Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Automatic Creation of Blending Surfaces in Hydropower Generators Turbine Blades

Emiliano G. Castro *,1 **Marcos S. G. Tsuzuki** *,2
Emilio C. N. Silva *,3 **Sandro Vatanabe** *,4 **André K. Sato** *,5
Thiago C. Martins *,6 **Giulliano B. Gallo** **,7
Marco A. Marques **,8 **Hamilton Tiba** **,9

* *Escola Politécnica da Universidade de São Paulo, Brazil.*
Computational Geometry Laboratory
** *Companhia Energética do Estado de São Paulo, Brazil*

Abstract: Hydropower generators turbine blade is a one of kind product that requires specific maintenance. They are designed in very specific software tools based on features, which allow good CAD/CAE automation by simply tweaking parameters. However, such CAD/CAE automation has brought an undesired side effect: it's hard to add new features that are not considered in the original CAD. Specially in simulation-optimization applications, where parameters that are not considered in the design must be modified. This is the case in this work, where fluid-structure analysis is performed and a geometric feature that transcend the design activity (blending surface between the crown and the blade) is modified. This paper describes an ongoing project which involves the implementation of a module able to intercept the geometry generated by a CAD program using the IGES format, create/modify a new geometric feature and update the fluid volume and blade geometries. The surface boundary curves are extracted, and the adjacency relationship between the surfaces is determined by coordinate numerical comparison. The surfaces are coherently oriented and the solid models are completely determined. Using the de Boor's algorithm and the surface orientation a model with exclusive triangular faces coherently oriented is created, and then the blending surface is automatically created. The refined mesh for the models are created and fluid-structure analysis is performed for the simulation-optimization. *Copyright ©2014 IFAC.*

Keywords: B-Rep, IGES, solid models, hydro-power generator, B-Spline surfaces, CAD, CAE.

1. INTRODUCTION

Cavitation and cracks affect power generation turbines from Ilha Solteira hydroelectric plant, among many others. During cavitation repair in Ilha Solteira, $3,000$ kg of weld material is deposited. The turbine geometry quality is manually controlled using 2D templates. Sobrinho et al. (2009) reconstructed the turbine blades geometry from Ilha Solteira hydroelectric plant, and used the CAD model to perform static and fatigue numerical analysis. Of particular maintenance importance is the influence of the smoothing surface between the blade and the crown. The modifications in the geometry are manually executed. This work investigates the development of an automatic methodology connecting CAD and CAE systems with a geometric optimization purpose. It is desired to determine the radius of a blending surface that minimizes the turbine blade stress determined by a fluid-structure analysis.

Simulation-based optimization is an area where techniques from optimization and simulation analysis fields are integrated (Andradóttit, 1998; Ólafsson and Kim, 2002; Gosavi, 2003). The objective function is associated with a measurement obtained from numerical simulations and the project parameters are modified during the optimization process. Therefore, CAD/CAE/CAM feature based projects, in which regions of interest with functional significance are procedurally represented, are appropriated models for simulation-based optimization as it provides suitable parameters to modify the geometry (Shah et al., 1994).

An undesired effect of this project automation approach is the difficulty to add new features to the original CAD model. Even though most CAD systems provide programming support, the inclusion a new features is not a trivial task. A possible solution is the employment of an intermediate system which imports the CAD model, interprets the geometry, processes the modification then exports the new model for the CAE system.

The IGES (US PRO, 1996) is a popular vendor neutral file format and it is supported by most CAD/CAE/CAM systems. There are more advanced formats such as STEP (Stroud, 2006), though they are not easy to interpret and process. Accordingly, the IGES format was cho-

---

[1] e-mail: emilianogcastro@gmail.com
[2] e-mail: mtsuzuki@usp.br.
[3] e-mail: ecnsilva@usp.br.
[4] e-mail: sandro.vatanabe@gmail.com.
[5] e-mail: andre.kubagawa@gmail.com.
[6] e-mail: thiago@usp.br
[7] e-mail: giulliano.gallo@cesp.com.br
[8] e-mail: marco.marques@cesp.com.br
[9] e-mail: hamilton.tiba@cesp.com.br

sen as the data carrier between the intermediary module (Blade Transformer) and the CAD/CAE systems (BladeGen and ANSYS). The CAD system is responsible for the project (geometric model) and the CAE module for the numerical simulation (evaluation of the objective function).

