITION: Contact
CONDTYPE: over layer
CANREPEAT: YES
QUESTION: Number_of_surface:
VALUE: 1
END CONDITION
```

Tabla 1: Definición del tipo de problema Stampack_CondLayer.gid para escribir las coordenadas nodales.

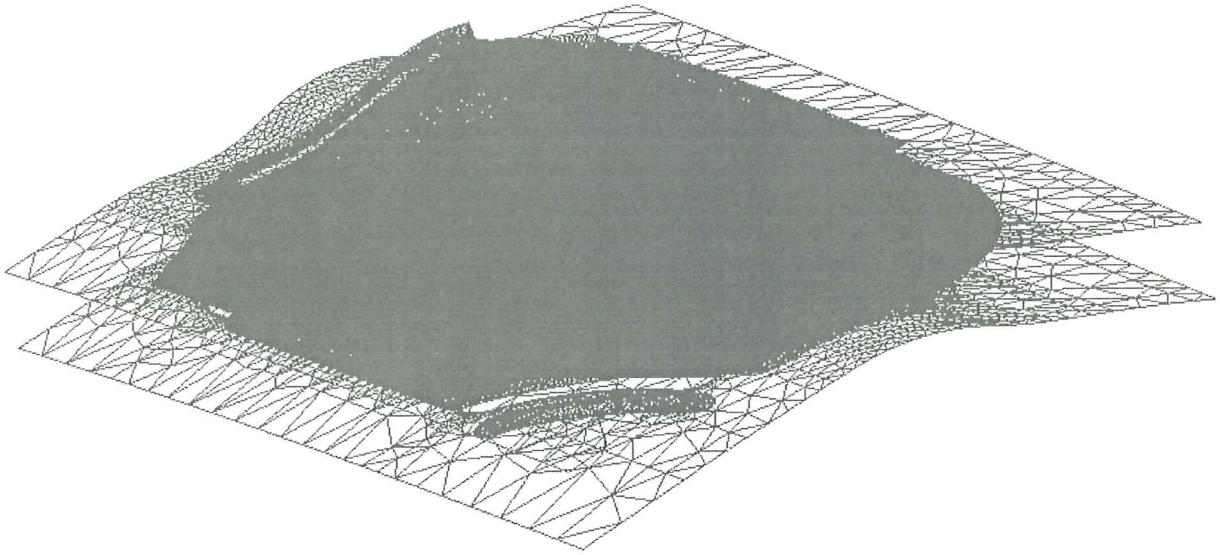


Figura 2: Malla de elementos finitos.

El proyecto presenta un total de 20 capas, 82 796 nodos y 250 000 elementos (véase la Figura 2); el archivo de datos ocupa un espacio de disco de 48 MB.

En la versión anterior de la interfaz aplicando la condición de geometría sobre todos los nodos, el programa tardaba aproximadamente 1 hora para escribir las coordenadas nodales, ahora con la utilización de las capas escribe dichas coordenadas en 5 minutos.

CONCLUSIONES

Esta nueva opción permite un ahorro considerable de memoria a la hora de asignar las condiciones, pues en lugar de asignar una condición sobre todos los nodos o elementos, ésta se asigna sólo una vez sobre la capa; esto reduce notablemente el gasto de memoria al aplicar condiciones con valores repetidos en múltiples entidades.

La implementación de los comandos (**time*, **clock*) para estimar el tiempo en la plantilla *.bas, permite analizar cuales son las algoritmos que más tiempo consumen y optimizarlos.

Para poder utilizar este nuevo esquema se requiere una importante modificación de interfaz del tipo de problema.

REFERENCIAS

1. GiD Reference Manual, version 6.0, CIMNE, Barcelona, Spain.
2. Proceedings of NUMISHEET'99, The 4th International Conference and Workshop on Numerical Simulation of 3D Sheet Forming Process, 13-17 September 1999, Besancon – France.

A methodology for the constrained optimization of tunnels with GiD

Lluís Gil and Antoni Andreu
Email: lluis.gil@upc.es
Dept. of Strength of Materials and Engineering Structures
Universitat Politècnica de Catalunya
Mòdul C1, Campus Nord UPC
c/ Gran Capitán s/n
08034 Barcelona. Spain

Abstract

In this paper we merely want to explore the possibilities of GiD as a powerful interface to deal with complex optimization problems. A general methodology is developed to incorporate GiD as an external aid-software. The proposed methodology has two main steps. First the problem statement must be built expressing explicit relationships between design variables and objective functions and constraints. In order to do that, for whatever problem you arise, we use interpolation schemes over a collection of different designs. The second step is the minimization of the problem using any relevant technique you can apply. Both steps can be repeated in an iterative process to find a minimum. The optimization of tunnels is a good example to evaluate the possibilities of this methodology. Moreover it can be extended to other types of problems like fluids, thermal, etc.

1 Introduction

The goal of the optimization of structures is to find values for some design variables:

$$\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \tag{1}$$

Minimizing an objective function:

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) \tag{2}$$

And satisfying, at the same time, some design constraints:

$$g(\mathbf{x}) = g_i(x_1, x_2, \ldots, x_n) = 0 \quad i = 1 \cdots m \tag{3}$$

Constraints are almost referred to a maximum allowable stresses or displacements according to normative and material capabilities. It is usual to express constraints as inequalities; nevertheless all inequalities can be converted in expressions like (3) with the help of slack variables [1]. Normally constraints are evaluated using the solution of the boundary problem of the structure. The equilibrium equation can be written as [2], [3]:

$$\int_V \delta\boldsymbol{\varepsilon}\ \boldsymbol{\sigma}\ dV = \int_V \delta\mathbf{u}\ \mathbf{b}\ dV + \int_{\partial V} \delta\mathbf{u}\ \mathbf{t}\ dS \tag{4}$$

Where $\delta\varepsilon$ refers a virtual deformation, σ internal stresses of the structure, δu a field of virtual displacements and b,t external loads over volume and surface. Finite Elements Method FEM is a common technique that solves equation (4).

In this paper we want to optimize the inner reinforcement of a tunnel with an internal shell of concrete and steel. In particular, we want to optimize the volume of concrete and we have constraints in stresses and in the geometrical shape definition. This problem is complex but we simplify it avoiding some anchoring problems [4], [5] and material anysotropies [6]. To increase complexity neither objective function nor constraints depend in explicit manner from the design variables in this problem.

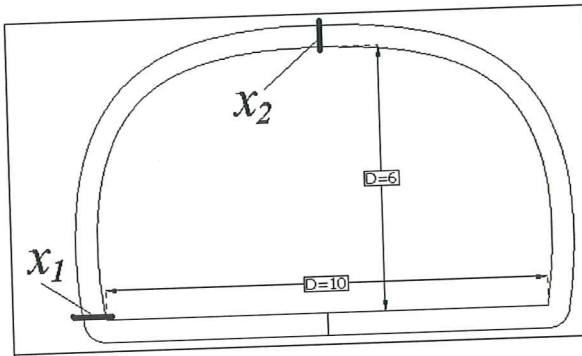


Fig. 1. Design variables and geometry

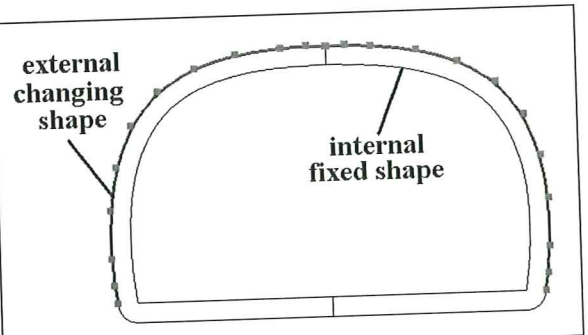


Fig. 2. Spline interpolation for external contour. Internal one is fixed.

2 Methodology

The methodology has two main steps. The first one builds an analytical expression of the problem and the second one minimizes the problem.

2.1 First step: Building the problem

Design variables are the base thickness x_1 and the crown x_2. The objective function is the area of concrete. This area A must be minima. Area depends on crown and thickness, however it is difficult to write an analytical expression because the external boundary was defined using splines with GiD. With this definition when these design variables change a new design is achieved with GiD creating a homothetic curve. Hence, objective function will be interpolated for simplicity:

$$f(x_1,x_2)=A(x_1,x_2) \tag{5}$$

In this problem we define three constraints: Maximum compression stresses in concrete, maximum displacement in the crown and minimum value for thickness. Neither stresses σ nor δ displacements depend explicitly on design variables, thus an interpolation scheme must also be done for them:

$$g_1(x_1,x_2)=\sigma(x_1,x_2)-\sigma_{\max}\leq 0 \tag{6}$$

$$g_2(x_1,x_2)=\delta(x_1,x_2)-\delta_{\max}\leq 0 \tag{7}$$

The last constraint is a minimum requirement for the base thickness, thus:

$$g_3(x_1) = x_1 - x_1^{\min} \leq 0 \tag{8}$$

Where $\sigma_{\max} = 1500\,\text{N/cm}^2$, $\delta_{\max} = 1\,\text{cm}$ and $x_1^{\min} = 0.05\,\text{m}$.

Then we interpolate [7] three quadratic polynomials for the definition of objective function (9) and constraints in stresses (10) and displacements (11):

$$A(x_1, x_2) = a_1 x_1^2 + a_2 x_2^2 + a_3 x_1 + a_4 x_2 + a_5 x_1 x_2 + a_6 \tag{9}$$

$$\sigma(x_1, x_2) = b_1 x_1^2 + b_2 x_2^2 + b_3 x_1 + b_4 x_2 + b_5 x_1 x_2 + b_6 \tag{10}$$

$$\delta(x_1, x_2) = c_1 x_1^2 + c_2 x_2^2 + c_3 x_1 + c_4 x_2 + c_5 x_1 x_2 + c_6 \tag{11}$$

Every interpolation has 6 unknowns a_i, b_i, c_i and according to the strategy six different problems must be defined and solved.

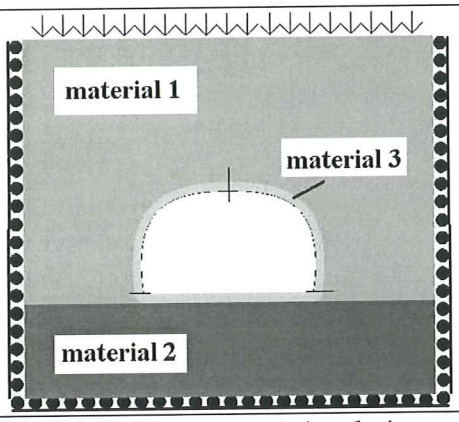


Fig. 3. Model FEM simulation

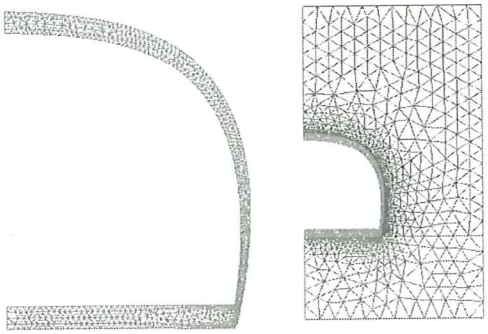


Fig. 4. Mesh generation

In this case we use the commercial software GiD [8], a mesh generator and post-processor, and the FEM analysis code Calsef [9]. The simulation computes a plane strain problem. Soil and concrete properties are in table 1. There was also a load on the top of 100 KN/m (see figure 3).

Table 1. Material properties for simulation

	Material 1: upper soil	Material 2: bottom soil	Material 3: concrete
Density	21 KN/m2	23 KN/m2	25 KN/m2
Young modulus	3.0 10^4 KN/m2	3.0 10^6 KN/m2	30.0 10^6 KN/m2
Poisson coefficient	0.35	0.35	0.25

From the results of programs we built data table 2.

Table 2. Design variables and constraint values for each design

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Thickness x_1 (m)	0.1	0.25	0.25	0.5	0.5	0.7
Crown x_2 (m)	0.7	0.25	0.5	0.5	0.5	0.7
Area A (m^2)	4.018	2.991	3.95	4.184	5.17	4.44
Stress σ (MPa)	13.61	8.86	7.71	8.49	6.71	10.48
Displacement δ (cm)	0.74	1.323	0.851	1.103	0.712	1.408

Next we found the constants a_i, b_i, c_i from three system of 6 equations (12), (13) and (14). We solved each system of equations with a simple sheet of Excel ®:

$$\begin{bmatrix} A_1 \\ \vdots \\ A_6 \end{bmatrix} = [\mathbf{M}] \begin{bmatrix} a_1 \\ \vdots \\ a_6 \end{bmatrix} \qquad \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_6 \end{bmatrix} = [\mathbf{M}] \begin{bmatrix} b_1 \\ \vdots \\ b_6 \end{bmatrix} \qquad \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_6 \end{bmatrix} = [\mathbf{M}] \begin{bmatrix} c_1 \\ \vdots \\ c_6 \end{bmatrix} \qquad \text{(12), (13) and (14)}$$

Notice that the system matrix **M** (15) is always the same for each interpolation function. This reduces very much the computational effort. However, if you employ different interpolations for objective function and constraints, the system matrix would be different for each one of them.

$$[\mathbf{M}] = \begin{bmatrix} x_{11}^2 & x_{21}^2 & x_{11} & x_{21} & x_{11}x_{21} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{16}^2 & x_{26}^2 & \cdots & \cdots & x_{16}x_{26} & 1 \end{bmatrix} \qquad (15)$$

2.2 Second step: Minimization strategy

Once we established the problem we started the minimization. In this case the selected technique was penalization [10], [11]. We used Matlab® and wrote a short minimization code to optimize the structure. Penalization converted the constrained problem to an unconstrained one adding the penalized constraints in the objective function, using equations (5)-(8) we built:

$$f(x_1, x_2) = A(x_1.x_2) + \alpha_1 g_1(x_1.x_2) + \alpha_2 g_2(x_1.x_2) + \alpha_3 g_3(x_1) \qquad (16)$$

Penalization technique has the advantage of flexibility because the modification of penalty parameters can help to reach the minimum. The values of penalty parameter $\alpha_1, \alpha_2, \alpha_3$ ranges from 0.01 to 10000.

2.3 Iterative procedure

After solving the minimization problem one solution was found. This solution replaced the worse of the first six designs. Then a new problem statement was defined incorporating this new design and it was minimized again. This iterative technique ended when the solution verified constraints and the improvement in the original objective function (without penalty) was less than 5%. After two trials the final solution was reached, see table 3.

Table 3. Values of the optimum for each iteration

	Thickness x_1 (m)	Crown x_2 (m)
First iteration	0.05	0.602
Second iteration	0.146	0.47

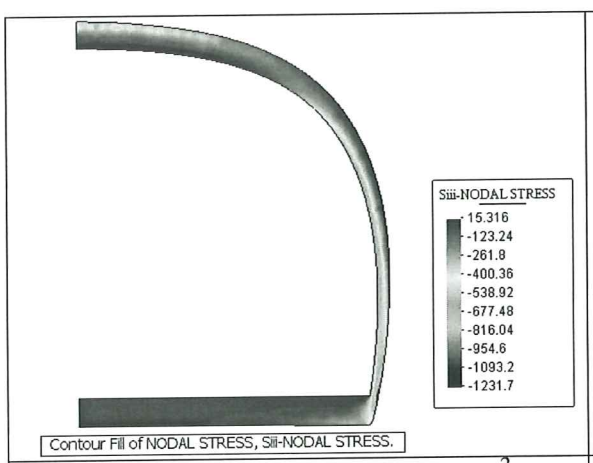


Fig. 5. Stress results (x 10 KN/m^2)

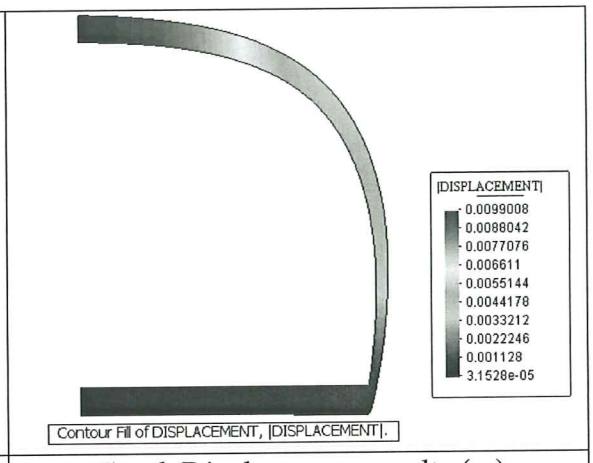


Fig. 6. Displacement results (m)

4 Conclusions

This paper presented a robust and flexible approach to deal with optimization problems for tunnel reinforcement. The proposed methodology takes advantage of GiD and Calsef commercial software. Other common tools like Excel ® and MatLab ® were also incorporated in the optimization process on demand.

The proposed formulation performs well when the objective function or some of the constraints are difficult to write in an explicit manner with respect the design variables. These functions can be interpolated from some desired base of functions. In the example of this paper second order polynomials were used to interpolate objectives and constraints. Nevertheless other interpolation schemes like trigonometric functions, splines, exponential functions, etc. could also be used. The choice for the interpolation basically depends on the nature of the problem, thus unfortunately no universal rule can be defined.

Another advantage of the methodology is that minimization technique is also a free choice. In this example penalty functions performed well, however another technique could be used. Notice that penalty functions does not force the accomplishment of the constraint, only penalizes, thus a quasi-solution near of the real minima can be easy to find. The possibility of adjusting the penalty coefficients gives the chance to manipulate the importance of some particular constraints, relaxing or forcing one or other. This manipulation facility could be considered an inconvenience, because of the sensitivity to the penalization parameters, but on the contrary it must be faced as an advantage. From this point of view, penalty technique is more flexible than other techniques.

We expect that the proposed methodology can be applied to other range of engineering problems like thermal, fluids, mechanics, etc.

5 Acknowledgements

This work has been developed under contract program TIC1999-0762-C02-02 CICYT (Spain).

6 References

[1] Vanderplaats, G.N. *Numerical optimization techniques for engineering design*. Mc Graw-Hill. 1984.

[2] Segerlind, L. J. *Applied finite element analysis*. John Wiley and Sons. 1984.

[3] Zienkiewicz, O. C. and Taylor, R. L. *The finite element method.* Vol 1. Mc Graw-Hill. 1989.

[4] Yin, L.Z., Yang, W. and Guo, T.F. "*Tunnel reinforcement via topology optimization*". Int. J. Num. Analytical Met. Geomechanics. 24 (2). 201-213. Feb. 2000.