The IGES format is capable of properly representing faceted B-Rep solid models. However, when the B-Rep model contains surfaces, new difficulties arise. Companies which adopts CAD systems with surface support usually requires that the same CAD system is used for all steps of the project chain. With such extreme strategy, it is guaranteed that all parameters associates with the procedural models are correctly interpreted. In this conservative approach, new features cannot be easily included by existing CAD system functions (Shah et al., 1994).

Main problems of IGES representation of B-Rep solid models with surfaces involve the correct mapping of curve and surface geometry and the topology represented by the union of primitive elements: vertex, edge and face (Mantyla, 1988). It is necessary for the surface contour to be connected (geometric adjacent), defining a totally closed volume. Surfaces must be coherently orientated with its normals pointing outwards (some CAD systems adopt an inverted convention).

In this work, the proposed approach is applied to the maintenance of hydroelectrical plant turbines. Turbine project is designed using the BladeGen CAD system and the fluid-structure simulation is performed in the ANSYS (Campbell and Paterson, 2011) software. For the fluid-structure analysis, turbine project modifications must affect the fluid volume. This adds an extra degree of complexity, as both geometries must share the same modifications. The new design feature is a blending surface between the blade and the crown that can reduce the turbine stress.

This paper describes the development of an intermediary module called Blade Transformer which allows for CAD/CAE automation with new design feature inclusion. Model files are transported using the IGES file format which provides an incomplete representation of solid models with surfaces. Thus, additional processing is necessary to ensure that the geometry and topology are consistently recreated. This text is structured as follows: section 2 briefly explains the motivations and the proposed approach flowchart; section 3 describes the Blade Transformer; section 4 shows the results obtained and section 5 the conclusions and future work.

2. MOTIVATION

The flow of the proposed approach is shown in Fig. 1. The turbine CAD model is created by the ANSYS BladeGen, an advanced, interactive blade design software that has application related terminology, also called as design features. Two models are required for fluid-structure analysis: fluid volume and turbine models. Both models are exported in IGES format with the B-Spline surfaces. The intermediary module, called Blade Transformer, reads the IGES file and interprets the geometry. A new blending surface is created connecting the crown and the blade. Two radius can be specified as parameters, the medium and boundary radius, and a smooth surface with non constant radius is created. Both models are modified accordingly and exported to IGES format. The Hypermesh reads the IGES file and creates a mesh appropriate for fluid-structure analysis. The result will be input to a meta-heuristic algorithm such as simulated annealing (Kirkpatrick et al., 1983; Martins and Tsuzuki, 2010),

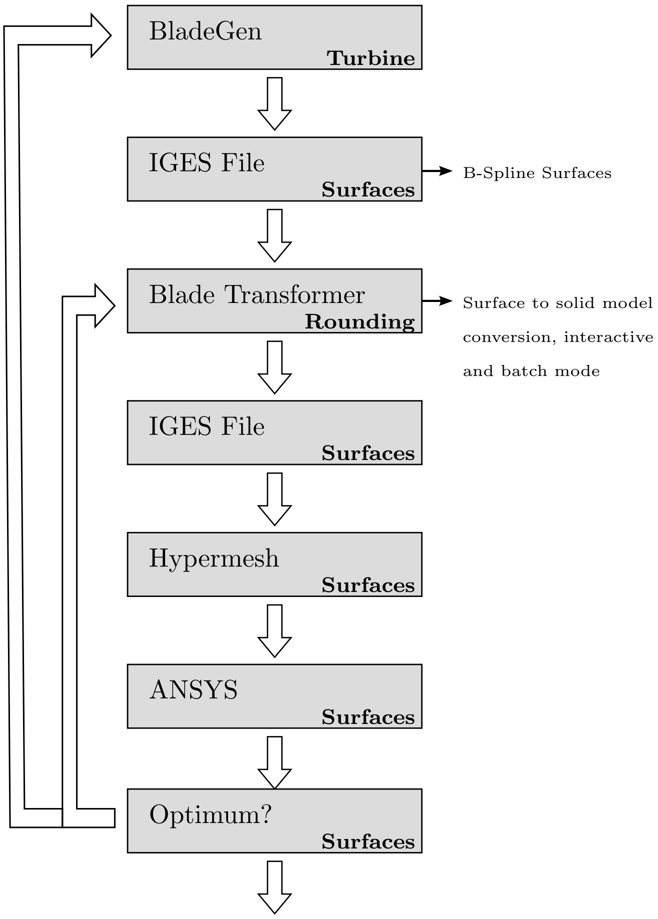

Fig. 1. Proposed approach flowchart. The Blade Transformer is the developed module to integrate and include a new geometrical feature to the framework.

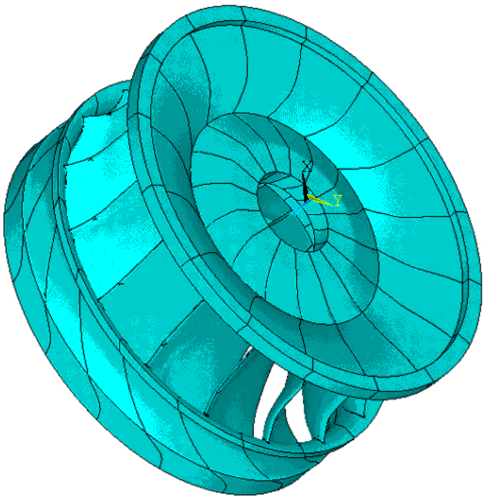

Fig. 2. Final CAD model of a Francis turbine.

```
IGES Export from  ANSYS BladeGen v12.0.035                              S      1
Created from D:\Users\BladeEditor\Dev_Environment\Francis\Francis_001_fiS      2
les\dp0\BG\TS\BG.bgd                                                    S      3
1H,,1H;,11HBlade Model,                                                 G      1
60HD:\Users\BladeEditor\Dev_Environment\Francis\Francis_001.igs,        G      2
24HANSYS BladeGen v12.0.035,24HANSYS BladeGen v12.0.035,32,38,6,308,15, G      3
16HIGES Blade Model,1.D0,1,2HIN,1,1.D-002,15H2013028.155046,0.D0,00.D0, G      4
6HSandro,,11,0,,;                                                       G      5
     128       1       0       0       0       0       0       000000000D      1
     128       0       0      24       0       0       0  B_SURF       1D      2
     126      25       0       0       0       0       0       000000000D      3
     126       0       0       7       0       0       0B_SPLINE       1D      4
     126      32       0       0       0       0       0       000000000D      5
     126       0       0       7       0       0       0B_SPLINE       2D      6
     126      39       0       0       0       0       0       000000000D      7
     126       0       0       6       0       0       0B_SPLINE       3D      8
     126      45       0       0       0       0       0       000000000D      9
     126       0       0       6       0       0       0B_SPLINE       4D     10
     116      51       0       0       0       0       0       000000000D     11
     116       0       0       1       0       0       0   POINT       1D     12
     116      52       0       0       0       0       0       000000000D     13
     116       0       0       1       0       0       0   POINT       2D     14
```

Fig. 3. Part of an IGES file with a B-Spline surface represented.

genetic algorithms (Thengade and Dondal, 2012), particle swarms (Castro and Tsuzuki, 2008), which continuously modifies some predefined parameters until convergence or a stopping criteria is met. The meta-heuristic algorithm will control BladeGen and Blade Transformer parameters. The meta-heuristic is not included in this research and it is left as future work.

The proposed approach is intented to hydroelectrical plant turbines maintenance (see Fig. 2). Due to the rotational symmetry of the model, only one blade is exported to the IGES format. It contains the blade geometry and its corresponding fluid volume. Thus, it is possible to perform fluid-structure analysis using the finite element method.

The focus of this work is the Blade Transformer module, which is responsible for the interpretation of the IGES file and the reconstruction of the model with new design features. One of the main obstacles is the presence of two models, the blade and the fluid volume, which shares a common surface. The exported model contains only information about the surfaces, however no adjacency data, which is very important to ensure that the model defines a correct and closed solid model and to determine the correct surface orientation. Adjacency is obtained by comparing the surface contour curves. The comparison can be performed numerically using the control points of the curves.

2.1 Data acquisition

The development of the IGES file format started in 1979 by a group of users and developers of CAD systems such as Boeing, General Electric and Xerox, among others, with the purpose of guaranteeing interoperability between their systems (US PRO, 1996).