[5] Yin, L.H. and Yang, W. "*Topology optimization to prevent tunnel heaves under different stress biaxialities*". Int. J. Num. Analytical Met. Geomechanics. 24(9). 783-792. Aug. 2000.

[6] Yin, L.Z. and Yang, W. "*Topology optimization for tunnel support in layered geological structures*". Int. J. Num. Met. Eng. 47 (12). 1983-1996. Apr. 2000.

[7] Ralston, A. and Rabinowitz P. *A first course in numerical analysis.* 2nd. Edition. Dover. 2001.

[8] CIMNE. *GiD the personal pre and post processor*. CIMNE. http://gid.cimne.upc.es.

[9] Oñate, E. and Botello, S. *Calsef a program for the linear static analysis of structures*. 1992. CIMNE. http://www.cimne.upc.es.

[10] Nocedal, J. and Wright, S.J. *Numerical optimization.* Spinger-Verlag. 1999.

[11] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. *Numerical recipes in C. The art of scientific computing.* Cambridge University Press. 1996.

From GiD to Market: COMPASS experiences in developing GiD-based RamSeries and Tdyn programs

Julio García-Espinosa
Compass Ingeniería y Sistemas
Barcelona, Spain
info@compassis.com
www.compassis.com

Abstract

GiD [1] is a pre/postprocessing system, specially designed as an universal and adaptative user interface for geometrical modelling, data input, meshing and visualisation of results for all types of numerical simulation programs.
GiD includes several tools for customising input and output files formats, menus and visualisation options, but 'where is the limit (if any) in this adaptation process?' or 'is it possible to create GiD-based, professional and fully integrated new software products?'
This document shows the experiences of the company Compass Ingeniería y Sistemas in performing GiD integration and industrialisation of the CFD and CAE solvers, Tdyn and RamSeries [2], and tries to answer those question.

References

[1] *GiD website*: http://gid.cimne.upc.es. GiD program is available to download. Reference manuals and technical information of this program are also available.

[2] *Compass website*: http://www.compassis.com. Tdyn and RamSeries programs are available to download. Reference manuals and technical information of these programs are also available.

[3] *Tcl-Tk* website. http://tcl.activestate.com. Software distribution and manuals are available to download.

Introduction

- COMPASS Ingeniería y Sistemas http://www.compassis.com is a company specilised in the industrialisation of FEM (and related) software.
- COMPASS has selected GiD to develop new products fully integrated in a graphical environment.

Step 1. Definition of the products

- Tdyn.
  - User friendly multipurpose Computational Fluid Dynamics environment.
  - Wide capabilities to deal with complex geometries.
  - Easy to use solver, including pre and post processing capabilities, wizard tools, ...
  - Low cost product.
- RamSeries.
  - Design tool based on a structural analysis solver.
  - Wide capabilities to deal with complex geometries.
  - Easy to use solver, including pre and post processing capabilities, verification tools, ...
  - Low cost product.

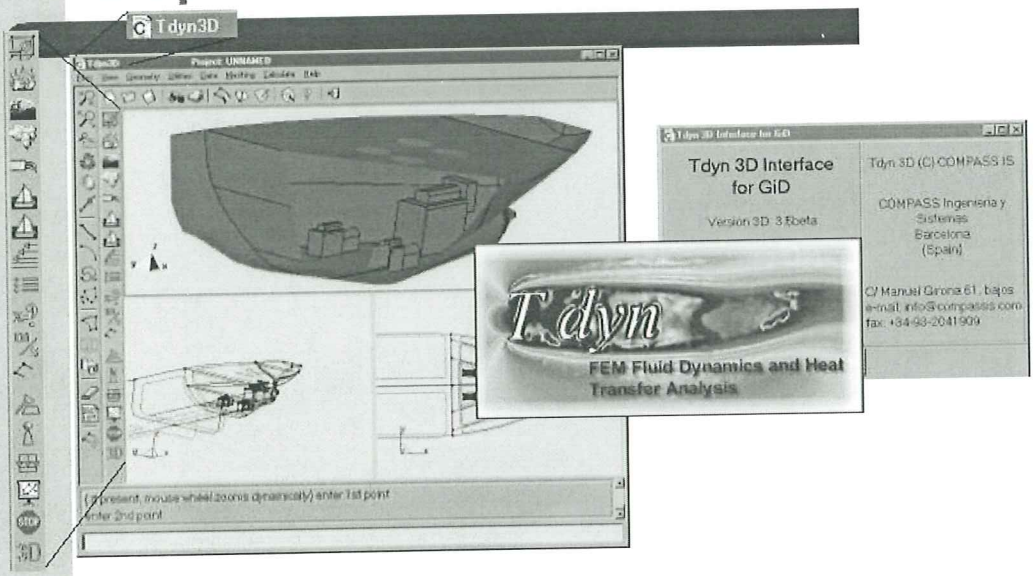
Step 2. Basic adaptation of GiD

- Standard step in the customisation process consist on creating the configuration, template and execution files:
  - problem_type_name.cnd Conditions definitions
  - problem_type_name.mat Materials properties
  - problem_type_name.prb Problem and intervals data
  - problem_type_name.uni Units Systems
  - problem_type_name.sim Conditions symbols
  - ***.geo Symbols geometrical definitions
  - problem_type_name.bas Information for the data input file.
  - problem_type_name.bat Operating system shell that executes the analysis process

Step 3. Basic TCL extension

- At this point the user may comfortably use the solver from within GiD GUI, but the user is still working in GiD, the new product GiD+ does not exists!, we need at least
  - Product icons
  - New specific toolbars (TCL extension)
  - Splash windows (TCL extension)
  - About window (TCL extension)
  - Remove GiD menu options not used (TCL extension)
  - GiD+ Installation files

Step 3. Basic TCL extension



Step 4. Final integration

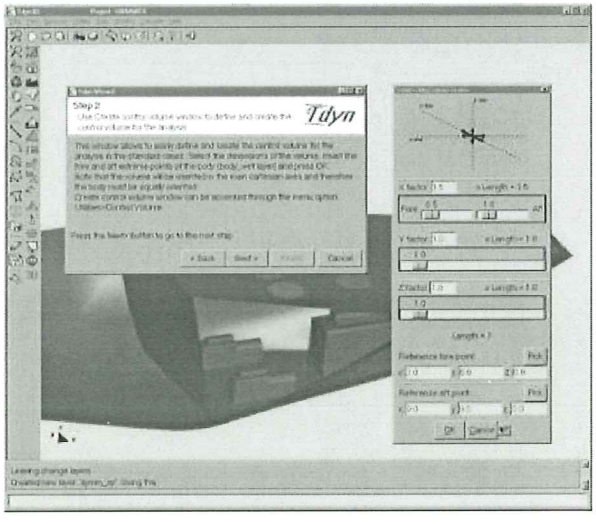
- Some more details have to be considered in order to obtain a profesional software
  - Error handling
  - Software manuals on line
  - Help on line
  - Tutorials on line

Step 5. Advanced adaptation

- Last step is the development of new tools and wizards.
- The objective is getting advantage of every possibility of the integrated system.
- This process requires the re-definition of the GiD menus.

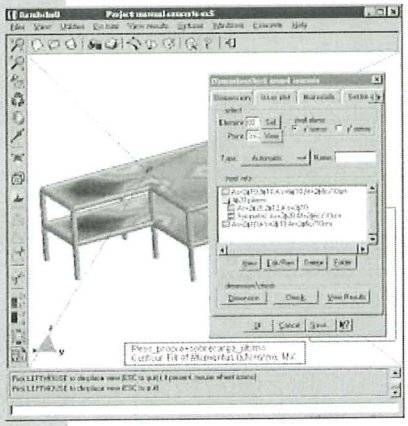
Step 5. Advanced adaptation

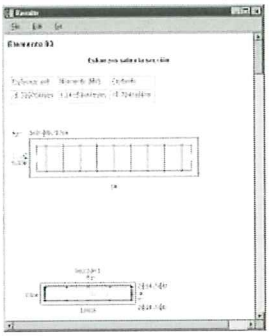
- Wizards

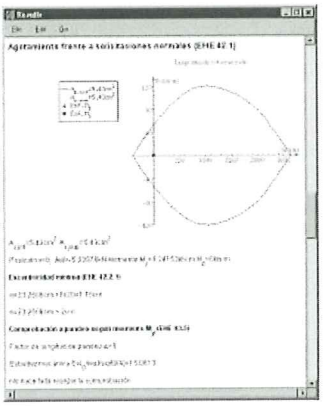


Step 5. Advanced adaptation

- Design tools, based on regulations







Conclusions

- Compass has succesfully created new products (Tdyn and RamS) based on GiD
- Tdyn and RamS accomplish the initial requirements (capabilities, easy to use, low cost, ...)
- Compass has detected the need of the market for this kind of services and developed the idea of Custom GiD (i.e. Compass develops new products based on GiD for third parts)

CODE_BRIGHT/GID: A 3-D PROGRAM FOR THERMO-HYDRO-MECHANICAL ANALYSIS IN GEOLOGICAL MEDIA

Jean Vaunat[1] and Sebastià Olivella[1]

[1]Department of Geotechnical Engineering and Geosciences,
Technical University of Catalonia (UPC), Módulo D2, Campus Nord, c/ Jordi Girona, 1-3
08034 Barcelona, Spain

SUMMARY: Code_Bright/GiD, a 3-D program for the thermo-hydro-mechanical analysis in geological media is described. Code_Bright is a Finite Element code which solves any partial set of 5 equations: stress equilibrium, water mass balance, air mass balance, energy balance and balance of conservative solute. The high number of variables and the strong coupling between them requires special interface features. GiD has been used to build such an interface. Theoretical bases and numerical strategies of the finite element model and elements of GiD interface are presented. The example of the 3-D thermo-mechanical analysis of an *in situ* test simulating a radioactive repository in creeping salt rock is finally described.

CODE PRESENTATION

Theoretical basis

CODE_BRIGHT is a Finite Element program designed to handle thermo-hydro-mechanical (THM) coupled problems in geological media. It solves in the most general case an Initial Boundary Value Problem consisting in a set of five governing equations (stress equilibrium, water mass balance, air mass balance, energy balance and balance of conservative solute), but the possibility exists to solve a partial set of equations. Model can account for large displacements, dissolution process of air into water, advective and non-advective flux of species and energy and water phase changes. Such complex problems require the consideration of a significant number of constitutive laws defining both uncoupled and coupled behaviour for all the variables.

Numerical solution

Special numerical techniques are used to handle this highly non-linear problem. Distinct treatments of storage, advective and diffusive terms are implemented in the discrete model. A mass conservative scheme is considered. Full Newton-Raphson is used to cope with the non-linearities. Direct and iterative solvers with sparse matrix storage can be alternatively selected by the user. Parallel version of the program is available.

Pre-and post-processing

To handle in a user-friendly way the large amount of information required to define the problem, CODE_BRIGHT uses GiD system for pre- and post-processing. CODE_BRIGHT interface follows GiD standards and considered 4 problem-wise windows: the general problem data window, the material parameter window, the initial and boundary problem window and the interval windows. To simplify the input data task, field dependencies are set within the windows. An example of the sheet defining the set of equations to be solved within the Problem data window is shown in Figure 1. The check boxes selected in this sheet determine the number of parameters to be entered. In Figure 2, an example of the Material

window is depicted. Constitutive relationships are divided into mechanical, hydraulic and thermal and physical laws. Due to the strong coupling between variables, up to 27 laws may be selected, some of then defined by 10 parameters. The code finally allows considering changes in material parameters and boundary conditions between time steps, which required a special implementation within GiD.

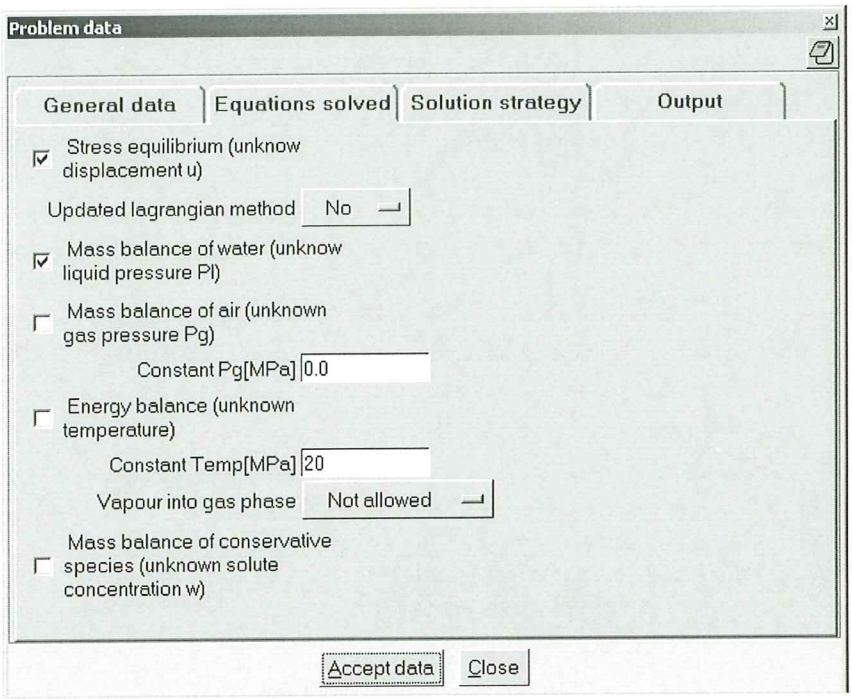


Fig. 1: Windows defining the system of equations to use.

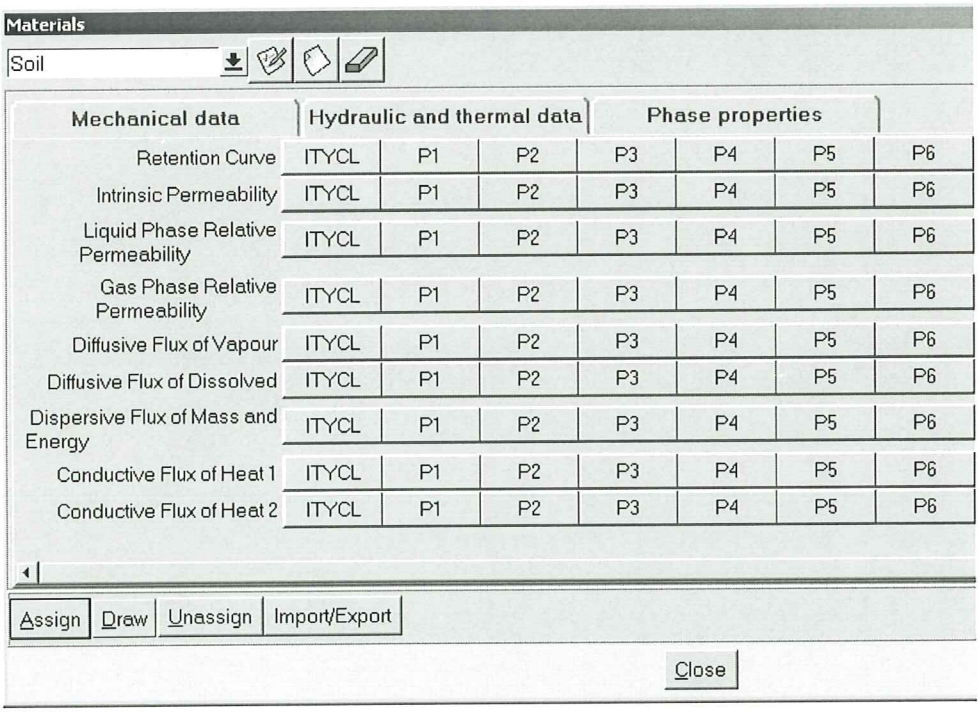


Fig. 2: Window defining the hydraulic parameters of the materials.

EXAMPLE OF APPLICATION

One of the applications performed with Code_Bright/GiD consisted in the three-dimensional thermo-mechanical simulation of an *in situ* test currently in progress in the Asse salt mine – Germany (Droste *et al.* 1996, see Fig. 3). The TSS (Thermal Simulation of Drift Emplacement) test aims at simulating the behaviour of a repository in rock salt. It consists of two parallel tunnels (length=70 m) separated 14.8 m and backfilled with crushed salt. Monitoring instrumentation was installed in the backfill and the host formation. Three heaters (length of 5.5 m, diameter of 1.5 m and thermal power of 6.4 kW) were emplaced in each tunnel to represent the thermal load of the radioactive waste canisters.

The finite element mesh is shown in Fig. 4a. Field measurements indicated that the problem can be considered symmetric with respect to the plane between drift A and drift B. Consequently, the domain modelled includes the cross section of only one tunnel (with three heaters) and part of the confining host rock. Initial and boundary conditions have been obtained from published reports by GRS (Droste et al 1996).