An IGES file is coded using ASCII text with a column of 80 characters wide and it is divided in five sections: *Start*, *Global*, *Directory Entry*, *Parameter Data* e *Terminate* (sections are indicated on column 73 of each line by the letters S, G, D, P e T, respectively). B-Spline surfaces are represented by a data structure identified by code 128. The complete documentation can be found online in (US PRO, 1996). An example of an IGES file is shown in Fig. 3.

Based on the IGES specification, a parser which interprets the files and retrieves the B-Rep reconstruction parameters was developed. For each B-Spline surface, the number of control points, the order and the knot vector in both directions (u and v), weights and control points are collected.

2.2 Surface reconstruction

De Boor's algorithm is proposed to create models with correctly orientated triangular faces. It is an adaptation for B-Spline and NURBS surfaces of De Casteljau's algorithm, which is designed for Bézier surfaces (Piegl and Tiller, 1997; Patrikalis and Maekawa, 2001).

De Boor's algorithm explores the B-Splines property which guarantees that, if a knot u is repeatedly inserted until its multiplicity is equal to the B-Spline order, the last control point generated by an insertion is the curve point corresponding to u. Thus, it is possible to define an automatic mechanism for knot insertion.

In this work, the knots of interest were obtained through the parametric discretization of the plane unit square surface. Such parametrization scheme was adopted to guarantee the required flexibility for surface construction with different resolutions.

The parametrization scheme is basically a grid dividing each dimension of the base surface (square) in 100 segments with equal length, generating 101 points (including endpoints), thus defining 10,201 (101×101) control points. For each control point, its corresponding point in the surface $\mathbf{p}(u, v)$ is obtained using the De Boor's algorithm.

The B-Spline surface equation can be written as:

$$\mathbf{p}(u, v) = \sum_{i=0}^{m} N_{i,p}(u) \left(\sum_{j=0}^{n} N_{j,q}(v) \mathbf{p}_{i,j} \right). \tag{1}$$

where $\mathbf{p}_{i,j}$ are control point and $N_{i,p}(u)$ and $N_{j,q}(v)$ are base functions defined as

$$N_{i,1}(u) = \begin{cases} 1, \text{ for } u_i \leq u < u_{i+1}, \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

for $p = 1$, and

$$\begin{aligned} N_{i,p}(u) = & \frac{u - u_i}{u_{i+p-1} - u_i} N_{i,p-1}(u) \\ & + \frac{u_{i+p} - u}{u_{i+p} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{3}$$

For a given index i, the expression inside the parenthesis in (1) is a B-spline curve defined by the control points with index i. For an easier understanding, $\mathbf{q}_i(v)$ is defined as

$$\mathbf{q}_i(v) = \sum_{j=0}^{n} N_{j,q}(v) \mathbf{p}_{i,j}. \tag{4}$$

Therefore, $\mathbf{q}_i(v)$ is a point corresponding to v in the B-spline curve defined by the control point from line i. If v is inside the knot interval $[v_d, v_{d+1})$, then only $q + 1$ control points in line i are involved in the computation of $\mathbf{q}_i(v)$, where q is the order of $N_{j,q}(v)$. These control points are $\mathbf{p}_{i,d}, \mathbf{p}_{i,d-1}, \ldots, \mathbf{p}_{i,d-q}$, if v is different from v_d. Otherwise, if v is equal to v_d, which is a knot of multiplicity t, then the control points involved are $\mathbf{p}_{i,d-t}, \mathbf{p}_{i,d-t-1}, \ldots, \mathbf{p}_{i,d-q}$.

Thus, using the control point from the column $d - q$ till column $d - t$, where t is zero if v is not a knot, it is

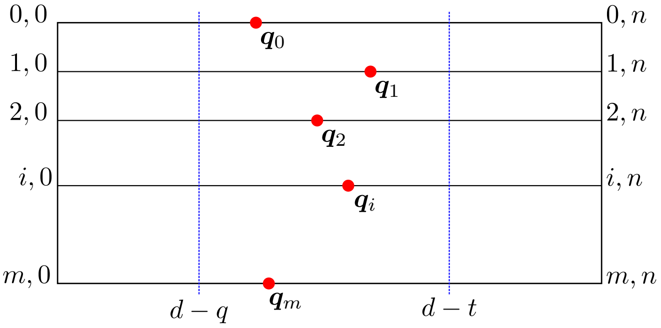

Fig. 4. De Boor's algorithm example.