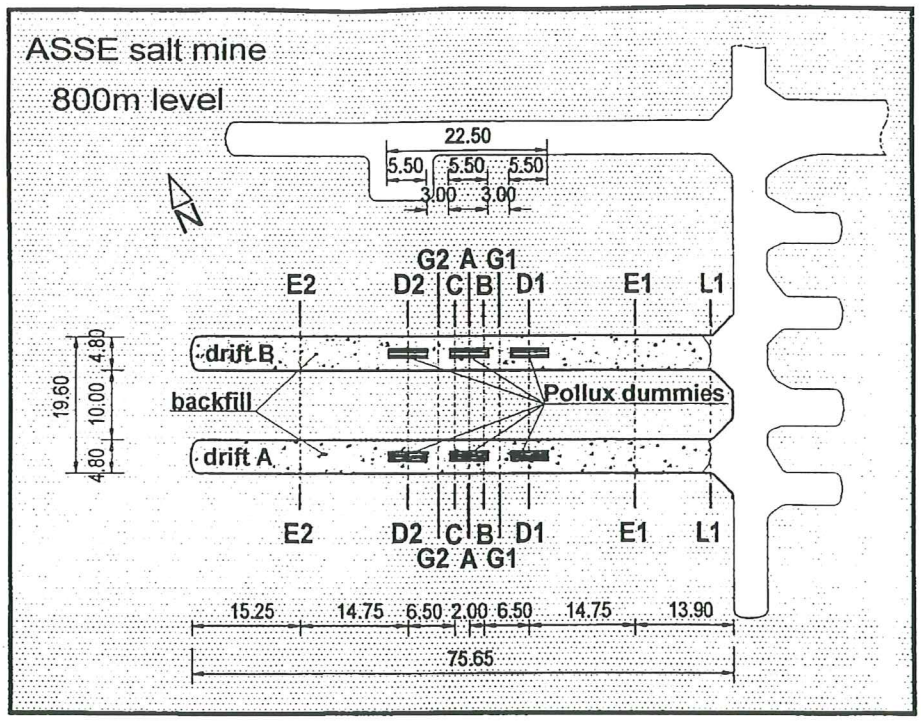
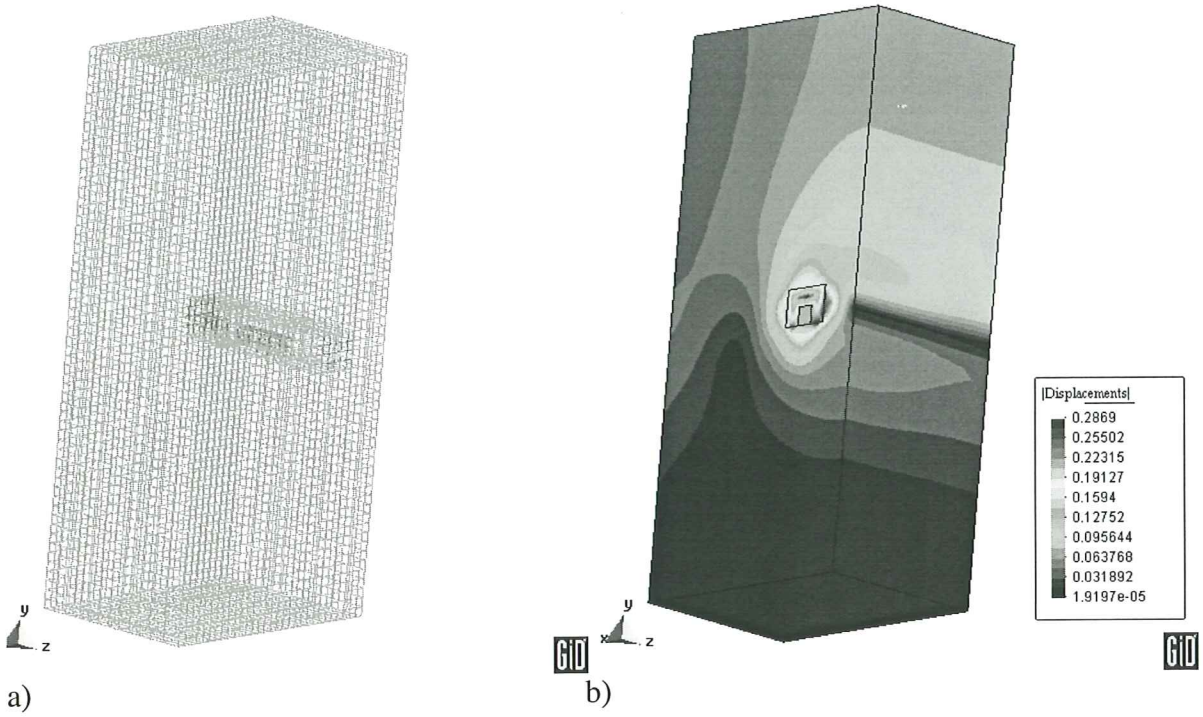


Figure 3: General view of the TSS test field on the 800 m level.

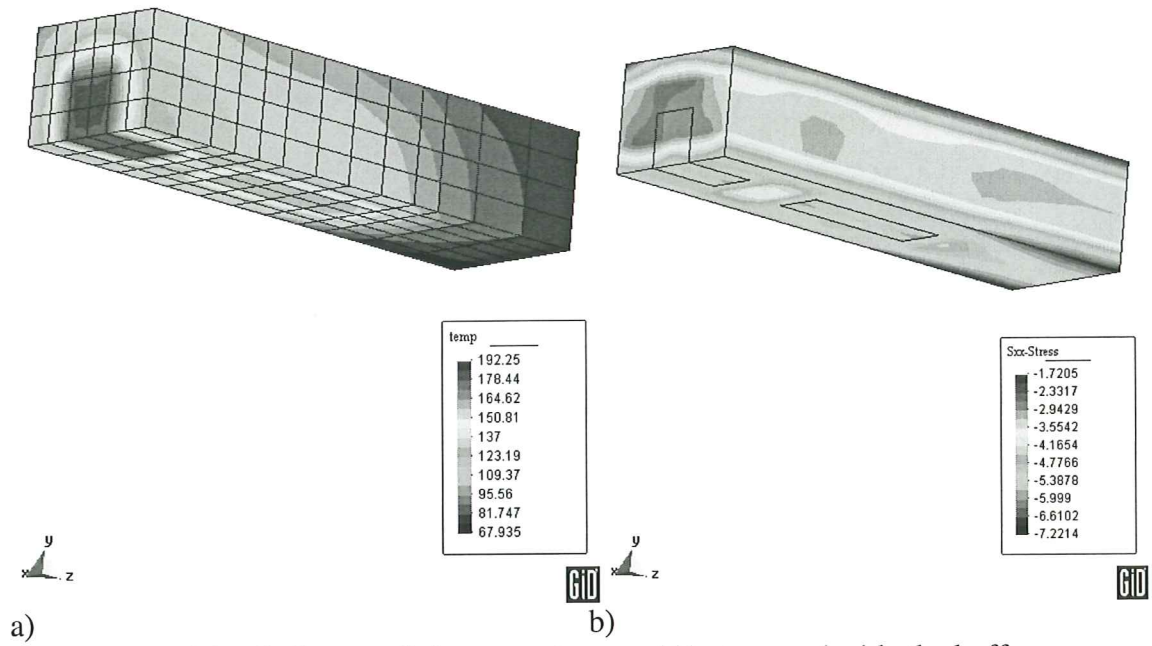
The simulation is achieved by solving stress equilibrium and heat transfer equations. A porosity dependent thermoviscoplastic behaviour, where the strains depend on the stresses through a power law, is considered for the salt (Olivella 1995). Strong couplings exist due to the dependency of the heat capacity on porosity, the dependency of the viscous coefficients on temperature and the dependency of the thermal conductivity on temperature and porosity. The model considers moreover two deformation mechanisms: the dislocation creep (DC) due to solid (salt) grains deformation and the fluid assisted diffusional transfer (FADT) due to dissolution/diffusion/precipitation of salt at the pore scale. Finally, stresses changes as a consequence of rock and backfill thermal expansion.

Outputs of displacements, temperature and stresses are shown in Fig. 4b, 5a and 5b. The results shown that the behaviour of an *in situ* test simulating a repository in a salt mine can be properly modelled.



a) b)

Fig. 4: a) Mesh and b) output of displacements



a) b)

Fig. 5: Output of a) temperature, and b) stresses, inside the buffer.

REFERENCES

1. Droste, J. H.K. Feddersen, T. Rothfuchs and U. Zimmer, 1996, The TSS Project: Thermal Simulation of Drift Emplacement.(GRS).

2. Olivella, S., J. Carrera, A. Gens, E. E. Alonso,1994a. Non-isothermal Multiphase Flow of Brine and Gas through Saline media. Transport in Porous Media, 15, 271:293

3. Olivella, S., A. Gens, J. Carrera, E. E. Alonso, 1996, 'Numerical Formulation for a Simulator (CODE_BRIGHT) for the Coupled Analysis of Saline Media " Engineering Computations, Vol 13, No 7, pp: 87-112

TEXTURES IN SIMULATION

Miguel A. de Riera Pasenau [1]

[1] *International Centre of Numerical Methods in Engineering*
Mod. C1 Campus Norte UPC, c/ Gran Capitán s/n, 08034 Barcelona, Spain
http://gid.cimne.upc.es gid@cimne.upc.es

Summary: Textures, a well known and widespread used technique in games, can also be useful in simulation of manufacturing processes, not only as a tool to improve the appearance of the results, but also to validate the simulation code and reduce manufacturing costs. Textures, basically, are images glued to polygons. These textured images can be used to make the simulation's results understandable at a glance, to improve quality and manufacturing efficiency, for instance in the can industry, and to contrast the simulation with experimental results.

Keywords: textures, manufacturing, textured images, experimental results comparison, image mapping, image projection, postprocess, realism.

Introduction

Texturing is a technique for efficiently modelling the surface's properties. Textured images, or simply *textures*, provide the most realism in a model and can be used efficiently to hide the model's polygonal aspects. This mechanism plays a key role in games for years and is one of the first features implemented in real-time rendering systems. From the several texturing methods: colour image texturing, which glues simple colour images onto surfaces, alpha blending in texturing, to specify holes and transparencies over the surface, reflections via environment mapping, to generate approximations of reflections in curved surfaces, and rough surface simulation using bump mapping, which makes a surface appear uneven in some manner (bumpy, wrinkle, wavy, etc.), among others, this paper will be focused in the simple colour image texturing.

Textures 1: a way to simplify the model and provide more realism

So far, in post-process, every polygon has been drawn as either a solid colour, smoothly shaded between the colours at its vertices or using colour scale when doing contour fill. In a human behaviour simulation[1], where each person is simulated as a simple point, when viewing the 2D simulation, only moving points are seen. This results representation makes it difficult for a non-initiate user to distinguish if the simulation handles with persons or sand grains through a funnel, a bee swarm or oranges storage.

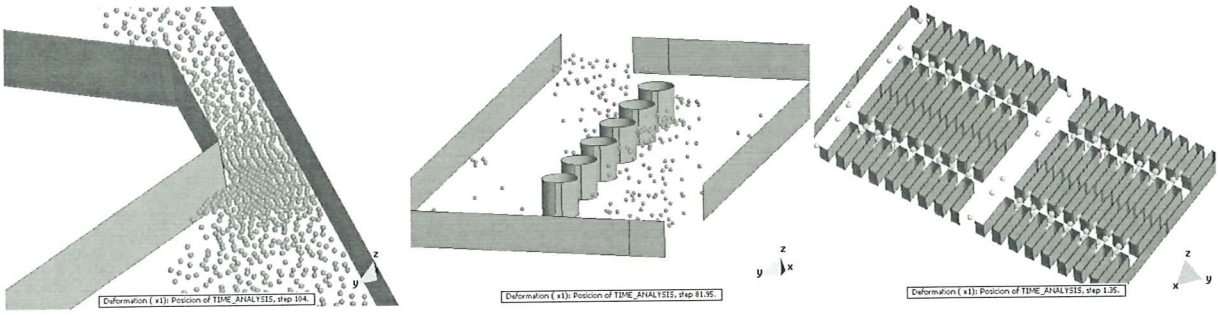


Fig. 1: 2D point simulation: from left to right, persons flow through a narrow pass, persons path while visiting a exposition and evacuation simulation of a airplane (courtesy of Reinald Lohner).

One way to improve the representation will be to drawn a person object for each point of the simulation with lots of polygons: a coarse representation with a sphere as head, and five cylinders as body, legs and arms, needs 20 quadrilaterals, 4 for each cylinder, and 18 triangles, 2 for each cylinder and 8 for the sphere, that is, a total of 58 triangles. In addition to this, if the analysis simulates thousands of points/persons, this will mean to handle hundred thousands of triangles, and even then all the persons will look the same and frozen when the analysis, animation, extends along time.
Another way will be to draw a quadrilateral for each points and glue an image of a person (perhaps a real one). But not only an image, but a sequence of images so that the appearance of each person changes when the animation of the simulation advances.

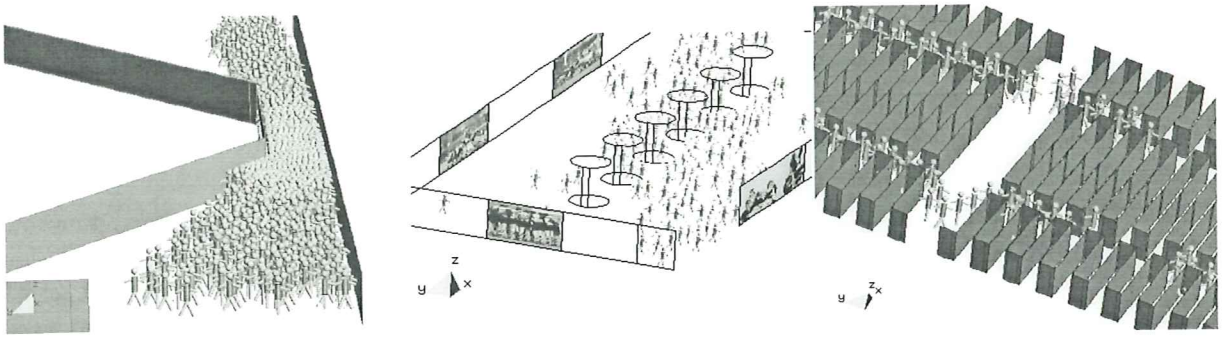


Fig. 2: 2D point simulation with images attached, the image is a snapshot of a person represented as a sphere and five cylinders: from left to right, persons flow through a narrow pass, persons path while visiting a exposition and evacuation simulation of a plane (courtesy of Reinald Lohner).

Understanding texture mapping

Textures are simply rectangular arrays of data, colour data in our case, but it can also be luminance data for light mapping, transparency values for alpha mapping, normal values for bump mapping, among others. Over this image-space, where coordinates $(a, b) \in ([0, image\_width], [0, image\_height])$ in pixels, a parameter-space is defined, called *texture-space*, where coordinates $(s, t) \in [0, 1]^2$. These texture-space values are used to find what the colour of the image is at this location.
To glue an image to a polygon, a *projector* function is needed so that for each point (x, y, z) of the polygon texture coordinates (s, t) are obtained. This process is called *mapping*. To draw the image region over a triangle using the facilities of the rendering system, only the texture coordinate (s, t) over each vertex has to be assigned. The texture coordinates at the vertices are interpolated by the rendering system to obtain texture coordinates at other pixels

in the triangle. Each interpolated coordinate is also used to do a colour lookup in the image. The texture coordinates *(s, t)* together with colour image values is called a *texture element*, or *texel* for short.

What makes texture mapping tricky is that a rectangular texture can be mapped to nonrectangular regions, and this must be done in a reasonable way. [2, 3, 4]

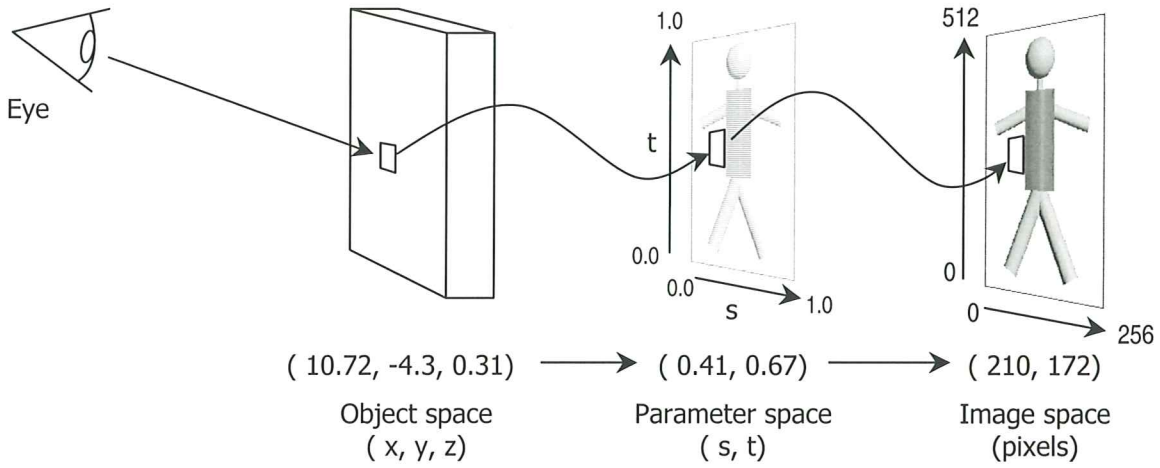


Fig. 3: texture mapping process: a world point is mapped to the parameter space to get the texture coordinates *(s, t)*, and with these coordinates the colour pixel is located on the textured image.

One of the advantages of this process is that, once the texture coordinates *(s, t)* are *mapped* to the vertices of a polygon, the same image region is rendered. If this polygon deforms or distorts in some way, the image region defined by its texture coordinates *(s, t)* will be deformed, distorted, in the same way.

Textures 2: improving quality and manufacturing efficiency

Using this advantage, another field of application of texture mapping is opened: the pre-distorted printing. An image applied to a surface involved in a, for instance, stamping analysis process will get the form of these surface at the end of the simulation. And the other way works too. An image applied to an already stamped surface will be deformed when this surface get its original form back.

Fig. 4: Can stamping simulation made by Quanteck ATZ for Metalpack SA: after deforming the sheet mesh to the final can state, the decorative labels are applied to the mesh and then its original flat shape is restored to get the deformed images, ready for the press (courtesy of Quantech ATZ, Metalpack SA).

This application of textures is being used by the Stampgen program, at Quanteck ATZ[5, 6]. The tool-making module employed for can-making has been enhanced by simulating the distortion of flat images printed on sheet into the finished product. So time can be saved by just applying the decorated label on the finished can at the end of the simulation process and then obtain the initial decoration which will be printed to the flat sheet.
Two limitations have to be overcome to match the quality and needs that this printing process needs. These limitations are inherent to the texture mapping process and the rendering system: one is the size constraints on textured images that the rendering system imposes, and the other is the limited texture coordinate interpolation scheme used by the rendering system, which is shown when using a too coarse mesh for the definition of the surface.

Texture limitations

One texture limitation is that the size of images used for texture mapping must be power of 2 and cannot exceed the size limit imposed by the rendering system. On some systems this limitation is as small as 256x256 pixels, but generally a texture image can be as big as 1024x1024 or 2048x2048. This is not enough for the press, where they demand images with a resolution of, at least, 1200dpi. For example, a can with a diameter of 70mm and a height of 30mm the image size of the decorative label has 10390x1417 pixels.
To map these huge images to the finished can and get its original flat sheet printing, these images have to be subdivided into several textures and map each of them over the corresponding surface region. Following the previous example, the 10390x1417 image must be subdivided into 41x6 = 246 textures of 256x256 pixels, or 11x2 = 22 textures of 1024x1024 pixels.

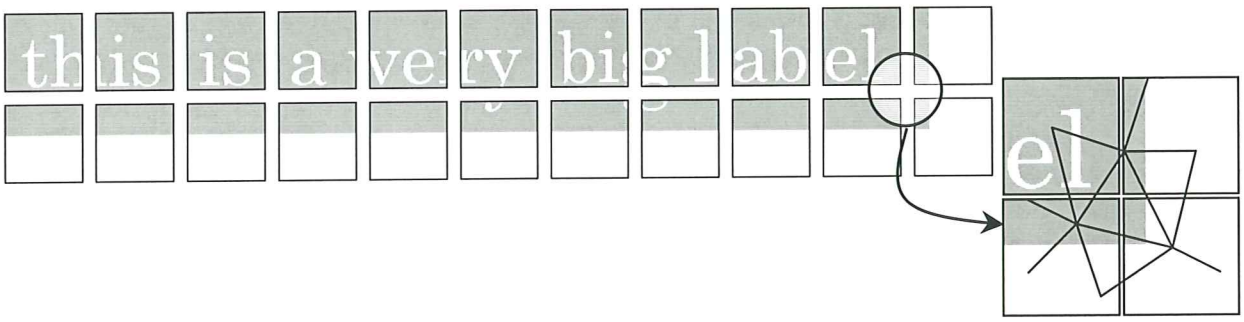


Fig. 5: example of a label subdivision, the grey shaded label is subdivided into 11x2 textures of 1024x1024 pixels. The zoom shows a triangle which shares 4 textures at once.

With this subdivision another difficulty appears, some triangles of the surface mesh will lie across two, three or even four textures and have to be divided accordingly, so that each piece of each triangle is rendered with its corresponding texture.
It is also worth to note that the context exchange when loading different textures in a rendering system is a big penalty which has to be minimized. When drawing the mesh, instead of traversing the elements and for each one of them load its texture and draw it, it is better to load the texture and then traverse the triangles and render the ones that uses this texture.
Another limitation is that there are cases where the interpolation of the texture coordinates along the triangles does not match the desired result. If a texture is applied to a warped quadrilateral, neither triangulation will do the correct interpolation as if the texture where

applied directly to the quadrilateral. What is needed is either another interpolation scheme, or a previously deformed image or a fine meshing. [2, 3, 4]

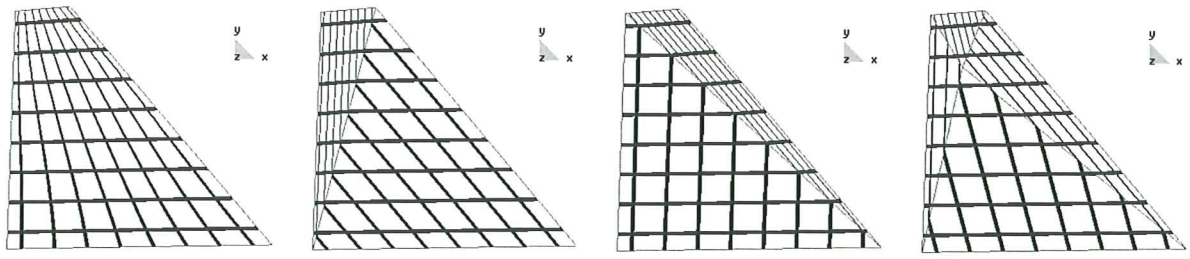


Fig. 6: applying a grid image to a quadrilateral surface: at the left, the desired result, on the right three images showing the interpolation problem with three different triangulations.

Textures 3: comparison between real and virtual results

In addition to the decorative use of textures, and to the industrial applications, using textures in simulation can also be used to contrast experimental results and the ones from the analysis. This can be accomplished by applying a precise pattern to a physical sheet in laboratory and applying the same pattern, as a textured image, in the computer sheet. After running the analysis and carry out the experiment in laboratory, the final deformed sheet can be compared measuring the pattern deviation from the original state and the deformed one, and between real and virtual analysis.

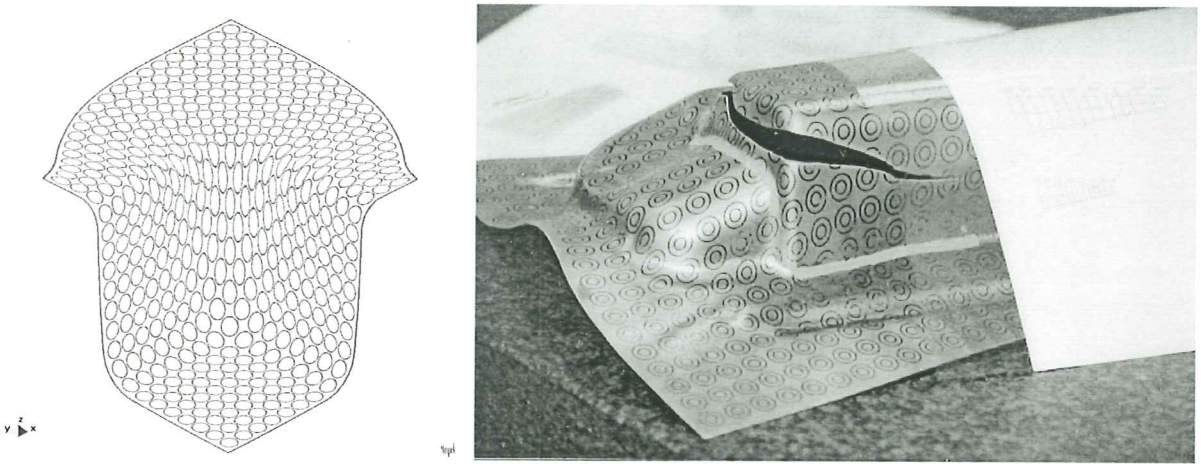


Fig. 5: Circles image applied to a quarter butterbox in simulation (courtesy of Quanteck ATZ) and using circles with the Mylar rule at the laboratory (courtesy of ASCAMM).

References

1. Reinald Lohner, Jul 2000.
2. Mason Woo, Jackie Neider and Tom Davis, *OpenGL Programming Guide, Second Edition*, Addison-Wesley Developers Press, 1997.
3. Tomas Möller and Eric Haines, *Real-Time Rendering*, A.K. Peters, Ltd., 1999.
4. David H. Eberly, *3D Game Engine Design*, Morgan Kaufmann Publishers, 2001.
5. "Distortion Made Clear", *The Canmaker*, May 2001, p. 35.
6. Quanteck ATZ, http://www.quantech.es

NEW ENVIRONMENT FOR THE SIMULATION, OPTIMISATION AND CONTROL OF FOOD PROCESSING

E. Balsa-Canto[1], C. González[1], J. Mora[1],
J. R. Banga[2], A. A. Alonso[2]

[1] *CIMNE International Centre for Numerical Methods in Engineering, UPC, Campus Nord, Gran Capitan s/n, C-1, 08034 Barcelona, Spain*
[2] *Process Engineering Group, IIM-CSIC, C/Eduardo Cabello 6, 36208, Vigo, Spain*

SUMMARY: Computer aided process engineering techniques have been successfully used for the simulation, optimisation and control of many industrial processes. However, in the case of the food industry, the complexity of the mathematical models of food processing, plus the often complex geometries of foodstuffs, makes the solution of this type of problems very complicated. The development of a new tool based on the combination of GID with suitable numerical methods is presented here as a new software package for the complete study of the most relevant food preservation techniques (sterilization, pasteurisation, and freezing, among others). This new tool will consist of a user friendly, ease to use environment allowing users from food companies to improve their processes in an effective way. As a consequence, they will be able to design better processes which will guarantee safety and quality of foodstuffs, with less cost, reduced energy consumption and minimum environmental impact.

KEYWORDS: Food preservation, simulation, optimisation, control, finite elements, GID

INTRODUCTION

One of the major concerns of the food industry is offering high quality and safe products to the consumers. Food safety and quality control ensures that the desirable characteristics of food are retained throughout the production, handling, processing, packaging, distribution and preparation stages. Food spoilage is a gradual process occurring because of enzymatic and/or chemical reactions, improper temperature control or microbial growth, resulting in undesirable changes in the colour, flavour, odour and/or texture.

It is well known that the use of adequate preservation techniques ensures a longer shelf life of the product. Food preservation through processing is an extremely broad area in food technology, including processes like refrigeration, freezing, pasteurisation, sterilization, fermentation, and drying, among others. Although the basic idea behind all these techniques is to either slow down or eliminate the activity of the bacteria causing spoilage, not all these processes are suitable for all types of food. Moreover, operating conditions must be carefully selected in order to guaranty a safe product with the maximum possible content of nutrients.

The increasing demand from industry of tools for the design of better processes and the computation of optimal operation policies has lead to extensive research in several areas related to food processing such as modelling, simulation, optimisation and control (e.g. see Teixeira et al, 1969; Banga et al, 1991; Fryer, 1994; Alonso et al., 1998, among others). The use of these techniques allows the reduction of cost and time if compared with traditional approaches usually based on extensive experimental work and "trial-and-error" procedures. However, regardless of all the advances in this field in the academic world, the use of these methodologies in industry is still very limited, with the absence of user friendly software environments being one of the main reasons. Moreover, although there is a number of commercial tools that allow quite easy simulation of many relevant food processes, this is certainly not the case for the solution of dynamic optimisation and control problems.

The objective of this work is to present a new, easy to use software tool (CALISO) which combines a library of numerical routines for the simulation, optimisation and control of a number of relevant food preservation processes, with the pre/post-processing capabilities of GID and a user-friendly graphical interface.

MODELLING AND SIMULATION OF FOOD PRESERVATION PROCESSES

Modern process systems engineering methods rely on mathematical models based on sound first principles, considering heat, mass and momentum transport phenomena, plus the corresponding expressions for the kinetics and thermo-physical properties. Moreover most of the relevant variables (e.g. temperature and moisture content) depend on both the position inside the food load and time. Therefore, the resulting mathematical models consist of usually highly non-linear sets of partial differential and algebraic equations (PDAEs) whose solution usually relies in numerical techniques. Most of these numerical approaches are based on the discretisation of the spatial domain, such as the finite differences approach, the numerical method of lines (NMOL, Schiesser, 1991) or the finite elements method (Zienkiewicz and Taylor, 2000; or Datta, 1998 for applications related in food processing).

DYNAMIC OPTIMIZATION OF FOOD PRESERVATION PROCESSES

Dynamic optimisation involves the calculation of time-varying control profiles (e.g. the heating temperature profile in a sterilization process) that optimise (minimise or maximise) a desired objective functional (usually related to cost, quality of final product, energy consumption, etc.) subject to the system dynamics and a number of constraints (e.g. microbiological safety). During recent years, dynamic optimisation methods have been successfully applied to a number of relevant processes from the food industry, such as thermal sterilization (Banga et al, 1991; Silva et al 1993; Durance, 1997) or drying (Banga y Singh, 1994).

CONTROL OF FOOD PRESERVATION PROCESSES

In order to implement these optimal operating policies obtained through the solution of the dynamic optimisation problems, it is necessary to use adequate control methodologies, usually based in the model predictive control scheme. This type of controllers has been applied to several food processes such as extrusion (Nikolau, 1996), drying and refrigeration (Trelea et al, 1998) and thermal sterilization (Alonso et al, 1998; Chalabi et al, 1999).

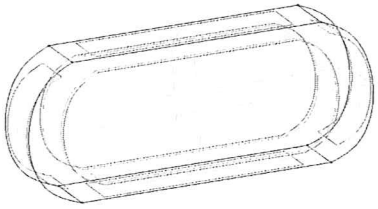
CALISO CAPABILITIES

CALISO will remedy some of the current software limitations providing tools non only for simulation but also for both optimisation and control. It will consist of three main elements:

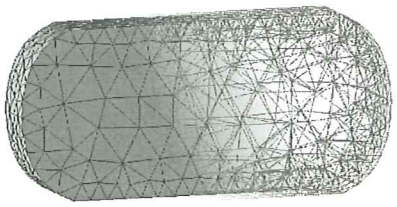
- A user friendly interface: in order to make work very simple, it will provide information regarding the modelling of different preservation processes, geometries, plus simulation, optimisation and control capabilities.
- Pre-postprocessor: GID will provide different mesh generation options, adaptability to simulation codes and visualization facilities, such as temperature distributions, animated sequences for dynamic analysis, graphics related to quality of products, etc.
- New modules with suitable and efficient numerical techniques will ensure the rapid solution of the simulation, optimisation and control problems.

The capabilities of CALISO may be shown through an example, the case of sterilization of canned foods. CALISO will allow the implementation of the can geometry, the simulation and the dynamic optimisation in order to maximize the nutrient contents inside the food while guarantying a level of sterility, that is safety. Figure 1 illustrates the process:

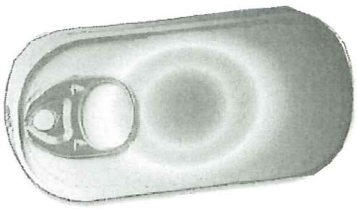
a) Can to be sterilized



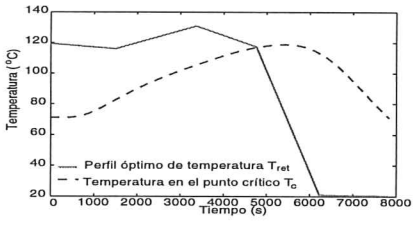
b) Geometry representation



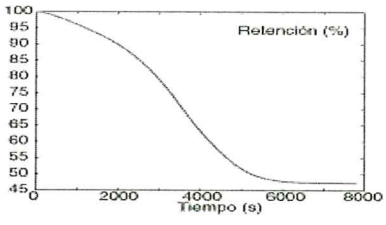
c) GID: Mesh generation



d) Temperature distribution at final time



e) Optimal operating conditions and temperature in the coldest point



f) Volumetric nutrient retention

Figure 1: CALISO solving approach

Moreover, for some very complicated cases the computational effort may be too large for the solution of dynamic optimisation and control problems. However, the use of reduced order models allows the efficient solution of these type of problems (Balsa-Canto et al., 2001 I,II). Therefore, CALISO will also include an appropriate module for the automatic generation of reduced order models.

CONCLUSIONS

Although there are a number of simulation tools based on the finite element method which could be somehow programmed to simulate food processing operations, this is not the case for dynamic optimisation and control. CALISO is a new, user friendly, easy to use environment which will allow the systematic application of computer-aided process engineering techniques to food preservation processes. This new tool will allow the modelling and simulation of highly complex food preservation processes and geometries, plus the possibility of computing their optimal operating conditions and designing closed loop controllers. This software environment will allow the food industry to improve both the performance of their processes and the quality of their products.

REFERENCES

1. Alonso, A. A., Banga, J.R. and Martín, R.I.P. "Modeling and adaptive control of a batch sterilization process", Comput. Chem. Eng. 22(3):445-458, 1998.
2. Balsa-Canto, E., Alonso, A.A. and Banga, J.R. "A novel, efficient and reliable method for the thermal process design and optimisation. Part I:Theory", J. Food Eng. , in press.
3. Balsa-Canto, E., Banga, J.R and Alonso, A.A. "A novel, efficient and reliable method for the thermal process design and optimisation. Part II: applications", J. Food Eng., in press.
4. Banga, J. R.; Perez-Martin, R. I.; Gallardo, J. M.; Casares, J. J. "Optimization of the thermal processing of conduction-heated canned foods: study of several objective functions", J. Food Eng. 14(1): 25-51, 1991.
5. Banga, J. R. and Singh R. P. "Optimization of the air drying of foods", J. Food Eng., 23:189-211, 1994.
6. Chalabi, Z.S, L.G. van Willigenburg and G. van Straten. "Robust optimal receding horizon control of the thermal sterilization of canned foods", J. Food Eng, 40(3):207-218, 1999.
7. Datta AK. "Computer-aided engineering in food process and product design". Food Technol-Chicago 52 (10): 44-52, 1998.
8. Domínguez, N. and Soler P. , "GID User Manual", Ed. CIMNE, Spain,1999.
9. Ribó, R., de Riera, M. and Solano, E., "GID Reference Manual", Ed. CIMNE, Spain, 1999.
10. Durance, T.D., "Improving canned food quality with variable retort temperature processes", Trends In Food Sci. & Tech. 8(4):113-118, 1997.
11. Fryer P, "Mathematical-Models In Food-Processing", Chem Ind-London (13): 515-518, 1994.
12. Nikolaou, M., "Computer-aided process engineering in the snack food industry", Chemical Process Control (CPC-V) Conference, January 7-12, Tahoe City, CA, 1996.
13. Schiesser, W. E. " The numerical method of lines". Academic Pres, New York, 1991.
14. Silva, C. L. M., Oliveira, F. A. R. and Hendrickx, M. "Modelling optimum processing conditions for the sterilization of prepackaged foods", Food Control 4(2): 67-78, 1993.
15. Teixeira, A.A., Dixon, J.R., Zahradnik, J.W. and Zinsmeinster, G.E. "Computer optimisation of nutrent retention inthermal processing of conduction heated foods". Food Technol., 23:137-142, 1969.
16. Trelea, I.C., Alvarez, G. and Trystram, G., "Non linear predictive optimal control of a batch refrigeration process", J. Food Process Eng. 21:1-32, 1998.
17. Zienkiewicz, O.C. & R. L. Taylor (2000), "The Finite Element Method", 5ª Ed. Butterworth-Heinemann, 2000.

NUMERICAL SIMULATION OF COMPLEX LARGE DEFORMATION PROCESSES

B.M. Chaparro[1], J.M. Antunes[1], L.F. Menezes[2], S. Thuiller[3], P.Y. Manach[3]

[1]*Instituto Politécnico de Castelo Branco, Escola Superior de Tecnologia, Avenida do Empresário, P-6000 Castelo Branco – Portugal*
[2]*Universidade de Coimbra, CEMUC, Departamento de Engenharia Mecânica – Universidade de Coimbra, Pinhal de Marrocos, P-3030 Coimbra – Portugal*
[3]*Laboratoire Génie Mécanique et Matériaux, Université de Bretagne Sud, Rue de Saint Maudé, 56325 Lorient – France*

SUMMARY: In recent years, the development of new material models has allowed to predict the mechanical behaviour of metals in complex strain paths. In this paper, a comparative analysis of numerical results, obtained with different constitutive models, is presented. Those models were implemented in the finite element code DD3IMP, devoted to the numerical simulation of the deep-drawing process. We focus the ability of the GID application to visualize and compare the results of different mechanical models, particularly the evolution of the stress and strain fields, contact forces, displacements, etc., calculated during the numerical simulation of the deformation process.

KEYWORDS: Large Plastic Deformation, Numerical Simulation, Deep-drawing.

INTRODUCTION

The accuracy of the numerical simulation results depends on various factors being one of them the constitutive models adopted. This is particularly relevant in sequential deformation paths. Nowadays, the industrial numerical simulations in metal forming use simple phenomenological models. In metal forming, the strain paths imposed are usually complex and, in order to describe the material behaviour with increasing accuracy, one needs to use new complex mechanical models based on microstructural evolution during plastic deformation. In this work, we compare the simulation results obtained with the recently proposed Teodosiu & Hu physical model [1] with others widely used phenomenological models. The classical Numisheet'93 square cup drawing is simulated.