possible to apply de Boor's algorithm for each line and obtain $m+1$ new points $\mathbf{q}_0(v), \mathbf{q}_1(v), \ldots, \mathbf{q}_m(v)$. These points are represented in Fig. 4. Substituting these new points in (1), one has

$$\mathbf{p}(u,v) = \sum_{i=0}^{m} N_{i,p}(u)\mathbf{q}_i(v). \tag{5}$$

Hence $\mathbf{p}(u,v)$ is a point in the B-Spline curve defined by $\mathbf{q}_0(v), \mathbf{q}_1(v), \ldots, \mathbf{q}_m(v)$. Then, in order to find $\mathbf{p}(u,v)$, it is necessary to find the point in such curve which corresponding to u, which can be accomplished by applying de Boor's algorithm.

3. THE BLADE TRANSFORMER

The blade transformer is the intermediary system that reads the IGES model, determines the correct surface orientation, interprets the geometry, creates a blending surface between the correct two surfaces and exports the IGES model. The blending surface is not included in the Blade Gen set of commands.

After the IGES file with B-Splines is loaded, the adjacency relationship of the surfaces is determined. This operation is performed in three steps:

1. First, the surface boundary curves are determined and enumerated.
2. The surface boundary curves are compared against each other.
3. The boundary curves orientation are determined.

The first module just enumerates the boundary curves. Then the curves are compared against each other, initially only the initial and final control points are considered. Contour curves which shares common endpoints defines a match and the other control points of each boundary curve is compared. Fig 5(a) shows a set of four surfaces with their adjacency determined. Each edge is adjacent to at least two surfaces. There may be more than two surface as there are two objects (the blade and the fluid volume).

Once the adjacency relationship is determined, the boundary surface orientation must be determined. Initially, an orientation is set for the first surface and it is considered correct. Orientation of adjacent surfaces can be defined considering that each curve is shared, with opposed directions, by two neighboring surfaces. This process is repeated for every boundary curve which is part of at least two adjacent surfaces. When there are no more boundary curves to process, a non orientated surface is chosen to start the next cycle of the algorithm. Thus, two objects can be separated by grouping coherent surfaces. This algorithm is similar to the open boundary marching cubes proposed by Tsuzuki et al. (2007).

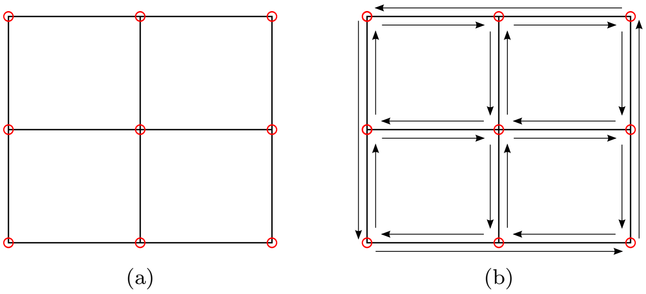

Fig. 5. (a) Surface adjacency information is already determined. Red circles highlights shared endpoints. (b) Surfaces with correct orientations.

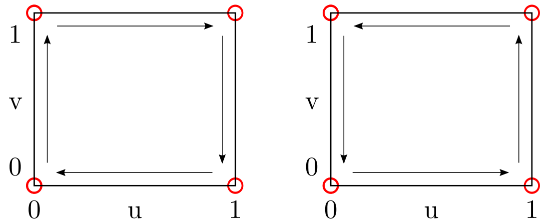

Fig. 6. Two possible orientations configurations for the surface parametric space.

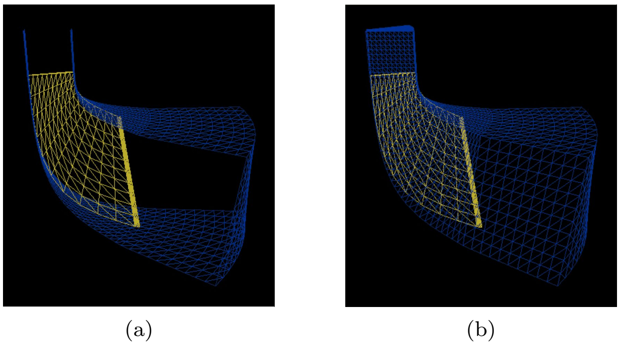

Fig. 7. (a) Wireframe view of the BladeGen exported model with incorrect normals. Yellow if the the blade and blue is the fluid volume. (b) Model with corrected normals.

The orientation of the surfaces is checked by comparing with the object central point. The central point is internal to the object. If an incoherence is found, the orientations are inverted. Fig. 5(b) shows a set of surfaces with their orientations coherently defined.

After all orientations are defined, it is possible to associate this orientation with the parametric space. Fig. 6 shows the two possible orientations. Using de Boor's algorithm, coherently oriented triangles are created.

Fig. 7 shows an example of the normal correction procedure applied to a BladeGen exported model. Fig. 7(a) shows a model with some invisible surfaces which have the normal pointing inward. In Fig. 7(b), the normal correction routine was applied and the correct result, with all surface normals pointing outwards, is shown.

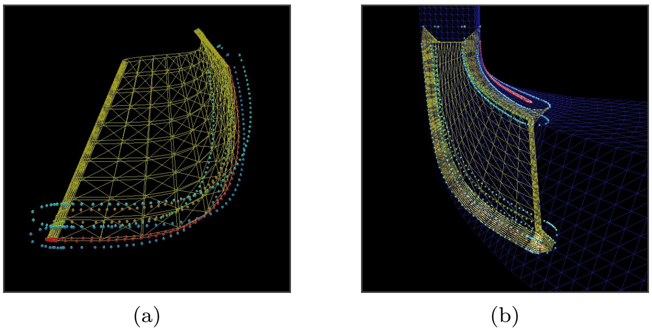
(a) (b)

Fig. 8. (a) Center and contact curves shown with blue balls. (b) Another view with surfaces already trimmed.

3.1 Blending Surface Construction

The blending surface must combine one surface from each object: the crown surface from the fluid volume and the blade surface from the blade. A surface blending algorithm with constant/variable radii was developed. The blending algorithm was based on the rolling ball blending between two surfaces at a given contact curve (Huang and Zhu, 2000). A ball of given radius rolls while the contact with the two selected surfaces is kept. The algorithm has four main steps:

1. Construct the offset surfaces from the given surfaces.
2. Compute the intersection between the offset surfaces in order to determine the center and contact curves.
3. Construct a blending surface by sweeping the section curves along the center and contact curves.
4. Trim the original surfaces, such that it has a new boundary: the contact curve.

Consider that surfaces $F(u, v)$ and $G(s, t)$ have respective normals $n_f(u, v)$ and $n_g(s, t)$ and a offset d. The offset surfaces $F_d(u, v)$ and $G_d(s, t)$ are given by

$$F_d(u, v) = F(u, v) + d \cdot \boldsymbol{n}_f(u, v)$$
$$G_d(s, t) = G(s, t) + d \cdot \boldsymbol{n}_g(s, t).$$

An initial point is determined and a marching process is performed to create the contact and center curves. The vector product between the normals at the offset surface $n_{fd}(u, v) \times n_{gd}(s, t)$ gives the marching direction for the center curve. The tangent vector at the surfaces with the $\partial F(u, v)/\partial u$, $\partial F(u, v)/\partial v$, $\partial G(s, t)/\partial s$ and $\partial G(s, t)/\partial t$ are employed to determine the contact curves using the offset surface normals. Fig. 8(a) shows the center and contact curves. The crown surface is not shown to for better visualization. Fig. 8(b) shows the surfaces after trimming. Fig. 9 shows the determined blending surfaces. Both objects are updated accordingly: the fluid volume and the blade.

4. RESULTS

The Blade Transformer has some visualization features, Fig. 10 shows an example which indicates the blade surface using the yellow color and the fluid volume in blue. Fig. 11 shows the blending surface created by the Blade Transformer. Fluid-structure coupling is performed by sending the fluid stress information from the flow to the structure and structural displacements from the structure to the flow. The fluid-structure analysis is performed by MatLab, which controls ANSYS CFX and ANSYS Mechanical by command line. Fig. 12 shows the fluid speed vector flow determined by ANSYS CFX. Fig. 13 shows the mechanical stresses determined by ANSYS Mechanical. Fig. 14 shows the diplacement modulus determined by ANSYS Mechanical.