THE FEM CODE DD3IMP

The DD3IMP finite element code (Deep Drawing 3 Dimensional Implicit) is devoted to the simulation of the deep drawing processes [2]. The used formulation considers large plastic deformations and rotations. This code uses the Hill's 1948 criteria to describe the plastic anisotropic behaviour. The elastic behaviour is considered as isotropic. The contact issue is solved using the classic Coulomb model. In the mechanical formulation an augmented Lagrange multiplier is applied to associate the equilibrium equations with the contact problem. This leads us to a mixed non-linear problem where the final unknowns are the nodal

displacements and the contact forces. A Newton-Rapshon scheme is used to solve this problem in a single iterative loop algorithm. In this code four isotropic hardening constitutive models were implemented to describe the work-hardening behaviour: Isotropic hardening, described by the Swift law; isotropic plus kinematic hardening, where the Swift law is used to describe the isotropic hardening whereas a saturation Voce type equation describes the kinematic hardening; isotropic and kinematic hardening using in both situations a saturation Voce type equation and, finally, a distortional hardening described by the microstructural based Teodosiu & Hu model.

TOOLS AND SHEET DESCRIPTION

To increase versatility and usefulness, the code use Bezier surfaces to describe the stamping tools. The present problem is the classical Numisheet'93 cup deep-drawing. The sheet is modeled with three-linear eight nodes isoparametric hexahedron associated to a selective reduced integration technique. Taking into account the geometry, it is only simulated a fourth part of the tools and the sheet (fig. 1 and 2).

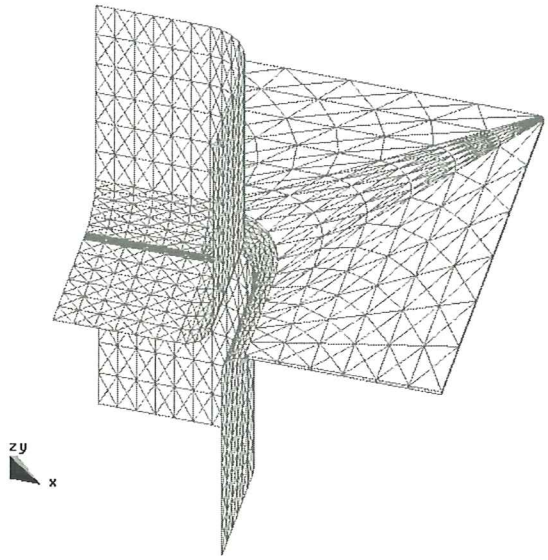


Figure 1: Die, blank-holder and punch geometry's.

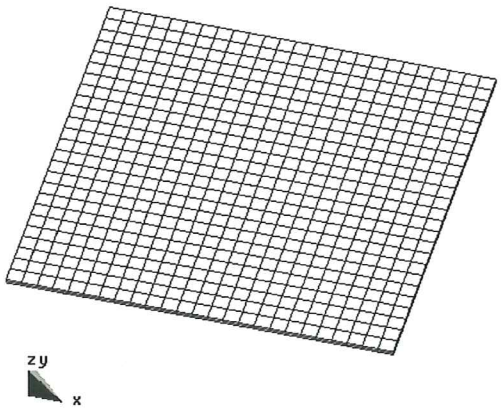


Figure 2: Sheet discretization.

The deep-drawing cup is discretized with 1250 elements: 25x25 elements in the sheet plane and 2 elements in thickness. Sixth order Bezier surfaces were used to simulate the geometries of the die, punch and blank-holder. The boundary conditions were: the die is locked, a constant force is applied (4900 N) in the blank-holder tool and the punch stroke is 40 mm. In order to simulate only a quarter of the total problem symmetry displacement restrictions are imposed to the borders of the sheet.

NUMERICAL SIMULATIONS

In order to evaluate the work hardening process, the Teodosiu & Hu model is compared with three classic models. The first one only considers isotropic hardening, fitted by the Swift law; the second one also considers the isotropic work hardening with the Swift law but with kinematic work hardening with a saturation value, described by a Voce type law; the last one considers a saturation value for both work hardening components. The material properties correspond to a mild steel and were taken from the literature [3]. The Coulombs friction coefficient used in the simulations was 0.144.

RESULTS

To compare and visualize the evolution of the main variables of the deformation process the software GID was used. Figures 3 to 10 shows the distribution over the deformed sheet of some of them.

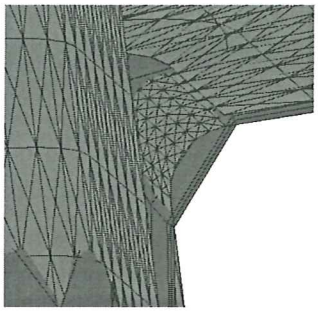


Figure 3: Detail of the tools-sheet contact.

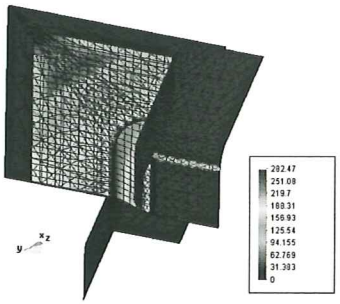


Figure 4: Global view at 5 mm punch displacement obtained with Swift model.

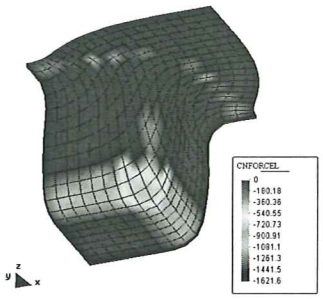


Figure 5: Normal contact forces distribution at 40 mm punch displacement obtained with Teodosiu & Hu model.

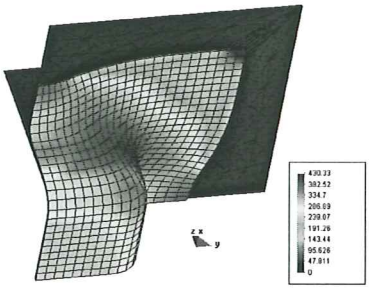


Figure 6: View of the deformation process without the die tool at 40 mm punch displacement for Teodosiu & Hu model.

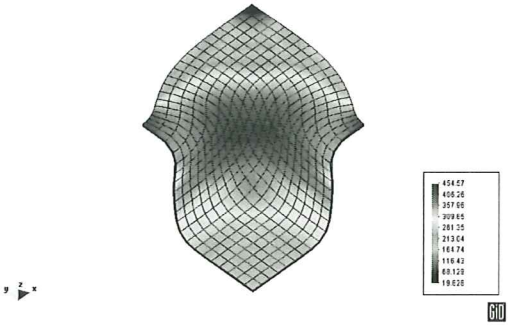


Figure 7: Equivalent stress distribution for the simple Swift model at 40 mm punch displacement.

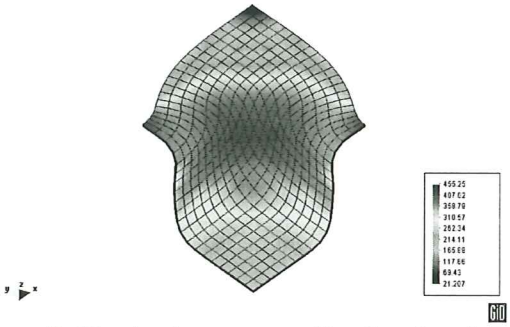


Figure 8: Equivalent stress distribution for the Swift law with back stress Kinematic Voce type law at 40 mm punch displacement.

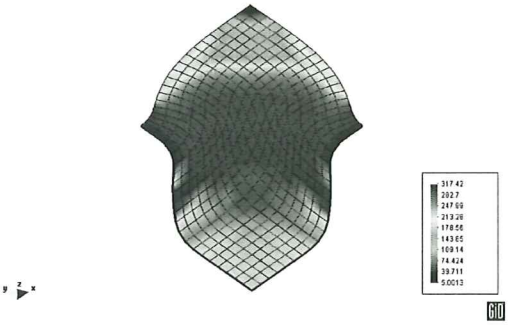


Figure 9: Equivalent stress distribution for the Voce-Voce type model at 40 mm punch displacement.

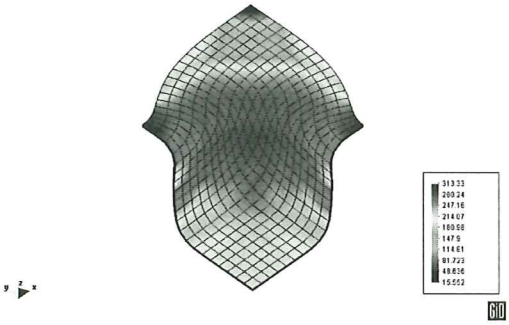


Figure 10: Equivalent stress distribution for the Teodosiu & Hu model at 40 mm punch displacement.

It is possible to observe in figure 3 some contact details that can help us to define the level of discretization needed for our problem. In figures 4 to 6, it is shown, respectively, an initial deformation state, the final normal contact forces distribution and, finally, a detail showing the equivalent stress state distribution on the sample sheet at the final of the deformation process. In figures 7 to 10, it is possible to compare the stress state distribution obtained at 40 mm punch stroke for the four constitutive models used.

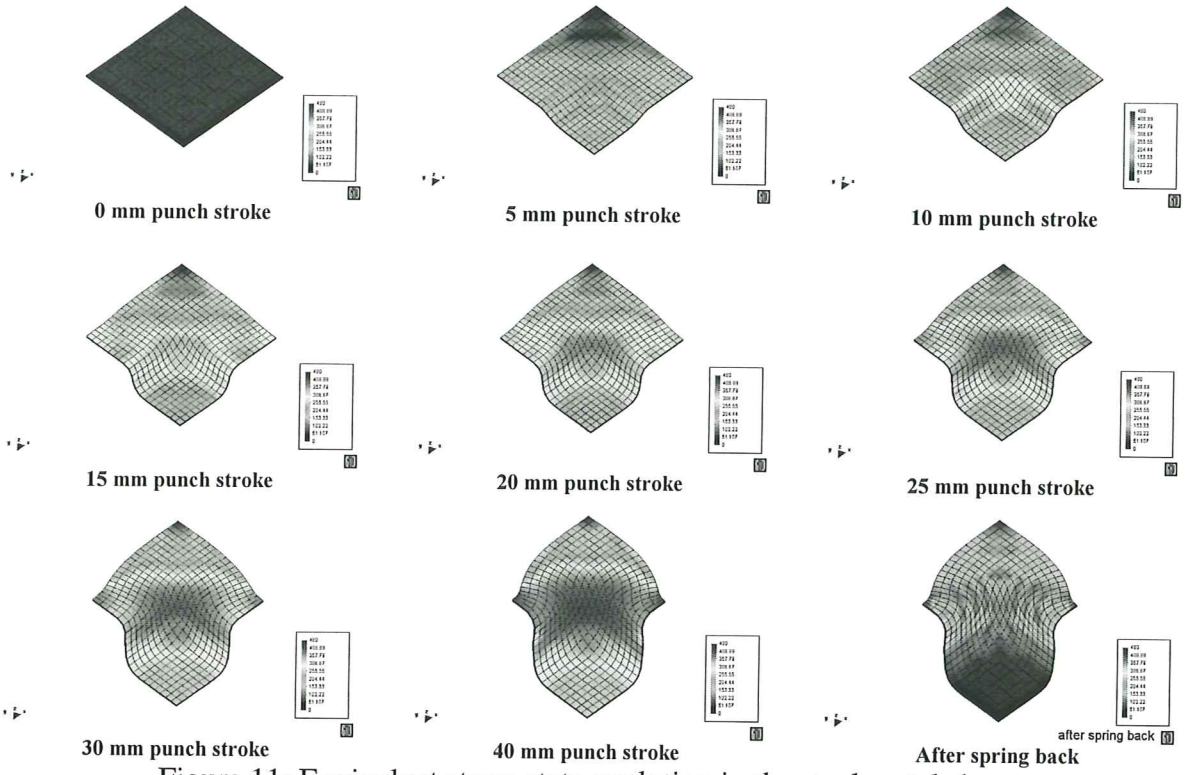


Figure 11: Equivalent stress state evolution in the steel metal sheet

In figure 11 it is presented the equivalent stress evolution during the deformation for the isotropic Swift hardening without kinematic hardening.

CONCLUSION

It is easy to observe global or detailed information about large deformation parameters evolution using GID: this includes the typical state parameters in nodes or on the integration points. In this paper, the results of the intuitive interaction between the pre and post processor GID with the DD3IMP solver developed at the CEMUC (Centro de Engenharia Mecânica da Universidade de Coimbra) are shown. The GID software was used to create the initial sheet geometry and to do all the post-process tasks. To conclude is possible to assure that this program brings us a real answer to many solver developers' problems.

REFERENCES

1. Teodosiu, C., Hu, Z., Evolution of the intragranular microstructure at moderate and large strains: Modeling and computational significance, Simulation of Materials Processing: Theory, Methods and Applications, 173, 1997.

2. L. F. Menezes, C. Teodosiu, A. Makinouchi in: FE Simulation of 3D Sheet Metal Forming Processes in Automotive Industry, Tangungsbericht Der VDI-Gesellschaft Fahrzengtechnik, Düsseldorf, 894, 1991.

3. Reis, A., Modelação Numérica por Elementos Finitos de Processos de Conformação Plástica, mSc thesys, Oporto University, 2001.

FINITE ELEMENT SIMULATION OF ULTRAMICROHARDNESS TESTS

J. M. Antunes[1], B. M. Chaparro[1], L. F. Menezes[2], J. V. Fernandes[2]

[1]*Instituto Politécnico de Castelo Branco, Escola Superior de Tecnologia, Avenida do Empresário, P-6000 Castelo Branco – Portugal*
[2]*Universidade de Coimbra, CEMUC, Departamento de Engenharia Mecânica – FCTUC, Pinhal de Marrocos, P-3030 Coimbra – Portugal*

SUMMARY: A numerical study of the ultramicrohardness test is presented. Mesh refinements were tested in order to evaluate the influence of the finite elemente mesh on the simulation results. The influence of the friction coefficient value between indenter and sample was also evaluated. For that purpose specific finite element simulation software was used, and different materials were simulated using Vickers indenters with two different offsets. Post process analysis was full performed with GID.

KEYWORDS: Numerical simulation, Hardness.

INTRODUCTION

The development of depth sensing indentation equipment has allowed easy and reliable determination of two of the most popular mechanical properties of materials: the hardness and the Young's modulus. However, some difficulties emerge in the experimental procedure to calculate the accurate values of the refereed properties. This is related, for example, with the geometrical imperfections of the tip of the diamond pyramidal indenter (such as the offset of the Vickers indenter), the definition of the contact area at the maximum load and the compliance evaluation. The numerical simulation of ultramicrohardness tests can be a helpful tool to better understand the influence of these parameters on the determination procedures for the calculation of the mechanical properties. In this way, different types of finite element mesh refinements were tested in order to evaluate the influence of the FE mesh on the hardness values. The influence of the friction coefficient value between indenter and sample was also evaluated.
To perform this study, specific simulation software, HAFILM, was developed. The simulations were performed with two different materials (AISI M2 steel and Nickel), using Vickers indenters with two different offsets. The software GID was used on the indentation results analyses.

DESCRIPTION OF THE FE CODE HAFILM

The mechanical model that is the base of the FE code HAFILM, considers the ultramicrohardness test as a process of large deformations and rotations. The plastic behaviour of the material is described by the anisotropic Hill's criterion with isotropic and kinematic hardening. Elastic behaviour is considered isotropic. It is assumed that contact with friction

exist between the sample and a rigid indenter. To model the contact problem a classical Coulomb law is used. The association of the static equilibrium with the contact with friction makes use of an augmented multiplier. This led to a mixed formulation, where the kinematics (material displacements) and static (contact forces) variables are the final unknowns of the problem. For it's resolution the HAFILM code uses a fully implicit algorithm of Newton-Raphson type. All non-linearities, induced by the elastoplastic behaviour of the material and by the contact with friction, are treated in a single iterative loop [1].

INDENTER AND SAMPLE DESCRIPTION

For the description of the sample, three-linear eight-node isoparametric hexahedron are used (see e.g. [2]), associated with a selective reduced integration technique [3]. In brief, the dilatational contribution to the stiffness matrix is considered constant over the element, and equal to its value at the central point of the element, whereas a full integration rule is used for the deviatoric part. The main characteristics of the three different meshes used in this work are presented in table 1. This table shows the number of elements and their size on the indentation region that compose the finite element meshes. Because of the symmetry along the X and Z-axis only a fourth of the sample is used in the simulation (fig. 1).

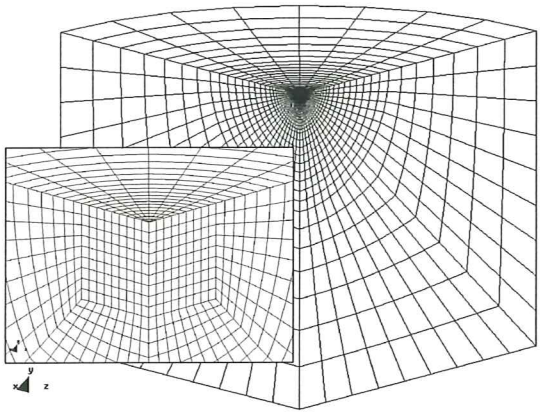


Figure 1: Finite element mesh

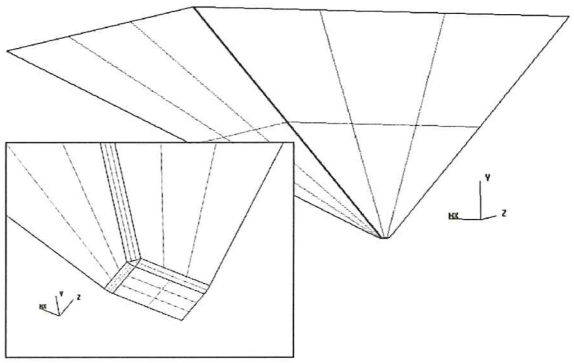


Figure 2: Numerical Vickers indenter

Table 1: Finite Element meshes.

Mesh	Mesh 1	Mesh 2	Mesh 3
Element Size (μm)	0.38	0.05	0.045
Number of elements	4680	5832	8008

The indenter geometry used in the numerical simulations is described by Bézier Surfaces, allowing a fine description of the tip. In this study two Vickers indenters with different offsets (0.0162 μm – Vickers 1; 0.0485 μm – Vickers 2) were used. A top view of the indenter tip presents a rectangular shape where one side is larger than the other (see fig. 2). The mechanical properties of the used materials are presented in table 2. In this table σ_y is the yield strength, ν is the Poisson ratio and K and n are the material constants of the Swift law describing the isotropic hardening.

Table 2: Mechanical properties of the used materials.

Material	σ_y (GPa)	E (GPa)	ν	K (GPa)	n
AISI M2 Steel	4.0	200	0.29	4.00	0.01
Nickel	0.2	220	0.31	1.05	0.08

RESULTS

Influence of the mesh size

Six numerical simulations using the three different meshes and two indenters (Vickers 1 and 2) were performed. The material used was the AISI M2 steel with one friction coefficient equal to 0.16. Figure 3 shows the hardness results obtained with the three meshes. The best hardness values were obtained with mesh 3. However mesh 2 also presents results with good accuracy. Evaluating the CPU time spent in simulations, its possible to conclude that mesh 2 conducts to the best compromise present between hardness results accuracy and CPU time (CPU mesh 2 = 0.4 CPU mesh 3).

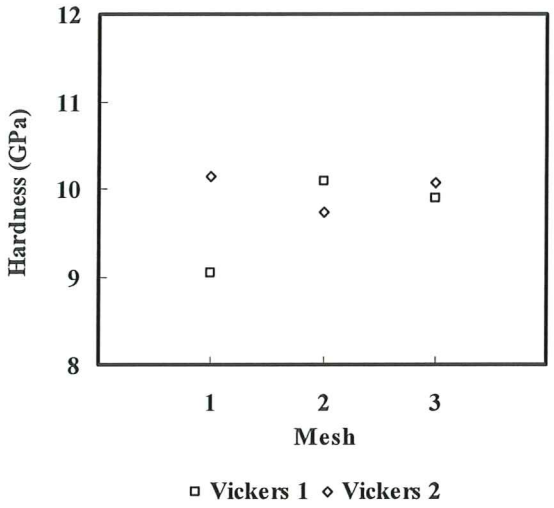


Figure 3 – Hardness distribution for the three studied meshes.

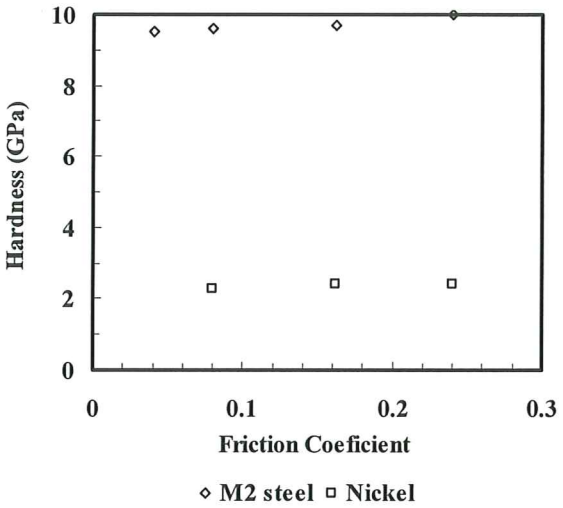


Figure 4 – Hardness distribution obtained with different friction coefficients.

Influence of friction coefficient

Mesh 2 were used with indentador Vickers 2 in this analysis. The materials were the AISI M2 steel and nickel. The use of these two materials was related to the fact that they have significant different hardness values. In this analysis the following friction coefficients were tested: 0.04, 0.08, 0.16 and 0.24 [4, 5].

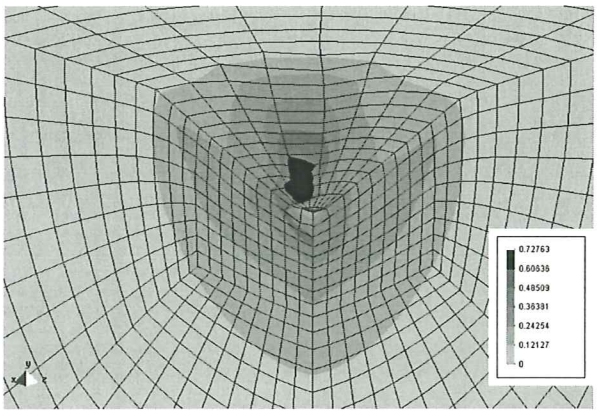


Figure 5 – Equivalent strain distribution on the steel sample for a friction coefficient equal to 0.08.

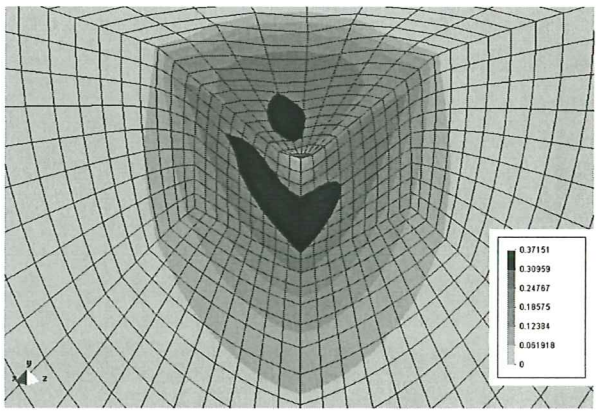


Figure 6 – Equivalent strain distribution on the steel sample for a friction coefficient equal to 0.24.

The hardness value obtained with the different friction coefficients is presented in figure 4. The small variation of the hardness values observed in figure 4 seems to indicate an apparent

independence on the value of the friction coefficient used. However, the analysis of the distribution of equivalent plastic strain at maximum load for the different cases of friction coefficients shows that for low values of the friction coefficient, the plastic deformation sharply increases (for comparison see figures 5 to 8). Low values of the friction coefficient produce a stress concentration on the offset region. Increasing the friction coefficient the stress distribution spreads over the contact region and the equivalent plastic strain is not so concentrated in the offset zone.

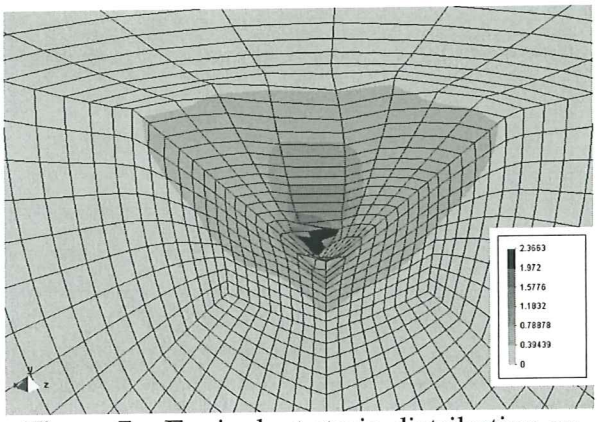


Figure 7 – Equivalent strain distribution on the nickel sample for a friction coefficient equal to 0.08.

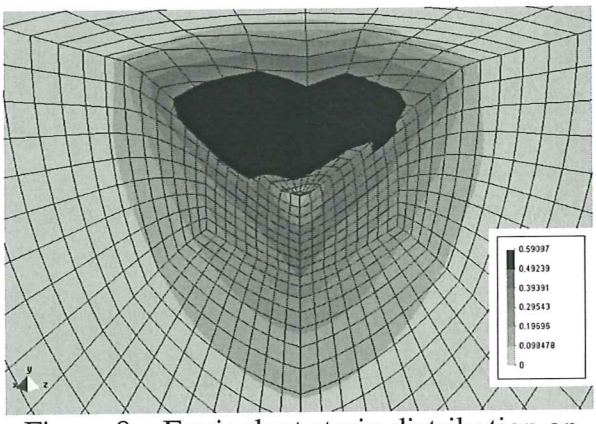


Figure 8 – Equivalent strain distribution on the nickel sample for a friction coefficient equal to 0.24.

CONCLUSIONS

Results of finite element simulations of ultramicrohardness tests are presented. These results highlight the ability of finite element code HAFILM to simulate this type of tests.
The influence on the results of parameters such as the finite element refinement mesh and the friction coefficient between the indenter and the sample are presented. The study presented enables us to conclude that the hardness values are not significantly dependant of the friction coefficient considered. Accuracy is strangely dependent of the FE mesh. Sensitivity studies should always be performed. However, this parameter has a strong influence on the stress and strain distributions over the sample, which can be critical important for the calculations of other mechanical properties.

REFERENCES

1. Menezes, L. F., Teodosiu, C., Three-Dimensional Numerical Simulation of the Deep-Drawing process using Solid Finite Elements, Journal of Materials Processing technology, 97, 100-106, 2000.
2. L. G. Dhatt, G. Touzout, The Finite Element method displayed, Wiley, Chichester, New York, Brisbane, Toronto, Singapore, 1984.
3. L. F. Menezes, C. Teodosiu, A. Makinouchi in: FE Simulation of 3D Sheet Metal Forming Processes in Automotive Industry, Tangungsbericht Der VDI-Gesellschaft Fahrzengtechnik, Düsseldorf, 894, 1991.
4. Lynch, C. T., Handbook of Materials Science, vol. 1-General Properties, CRC Press, 1980.
5. Rother e Tabor, The Friction and Lubrication in solids, Vol. 1, Clarendon Press, 1950.

FINITE ELEMENT MODELLING AND ANALYSIS OF RESIDUAL STRESSES IN AL–SIC METAL MATRIX COMPOSITES WITH GID®

A. Andrade–Campos[1,*#], J.A.M. Pinho–da–Cruz[1,#] and F. Teixeira–Dias[1]

[1]*Dept. Mechanical Engineering, University of Aveiro, 3810–193 Aveiro, Portugal*

SUMMARY: Residual stresses are often induced in metal matrix composite materials (MMC) when these are cooled down from fabrication to room temperature. This is mainly due to the difference in the coefficients of thermal expansion of the matrix and reinforcement materials. In the present paper, these stress fields are studied with a fully three–dimensional thermomechanical model. The strength differential in subsequent uniaxial tension from the thermal residual stress state is also analysed. The thermomechanical model implemented to describe these processes considers that the reinforcement component has a thermoelastic behaviour and that the metallic matrix exhibits a thermoelastic–viscoplastic behaviour. The Finite Element analyses presented are based on three–dimensional geometrical composite unit cell models built with the GiD® preprocessor. Results obtained with long cylindrical SiC fibres embedded in an aluminium matrix are presented. All the process data are introduced in a dedicated problem type with a graphical user–friendly input data interface (GUI) built inside GiD®. Postprocessing tasks are also carried out with GiD®, graphically presenting, for example, the triaxial stress state, the equivalent stress, the equivalent plastic deformation rate and the resulting displacement fields in the composite unit cell. The results are then bounded with an analytical model and compared with numerical results from other authors.

KEYWORDS: Preprocessing, postprocessing, metal matrix composites, residual stresses.

INTRODUCTION

Metal matrix composite materials (MMC) have become increasingly attractive in recent years for their high strength and creep resistance properties. Thermally induced residual stresses are developed in MMCs during cooling from the fabrication temperature to room temperature. For discontinuously reinforced MMCs this has often been attributed to the mismatch between the thermal expansion coefficients of the fibre and matrix materials. The subsequent tensile flow stress of the composite is influenced by the thermal residual stresses generated during cooling down from fabrication temperature[1–4].

In the present work, three–dimensional (3D) unit cell Finite Element (FE) models are developed for composite materials with cylindrical reinforcements in order to analyse the magnitude and distribution of the residual stresses that result from the cooling process and their effects on subsequent deformation in tension. 3D FE models were chosen for their advantages over axisymmetric and two–dimensional models in that they consider a more realistic fibre geometry at fibre ends and do not ignore a portion of the matrix material. In this kind of research it is crucial to have an adequate graphical support to built the model,

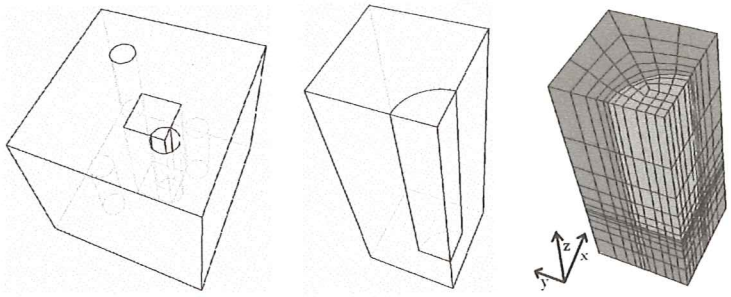
* Corresponding author: Tel: +351.234.370830; Fax: +351.234.370953; <gilac@mec.ua.pt>

# These authors acknowledge financial support given by Portuguese Foundation for Science and Technology.

introduce the process data, execute the analysis modules and analyse results. The authors used the pre/postprocessor GiD® to perform all the graphical tasks. The model was created using GiD®'s 3D mesh generator. A dedicated problem type was built in order to accept process input data and, along with the coordinates of the nodes and element conectivities, write into a file to be read by an analysis module. This module calculates residual stress fields and strength differentials in dual phase materials. Post–processing was also performed with GiD®.

MODELLING CONSIDERATIONS

It is assumed that all the reinforcement fibres have the same dimensions and orientation and are uniformly distributed. The matrix and fiber materials are isotropic in stiffness and thermal expansion. A perfect bonding between the aluminium matrix and the SiC reinforcement is assumed. The behaviour of the reinforcement is considered to be thermoelastic and the matrix to exhibit a thermoelastic–viscoplastic constitutive behaviour. The temperature field among the composite is assumed homogeneous at all times.
The representative unit cell, shown in figure 1, is hexahedral and includes the cylindrical reinforcement fibre. An orthogonal cartesian coordinate system was used as reference with *Ox*, *Oy* and *Oz* axes aligned with the main dimensions of the unit cell. The longitudinal axis of the reinforcement fibre is aligned with the uniaxial loading direction[2,3].



(a) (b) (c)

Fig. 1: Representative unit cell (RUC) for the cylindrical reinforcement MMC. (a) Geometrical material model, (b) RUC and (c) finite element mesh.

During the first stage of the analysis, the unit cell is subjected to a thermal load simulating the cooling down from fabrication (493K) to room temperature (293K). On the second step of the analysis, the composite unit cell is submitted to an uniaxial longitudinal tensile load. Symmetry and restraint conditions are the same in both steps. A uniform displacement is applied to all the nodes on the top surface of the unit cell, along direction *Oz*. The model consists of 689 hexahedral trilinear eigth–noded finite elements (fig.1c). The problem type TriCreator is defined by the (i) boundary conditions; (ii) material properties; (iii) material assignment; (iv) time and numerical optimisation data and (v) problem general data.

GRAPHICAL AND NUMERICAL IMPLEMENTATION

The conditions that can be assigned to an entity in the problem type TriCreator are: (i) *Point/Line/Surface–Constraints,* restraining the displacements along the *Ox*, *Oy* and/or *Oz*; (ii) *Point/*Surface–*Load*, loading entities as boundary forces; (iii) *Boundary–Displacements*, assigning nodes to move a predefined displacement. These conditions can be assigned either on the geometry and/or on the mesh. It is also possible to change material properties in the *Data Materials* menu box. The *Interval data* window contains the numerical optimisation parameters that control the iterative scheme in the analysis module. Problem data are the initial and final temperatures, cooling rate and total process time.
Once the conditions, materials, problem and interval data are properly assigned, GiD® creates

an input file for the analysis module (fig. 2). The solver – Nostradamus – is dedicated to the numerical calculation of both thermally and mechanically induced residual stresses in dual–phase materials. Nostradamus produces two output files. These include gauss points lists, stresses, strains, etc. Another module, designated Sirius, was developed to interface the output files into postprocess files.

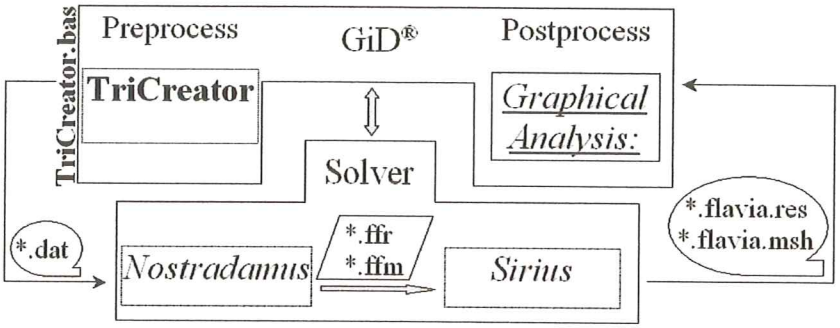


Fig. 2: Schematic representation of the Nostradamus–GiD® interactions.

RESULTS AND POST–PROCESSING

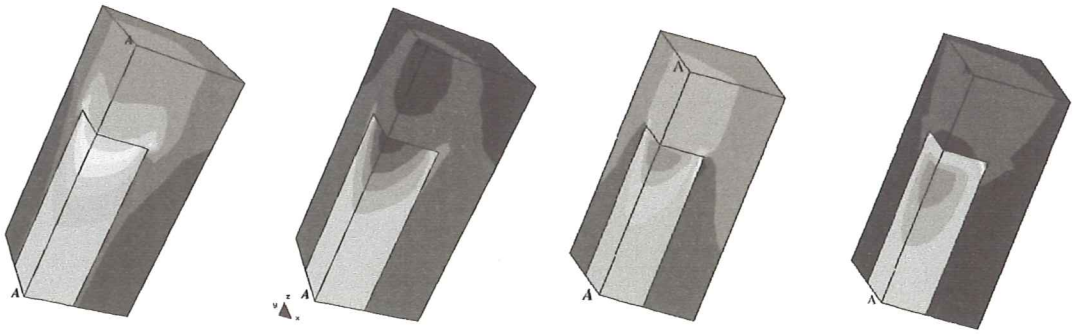
Numerical results were analysed using the post–processor GiD®. After the cooling stage, large stress gradients occur close to the fibre–matrix interface (figs. 3(a) and 3(b)), especially near the top corner of the fibre. The magnitude of the compressive stress is maximum at the fibre–matrix interface and progressively decreases as one moves away from the interface into the matrix. At 1% tensile strain the longitudinal stress, σ_{zz}, evidences high stress gradients near the fibre ends. These stresses are tensile throughout the unit cell, with its the maximum value near the fibre top corners. The von Mises equivalent stress, σ_{eq}, has strong gradients localised in the corner regions and in the fibre–matrix interface. The evolution of the stress fields during the far field loading, up to 1% strain in tension, are shown in figures 3(c) and 3(d).