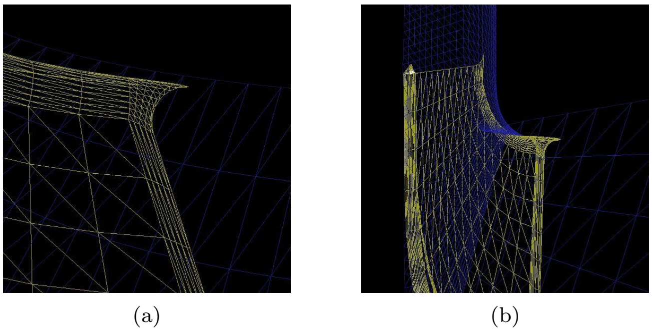
(a) (b)

Fig. 9. (a) Blending surface created between the crown and the blade. (b) Different view of the blending surface.

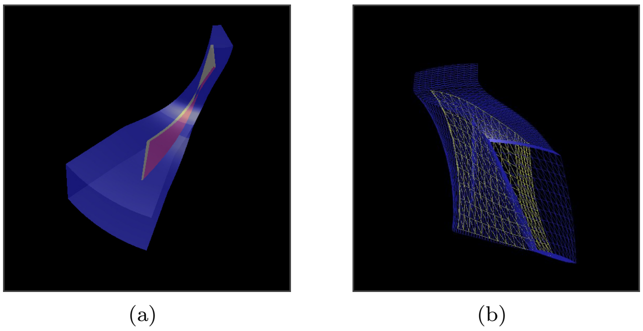
(a) (b)

Fig. 10. (a) Blade side surfaces in red and fluid volume transparent. (b) Francis turbine example.

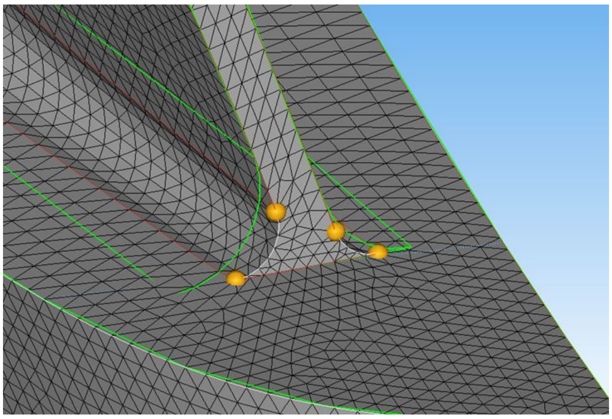

Fig. 11. Hypermesh output showing the meshed blending surface.

The flowchart shown in Fig. 1 is not completely implemented. It remains to include the Hypermesh and the Blade Gen in the automatic process. Additionally, the meta-heuristic for the optimization must be defined as well.

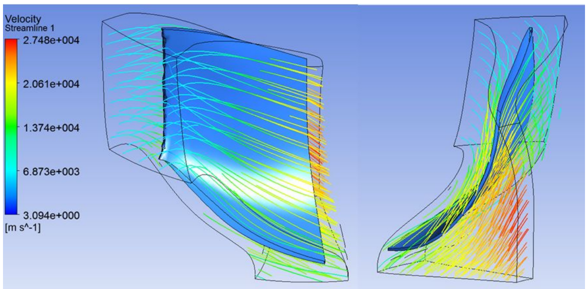

Fig. 12. Fluid speed vector flow.

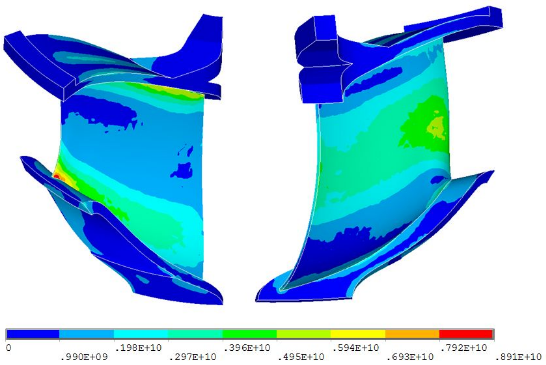

Fig. 13. Von Mises mechanical stresses. Model with the blending surface.

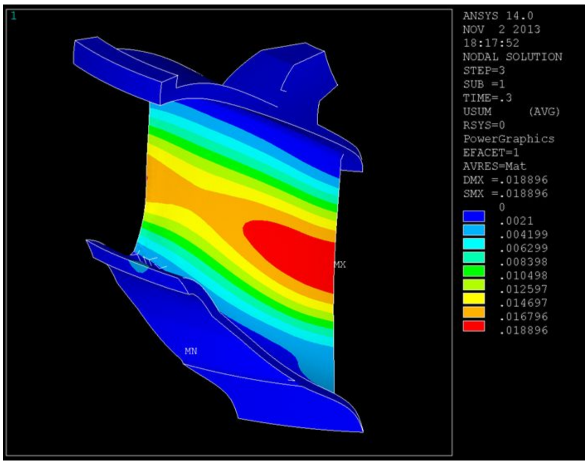

Fig. 14. Diplacement modulus.

5. CONCLUSIONS

A methodology for the simulation-based optimization CAD/CAE, in which new features are included was proposed in this work. The geometry design is exported using an IGES file where curved solid models are contained (fluid volume and blade). With the data from the file it is possible to recreate the models, however multiple interpretations exist. The proposed intermediary system determines the valid interpretations, verify the model final consistency, creates the blending surface and updates both models, which were then exported to Hypermesh and a fluid-structure analysis was performed.

ACKNOWLEDGEMENTS

E. G. Castro is supported by FUNDESPA. M. S. G. Tsuzuki, T. C. Martins and E. C. N. Silva are partially supported by CNPq (Grants 309.570/2010–7, 306.415/2012–7 and 303.689/2009-9, respectively). A. K. Sato is supported by FAPESP (Grant 2010/19646-0). This research is supported by the ANEEL PD-0061-0033/2011 project.

REFERENCES

Andradóttit, S. (1998). A review of simulation optimization techniques. In *Proc 1998 Winter Simulation Conference*, 151–158.

Campbell, R. and Paterson, E. (2011). Fluid–structure interaction analysis of flexible turbomachinery. *J Fluid Struct*, 27(8), 1376 – 1391.

Castro, E.G. and Tsuzuki, M.S.G. (2008). Swarm intelligence applied in synthesis of hunting strategies in a three-dimensional environment. *Expert Syst Appl*, 34, 1995–2003.

Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning.* Kluwer Academic Publishers.

Huang, L. and Zhu, X. (2000). Construction of blending surfaces. Technical Report HZ-TMSurf-Huang04, Beijing University of Aeronautics & Astronautics.

Kirkpatrick, S., Gellat, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.

Mantyla, M. (1988). *Introduction to Solid Modeling.* W. H. Freeman & Co., New York, NY, USA.

Martins, T.C. and Tsuzuki, M.S.G. (2010). Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. *Expert Syst Appl*, 37, 1955–1972.

Ólafsson, S. and Kim, J. (2002). Simulation optimization. In *Proc 2002 Winter Simulation Conference*, 79–84.

Patrikalis, N.M. and Maekawa, T. (2001). *Shape Interrogation for Computer Aided Design and Manufacturing.* Springer-Verlag.

Piegl, L. and Tiller, W. (1997). *The NURBS book.* Springer-Verlag.

Shah, J., Mantyla, M., and Nau, D. (1994). *Advances in Feature-Based Manufacturing.* Elsevier Science Publishers.

Sobrinho, E., Sanomya, R., Ueda, R., Tiba, H., Tsuzuki, M.S.G., Adamowski, J.C., Silva, E.C.N., carnobari, R.C., and Buiochi, F. (2009). Development of a methodology for evaluation of a structural damage in turbine blades from hydropower generators. In *Proc COBEM 2009.* Gramado, Brazil.

Stroud, I. (2006). *Boundary Representation Modelling Techniques.* Springer Verlag.

Thengade, A. and Dondal, R. (2012). Article: Genetic algorithm - survey paper. *IJCA Proc National Conference on Recent Trends in Computing*, NCRTC(5), 25–29.

Tsuzuki, M.S.G., Takase, F.K., Garcia, M.A.A.S., and Martins, T.C. (2007). Converting CSG models into meshed B-Rep models using euler operators and propagation based marching cubes. *J Braz Soc Mech Sci*, 29, 337 – 344.

US PRO (1996). Initial Graphics Exchange Specification - IGES 5.3.