DISCUSSION

In order to validate the numerical results obtained with the Nostradamus–GiD® system, these were compared with results obtained by another author (Jain et al.[1]), using different mechanical models and bounded with analytical calculations. The chart in figure 4 represents the thermal residual stress and the longitudinal stress profiles along the AA direction (fig. 3). The differences displayed in figure 3 can be attributed to the differences in the constitutive model which Jain et al.[1] consider to be elastic–plastic.

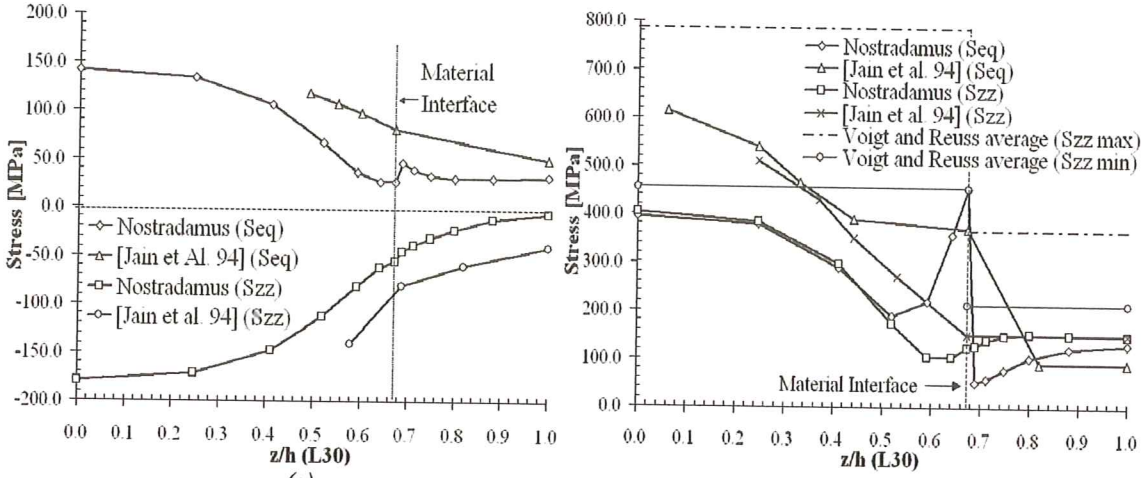
The thermomechanical loading of a composite material unit cell leads to non–uniform local (microscale) stress and strain fields, whose analytical descriptions correspond to complex problems. However, many simplified and useful results can be obtained in terms of average stress and strain values[5]. In this context, the elastic constants of composite materials can be evaluated with the aid of mean field methods. These techniques assume that averaged values of the stress and strain are representative of the behaviour of each material phase.

The earliest applications of mean field methods to compute the elastic constants of composite materials are associated to the works of Voigt and Reuss. These investigators derived simple expressions for the elastic constants of the composite that depend only on the elastic constants and volume fraction of the phases, and were independent of the phase geometry and spatial distribution. These expressions represent, respectively, upper and lower bounds for the composite's stiffness. Based in the Voigt and Reuss average stiffness matrices[6], the values of "local" average elastic stresses at the reinforcement and at matrix materials were obtained considering the total corresponding thermomechanical strains. These average stresses correspond to the Voigt (upper) and Reuss (lower) bounds (figs. 4(b)).



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

Fig. 3: Residual Stress fields at room temperature: (a) σ_{zz} and (b) σ_{eq}. Uniaxial stress fields at 1% strain: (c) σ_{zz} and (d) σ_{eq}.



(a) (b)

Fig. 4: Stress profiles along direction AA. (a) Thermal residual stresses and (b) strength differential after the initial residual stress state.

CONCLUDING REMARKS

An example of application of GiD® in science and engineering was presented, confirming the importance of the pre and postprocessor in the analysis of the overall behaviour of multiphase materials. Stress gradients were determined in the surrounding area of the fibre–matrix interface and at the top corner of the fibre. The results of the simulations of thermal residual stresses and prediction of the strength differential are in qualitative agreement with other authors work. It is worth to notice that both Voigt and Reuss bounds define stress ranges whose values are superior than the numerical ones. This can be explained by the fact that analytical bounds are inherently elastic, while numerical results are predominantly elastoplastic.

REFERENCES

1. M. Jain, S.R. MacEwen and L. Wu, *Mater. Sci. Eng.* **A183**:111–120, 1994.
2. F. Teixeira–Dias and L.F. Menezes, *Int. J. Numer. Meth. Engng* **50**(3):629–644, 2001.
3. F. Teixeira–Dias and L.F. Menezes, *Comput. Mater. Sci.* **21**(1):26–36, 2001.
4. G. Meijer, F. Ellyin and Z. Xia, *Composites Part B* **31**(1):29–37, 2000.
5. R. Hill, *J. Mech. Phys. Solids* **11**:357–372, 1963.
6. S. Nemat–Nasser and M. Hori, *Micromechanics: Overall Properties of Heterogeneous* Materials, 2nd Rev. Edition, Elsevier Science B. V., 1999.

GID CUSTOMIZATION FOR APPLICATIONS IN METAL FORMING OPTIMIZATION

Pablo A. Muñoz-Rojas[1,2], Jun S. O. Fonseca[2] and Guillermo J. Creus[2,3]

1 Departamento de Engenharia Mecânica, Universidade do Estado de Santa Catarina, Brasil.
2 PROMEC - Universidade Federal do Rio Grande do Sul, Brasil.
3 CEMACOM – Universidade Federal do Rio Grande do Sul, Brasil.

SUMMARY: This paper presents a successful approach for preform and die-shape optimization in metal forming applications. The procedure is controlled by a shell program which makes use of GiD, METAFOR (a Metal Forming Finite Element Code) and some supporting routines in order to generate data for a non-linear mathematical programming algorithm. To this end, GID's pre-processing capabilities are customized to automatically generate input data for METAFOR and provide necessary information for the evaluation of preform shape sensitivity. After the optimization procedure is complete, GiD is called again for visualization purposes.

KEYWORDS: metal forming, optimization, preforms, die-shapes.

INTRODUCTION

The aim of the presented procedure is to optimize preforms or die-shapes in order to achieve a desired final geometry for a given component. This requires the selection of an appropriate boundary parameterization with its related design variables, an objective function, and efficient simulation and optimization codes. The procedure descibed is restricted to 2D applications although it could be extended to comprise 3D situations.

Concerning the preform boundary parameterization, the use of nodal coordinates of the FE mesh as design variables is not a good choice (Haftka & Grandhi,1986). On the other hand, Braibant & Fleury (1985) showed that B-Splines are well suited for the task. In this regard GiD presents its first advantage since it allows to define the geometric model using NURBS (Non Uniform Rational B-Splines). Once the preform boundary parameterization method is defined, one should choose which design variables to adopt. Based in some previous experience (Muñoz-Rojas et al., 2001), NURBS control point coordinates are selected. In the case of die-shapes optimization, the die boundaries are defined in terms of line and arc segments. In the case of lines, the design variables are the beginning and ending coordinates of the segments, while for the arcs, the radii and arc centers also take this function. The objective function adopted is the one proposed by the authors (Muñoz-Rojas et al., 2001) and the GCMMA – Globally Convergent Method of Moving

Asymptotes (Svanberg 1998) is adopted for optimization. The simulation code employed is METAFOR (Ponthot & Hogge, 1991), an elastic-plastic FE program specially developed for metal forming analyses at the University of Liège, Belgium.

GID CUSTOMIZATION

Some important features that GiD presents make it well suited for this application: the geometric representation includes B-Splines, automatic transfer of boundary conditions from geometric representation to FE model, possibility of operation in batch or interactive modes, and easy customization for solver data generation.

The first aspect that needs attention is GiD's customization to generate data for METAFOR. This is accomplished generating a *problem_type metafor*, i.e., a directory named *problem_type metafor.gid* containing the following files: *metafor.cnd, metafor.mat, metafor.prb, metafor.sim, metafor.geo, metafor.bas and metafor.bat*. These files must be adapted to METAFOR's needs according to the guidelines given in GiD's reference manual. In the present customization it was found best not to use *metafor.bas* and *metafor.bat*. The other files were modified as follows:

The file *metafor.cnd* contains the contact definitions and parameters, i.e., each boundary curve of the component was associated to a possible contacting die, contact model, friction coefficient, etc.; the file *metafor.mat* contains the material properties including the yield stress, the isotropic and kinematic hardening moduli; and the file *metafor.prb* contains algorithmic parameters existent in METAFOR such as the number of load steps, tolerance for equilibrium iterations, etc.

Once the *problem_type_metafor* is defined it is possible to call GiD from within another program using the batch file loading directive. Using these ideas it is possible to define a base batch file, containing the current geometric configuration and perturb the design parameters in this file in order to obtain perturbed responses calculated via METAFOR. These perturbed responses may then be used to calculate the gradients needed in the optimization algorithm. When all the gradients are calculated, the managing code calls the optimization algorithm which provides a new design associated to a lower objective function. In the sequel the base batch file is updated with the values of the new design. This process is repeated until convergence is obtained. The overall procedure is represented in Box 1.

GID'S SUPPORT TO PREFORM SHAPE SENSITIVITY CALCULATION

As it has been pointed out, GiD also plays a key role in the calculation of preform shape sensitivities. For this purpose it is possible to include the instruction *writeascii path\file.asc* at the end of the file *batchfile.bat*. This will save in *path* the file *file.asc* which contains all the information generated and used by GiD. Among all the information available one can gather the parameters that define the NURBS existent in the geometric model, i.e., the knots and the control points. This allows to reconstruct the expressions for these curves and to determine the velocity field analytically on the boundary. A Laplacian smoothing may be used then to obtain the velocity field in the domain. More details can be found in Muñoz-Rojas et al. (2001).

Box 1. Overall algorithm.

1. Define optimization case: preform / die-shape
   Call GCMMA algorithm

   GCMMA algorithm
2. Set initial data: design variables initial values, objective function and constraints definitions, etc.
3. Calculate objective function, constraints and respective gradients in the current design point: call GiD in batch mode loading file base1.bat. Call METAFOR to calculate the responses of the original and perturbed configurations. Use the information to calculate the sought quantities.
4. Use the gradients to construct the functions approximations and minimize the subproblem.
5. If the new design satisfies a merit function then
   use the design variable values found to redefine file base1.bat
   Else
   relax the functions approximation parameters to obtain a more conservative one, keeping the same gradients. Go to 4.
   End If
6. If design variables and objective function converge
   STOP
   Else
   Go to 3
   End If

DIE-SHAPE OPTIMIZATION EXAMPLE

A die-shape optimization problem is defined by 12 design parameters. Their initial and optimized truncated values are depicted in Table 1 together with the corresponding objective function. Fig. 1 shows the geometric disposition of these design variables in the lower die. The upper die is defined so that the space between dies is 2mm. Fig. 2 shows the desired final shape of the sheet in red and the optimized one in black. Note that the springback effect is taken into consideration as displayed in Fig. 3. The properties adopted are: $E=2.1\times10^5$ MPa, $\nu=0.3$, $\sigma_E=250$ MPa, H (isotropic hardening modulus)= 7 MPa and friction coefficient = 0.1.

Table 1

	DESIGN VARIABLES												
	X4	**X7**	**X8**	**X9**	**X11**	**Y4**	**Y6**	**Y8**	**Y9**	**R1**	**R4**	**R5**	**Obj. F.**
Initial	180.00	178.00	139.00	114.00	55.00	50.00	32.00	65.00	19.00	11.50	5.00	16.00	1.0000
Final	180.22	178.94	138.40	114.72	54.80	50.37	32.00	66.42	19.49	11.58	4.99	15.99	0.1152

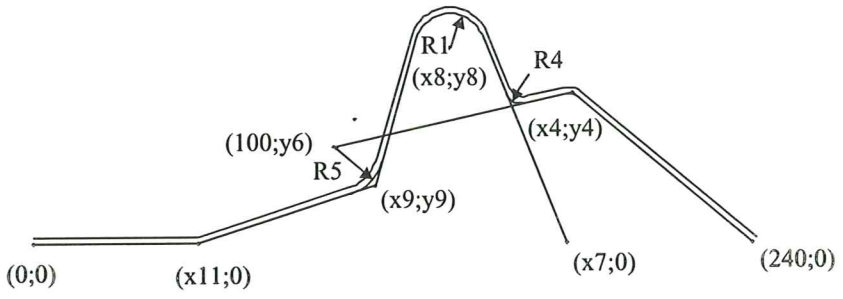


Figure 1: Die-shape parameterization.

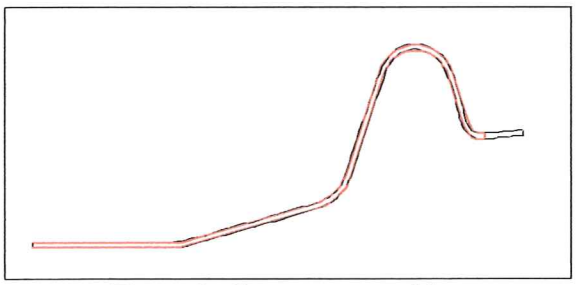


Figure 2: Geometry matching

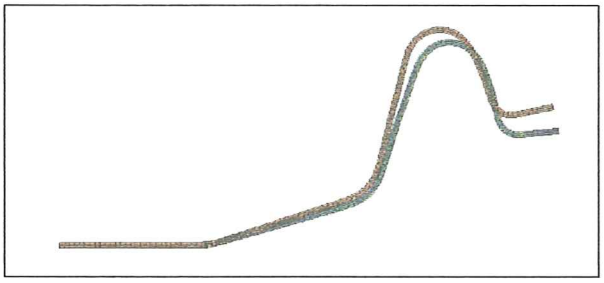


Figure 3: Springback return

CONCLUDING REMARKS

It has been shown that GiD may be employed as an adequate tool in software integration for optimization of die-shapes and preforms in metal forming. The features that make this integration possible are: geometric representation including NURBS, easy customization for solver data generation, batch mode option, automatic transfer of information from geometric model to FE mesh, and availability of an ascii file containing the information used and generated by GiD. The successful application of the proposed procedure is shown by a die-shape optimization example.

ACKNOWLEDGEMENTS

The financial support given to this work by CNPq and CAPES is greatly acknowledged. The first author would also like to express his gratitude to UDESC for the waving of his lecture duties during the development of his doctoral studies. METAFOR and the GCMMA codes are used under cooperation agreements with Prof. M. Hogge (University of Liège) and Prof. K. Svanberg (Royal Technical Institute of Sweden) respectively.

REFERENCES

1. Haftka, R.T. & Grandhi, R.V., "Structural Shape Optimization – A Survey", *Comput. Methods Appl. Mech. Engrg.* 57, pp. 91-106, 1986.
2. Braibant, V. & Fleury, C., "Shape Optimal Design Using B-Splines", *Comput. Methods Appl. Mech. Engrg.* 53, pp. 119-148, 1985.
3. Muñoz-Rojas, P.A, Fonseca, J.S.O. & Creus, G.J., "A New Approach for 2D Preform Optimization Using an Elastic-Plastic Material Model and Unstructured Mesh Generators", *CILAMCE*, UNICAMP, SP, Brasil, 2001.
4. Svanberg, K., "The MMA for Modeling and Solving Optimization Problems", *Third World Congress of Structural and Multidisciplinary Optimization – WCSMO3*, 1998.
5. Ponthot, J.P. and Hogge, M., "The Use of the Euler Lagrange Finite Element Method in Metal Forming Including Contact and Adaptive Mesh", *in: Proceedings of the ASME Winter Annual Meeting, Atlanta*, USA, 1991.

MODELLING OF POWDER BASED COMPLEX SHAPE MANUFACTURING

O.Mikhailov, M.Shtern

Institute for Problems of Materials Sciences, Krgiganovsky st. ,3, 03680 Kiev, Ukraine, e-mail: olmi@alfacom.net, mbsh@i.com.ua

SUMMARY: The internal version of finite element method aimed at the simulation of powder compaction is suggested. The powder material behaviour is described by the modified relationships of plasticity theory for porous bodies. The creation of initial geometric model as well as the generation of finite elements mesh, formulation of material properties, formulation of boundary condition and visualization of final results are carried out on the base of package GID. Various diagrams of die-compaction were considered. Both external friction and pressing diagram effect on final properties of part are analyzed. Obtained results are compared with experimental data of density field which are closed each other.

KEYWORDS: *powder compaction, metal forming, finite element analysis, numerical simulation*

INTRODUCTION

The computer simulation is used here to improve existing metods of manufacturing the PM - parts. The study of distribution of the density fields - is in the focuss of interest. The effect of the pressing diagrams on finite properties of parts is studied. All considerrations bellow are connected with part illustrated by Fig. 1.

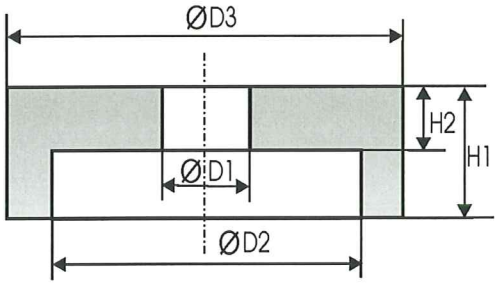


Fig. 1 : The configuration of part

The experimental information concerning the final distribution of a density and all geometrical parameters of part are taken from work [1]. The first of all the simplest diagram is considered: two lower punches are immobile. Then the diagram for which rates of punches are proportional to current heights will be analyzed. Finally, the pressing diagram suggested in [1, 2] will be considered. In the last case the process of pressing consists of three stages.

1.Methods of simulation

The simulation is carried out by use of the finite element method. The modified relationships of plasticity theory for porous body are used. It has been considered rigid - plastic behaviour of material.

The similar ideas were used in [3,4]. The boundary value problem reformulated in a finite element form has been solved using the step by step computation. For each step of loading (or straining) the system of non-linear algebraic equation has been solved by use of Newton – Rafson procedure. Iterations were finished if the convergence criterion was satisfied. On the base of obtained rate field the density field as well as the stress components has been defined. Than the same procedure has been carried out for the next step of loading. The rate of convergence was controlled by the choice of initial estimate. For this purpose the extrapolation on the base of previous rate field (obtained for previous moments) has been used. Besides we have used solution obtained on the base of linearization as a first estimate.

All punches, core – pine and sleeve are supposed to be rigid. At the surface of these elements the external friction is taking place.

To prevent the complications due to the strong deviation of the mesh of elements (because of deflection) special procedure has been used. The given mesh of finite elements is mapped at the final configuration of part. Then it was mapped onto initial configuration.

The creation of initial geometric model, generation of finite elements mesh, formulation of material properties, formulation of boundary condition and visualization of final results are carried out on the base of package GiD: pre- and post- processing system for F.E.M. calculations (International Center For Numerical Methods In Engineering CIMNE).

Academic and professional (one month password) versions of system have been used.

The system GID was customized and configured by means of creation of files: conditions file (.cnd), materials file (.mat),problem and intervals data file (.prb), template file (.bas), command execution file (.bat).

The examples of pre-processing and post-processing are presented at the Fig. 2, Fig. 3.

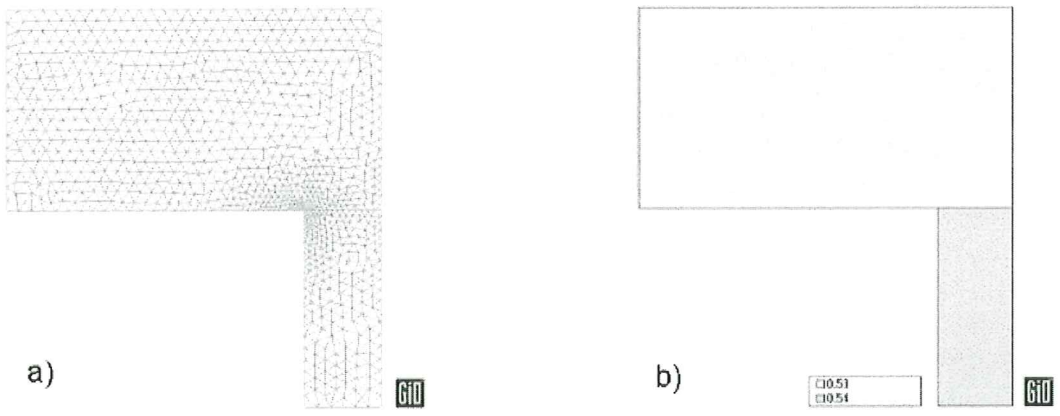


Fig. 2 : Pre-processing: a – mesh generation; b – conditions and materials properties (relative porosity)

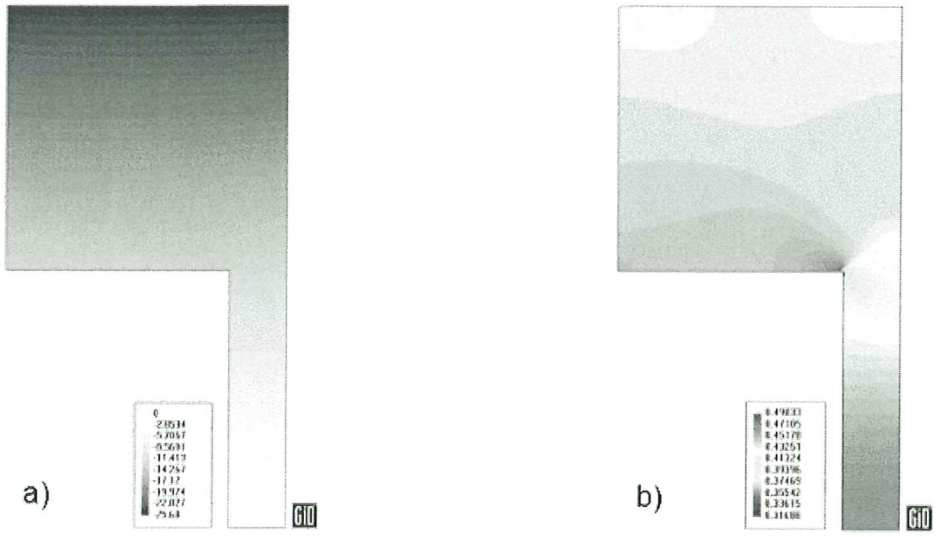


Fig. 3 : Post-processing: visualization of results (a - contours for equal axial component of velocity; b - contours for equal density)

2. **Obtained results**

One-side pressing. The given diagram is the simplest one, but can be practically used only when the ratio of height /diameter is very small. In our case the density variation is very significant. The shear strain rate achieves the largest value in the neighborhood of internal angle. It should be emphasized that the contours of equal shear strain rates remind to the paths of the overconsolidated cracks propagation are known from practice [2].

Diagram for which rates of punches are proportional to current heights. One should note that in condition of the absence of external friction the given diagram unlike to one – side pressing provides the uniform density distribution [5]. The presence of external friction results in irregular distribution of a density. Controlling a direction of external friction, it is possible to reduce the noted effect, however nonuniformity of a density remains.

Modified pressing diagram. Unlike to both previous diagrams in the case considered in paper [1] rates of punches is variable for period of compaction (Figure 4).

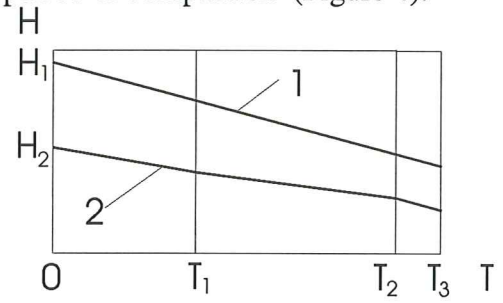


Fig. 2 : The pressing diagram, suggested in paper [1, 2]; 1 –upper punch, 2 –lower punch

Pressing diagram now consists of three parts. At the first stage the rate of the second punch - V_2 is less then V_1. At the second stage the second punch passes ahead the first one and at the third stage $V_2 = 0$. The initial properties is not uniformly distributed. The densities of the upper and lower parts are correspondetly 3.5 g/sm^3 and 3.4 g/sm^3.

This stipulates the same level of shear strain rates that in previous case and provides the most uniform density field.

The results of computer simulation and experimental data [1,2] are closed each other.

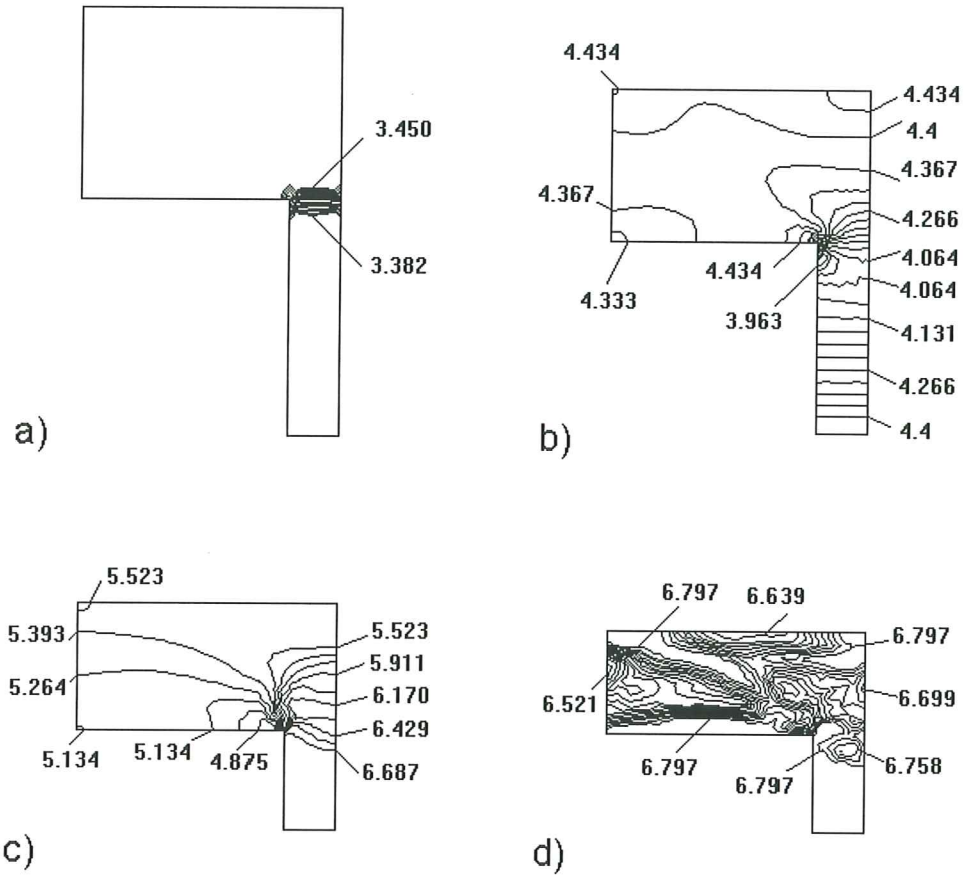


Fig. 5 : Contours for equal density (a –T=0.01s, b - T=0.44s, c – T=1.03s, d – T= 1.16 s)

3. Conclusions

GiD can be customized and configured by users. In our case it has allowed to modelling of powder based complex shape manufacturing

REFERENCES

1. Doremus, P.,Pavier, E., Kergadalan, J., Puente, G., Axisymmetric Part Compaction: Data Base For Numerical Simulation, *The International Journal of Powder Metallurgy*, Vol.**35**, N3, 1999, pp. 63 –68

2. Kergadalan, J., Puente, G., Pavier, E. and Doremus, P., Compression of an Axisymmetric Part with an Instrumented Press, In *Proc. Int. Workshop on "Modelling of metal powder forming processes,* Grenoble, France, 1997, 227 -285.

3. Zang, Z.L.and Niemi, E., A class of generalized mid-point algorithms for the Gurson-Tvergaard material model, *Int. j. numer. methods eng.*, **38**, (1995), 2033-2053.

4. Mikhailov, O., Integrated scheme of computer simulating for PM articles pressure treatment, *Powd. Metall. Metal Ceram., (translated from Russian)*, N **9/10**, (1995), 99-104

5. Yurchenko, E. and Shtern, M.Control of Complex Forms by the Density Distribution During Axial Compaction of Objects, *Powder Metall. Metal Ceramics* **4** (1993), 286 - 291

PECULIARITIES OF THE USE OF PACKAGE GID FOR THE ANALYSIS OF FORMING PROCESSES IN POWDER METALLURGY

M.Shtern, O.Mikhailov

Institute for Problems of Materials Sciences, Kiev, Ukraine

Some features of mechanical models of behaviour of powders sensitive to consolidation phenomenon are suggested. They are used to formulate the boundary – value problems describing the die - compaction in Finite Element form. The numerical code, including the procedure of the regularization is described. Specialties of the formulation of boundary – value problems in powder metallurgy are discussed. The approach to predict the failure under the die – compaction is suggested.

Introduction

The penetrating of powder materials in different branches of industries stipulates the increase of interest to analysis of powders behaviour under the forming processes. These processes including operations of compaction and sintering are predominantly targeted at the manufacturing of complex shape parts. Meanwhile the failure under the compaction and deflection under sintering inhibit the dissemination of Powder Technologies. Together with technological approaches usually used in Powder Metallurgy preliminary numerical analysis is considered now as the effective method providing the minimizing of the influence of factors above noted. It has been made possible due to essential progress in comprehension of main features of powder behaviour during compaction achieved last decades. Developed in framework of special scientific and industrial programs and based on modern notions of continuum mechanics the theory of compaction allows the definition of the density field and pressure during pressing as well as the shape deviation during the sintering. Further implementation of main results of the given theory into well-known users friendly packages, in particular GID [1] has made the prediction of main features and control of the die –compaction to be available even for production engineers without mechanical or mathematical education.

Meanwhile further dissemination of mentioned packages is restrained by some reasons. The most important among them is the absence of consensus relating the failure criteria. Another one – is the general formulation of the constitutive model, including the description of the sintering. In turn, the general structure of the constitutive relationships defines special features of numerical analysis.

Main Features of Constitutive Models in Powder Compaction and Sintering

Main relationships underlying the computer analysis of powder compaction and sintering are predominantly based on continuum hypothesis. All equations are formulated in the form inherent to continuum mechanics and contain parameters defined at the point of continuum. Among them general parameters describing the state of powder continuum – stress tensor, strain rate, velocity field, density, temperature as well as internal local variables such as accumulated matrix strain, cohesive measures, porosity, damaging. Constitutive equations relating stress tensor components with strain rates are usually [2,3] derived from the expression for constitutive potential Φ_e, which

is supposed to be written in special form $\Phi_e = \Phi_e(w)$, where w -is the smooth , convex and first order homogeneous calibrating function of strain rate components. As result the stress tensor components can be presented in the form

$$\sigma_{ij} = \Phi_e'(w)\frac{\partial w}{\partial e_{ij}} \quad . \tag{1}$$

Function $\Phi_e = \Phi_e(w)$ is determined both theoretically and from the experimental analysis [2,3]. It is usually connected with the rate sensitivity of matrix or powder mechanical response, whereas $w = w(e_{ij}, \rho)$ is sensitive to structure of material (shape of pores, particles, cohesive properties). In particular, Φ_e can be written so that the stress – strain rate relationship has the form

$$\sigma_{ij} = p_o\delta_{ij} + \varphi\frac{\sigma_{01}}{w}\left(e_{ij} + \frac{((1+\nu)\dfrac{\sigma}{\sigma_{01}} - (1-2\nu))}{3(1-2\nu)} e\delta_{ij} \right) , \tag{2}$$

containing two material functions ν and φ of porosity and other material parameters, defined and introduced in [4]. When

$$p_0 = -\sigma_{01}\sqrt{\frac{1}{3}\varphi\frac{1+\nu}{1-2\nu}} \tag{3}$$

and the equivalent stress σ – is the function of equivalent strain $\omega = \int_0^t w dt$ the given model is reduced to extended Cam – Clay relationships and describes the behaviour of powders ([4]).

Internal variables are satisfying to special evolution equations. In particular the change of relative density ρ is describing by equation following from mass conservation law

$$\frac{d\rho}{dt} = -\rho \ div\,\vec{v} , \tag{4}$$

whereas the equivalent strain is satisfying the equation

$$\frac{d\omega}{dt} = \sqrt{\varphi\gamma^2 + \frac{1}{3}\varphi\frac{1+\nu}{1-2\nu}e^2} \tag{5}$$

For hot pressing and sintering equivalent stress σ –is a function of w. Besides, right hand of equation (2) for sintering contains the additional term P_L , responsible for volume change without external forces.

Specialties of Boundary Value Formulation and Numerical Code

Unlike to conventional problems of metal forming, where the main attention is focussed at the analysis of stress and velocity fields the description of powder metallurgy processes is usually mainly concentrated at the fields of internal parameters, which directly define the service properties of powder parts. Therefore the evolution equations (4), (5) and some other possible relationships induced by additional parameters can give the desired distribution. For this aim, however, the rate field is needed, that in turn requires the consideration of constitutive laws together with

corresponding extremal principals. Mentioned feature defines the specialties of numerical procedures based on step by step integration of evolution equation. The preliminary finite element discretization is supposed to be carrying out. At the first step internal parameters are assumed to be equal to corresponding initial values. Non – linear algebraic equation following from constitutive relationships and extremal principals are solved then relating the nodal rates using iterative calculations. After the first step of the integration new values of material parameters are determined and procedure above noted is carried out f or the next step. It should be noted that procedures described here require the use of some type of regularization. One of them is inherent for the consideration of rigid – plastic flow, where the possible existence of rigid regions does not allow the obtaining of unique rate fields. To avoid this feature authors use the relation between equivalent stress and strain rate in regularized form

$$\sigma = \sqrt{1+n^2}\,\sigma_0 \frac{w}{\sqrt{w^2 + n^2 \gamma_0^{\,2}}} \qquad (6)$$

where n – is supposed to be the small value, and γ_0 has the dimension of strain rate. Another kind of regularization is used to provide the convergence of the iterative procedure.

There are possible two approaches to describe the powders flow. The conventional, based on assumption that the mass of cell is not changed, allows the most detail description of rate field. But on the other hand it hampers the analysis of the situation near singular points (corners). The modified approach – permeable elements method based on assumption that the mass of cell can be changed in accordance with the balance of mass for neighbour cells ([5]). Since the mesh evolution here is prescribed, problems of singular points can be easily avoided. But on the other hand the given approach is not sensitive to detail of the strain rate field.

On the Failure Criteria Under Die - Compaction

The modeling of die – compaction is usually targeted to solve two problems: to control of the density distribution, taking into account both the external friction and pressing diagram effect and to avoid the failure of billets or to predict the appearance and further behaviour of cracks and damages. The solution of the first of problems above can be obtained directly from the integration of boundary – value problem, where the density distribution is directly defined. On the base of obtained distribution all service parameters, depending on porosity are defined as well.

At the same time the prediction of failure usually requires the additional assumptions as a rule in a form of special criteria sensitive to compressive stresses. The most disseminated among them is that following from Drukker – Prager – Cup model. It should be noted that the given approach is closed to conventional mechanical consideration of failure: it only indicates the moment of failure and is not sensitive to the nature of destruction.

Authors use the approach based on stability analysis [6]. Since the model defined by equations (2) – (5) consists only the density as evolution parameter, the material flow is stable when the density grows and unstable under the diminishing of ρ. The loose of stability induces the possible appearance of the shear bands, which may be considered as initial manifestation of overconsolidated cracks. For density controlled model the loose of stability is possible for stress state, for which the potential contour is parallel to hydrostatic axis in space stress tensor components. The given situation is possible for extended Cam – Clay model introduced in [4].

Here the range of the stress states may include the point of pure shear (for extended Cam – Clay model the given state is possible for large compressive stresses) and, moreover, points located left of the peak of contour. It also implies the appearance of the regions in compacted body, when the diminishing of density is possible, that clarifies the nature of destruction as a process following by local softening.

References

1. Oliver Javier, Cante,J., Advances in Finite Element Modeling of Die Compaction at CIMNE, in Proceeding of Powder Metalurgy European Congress,22 – 24 October, 2001, Nice, France
2. M. Shtern & A.Cocks, The Structure of Constitutive Laws for the Compaction of Metal Powders, in Recent Developments in Computer Modeling of Powder Metalurgy Processes, IOS Press, 2001, pp.71 – 82
3. A.C.F.Cocks, Constitutive modelling of powder compaction and sintering, Progress in Materials Sciences, **46,** 2001, pp.201 – 229
4. M. Shtern, O.Mikhailov, Defects Formation in Die Compaction: Prediction and Numerical Analysis, in Proceeding of Powder Metalurgy European Congress,22 – 24 October, 2001, Nice, France
5. E.Olevsky, M.Shtern, G.Timmermans, L. Froyen, L.Delaey, The permeable element method for modelling of deformation processes in porous and powder materials, Powder Technology, 1997, N4,pp. 19 -29
6. Rice J., The Localization of Plastic Deformation, in Proceeding of the 14th IUTAM Congress, Delft, The Netherlands, Edited by W.T.Koiter, North-Holland Publishing Company, Amsterdam-N.-Y., - Oxford, 1977, 439 – 472