**ON THE VISUALIZATION OF DOMAIN RESULTS
IN TWO-DIMENSIONAL BOUNDARY ELEMENT ANALYSIS****André Maués Brabo Pereira****Marcos Aurélio Marques Noronha**

Depto. de Engenharia de Estruturas e Fundações – Universidade de São Paulo

Escola Politécnica – 05508-900 – São Paulo - SP, Brazil

{maues.pereira , marcos.noronha}@poli.usp.br

Remo Magalhães de Souza

Depto. de Construção Civil – Universidade Federal do Pará

Centro Tecnológico – 66075-970 – Belém - Pa, Brazil

remo@ufpa.br

***Abstract.** Most postprocessors for BE analysis uses an auxiliary domain mesh to display domain results, working against the profitable modeling process of pure BEMs. Recently, the two first authors of this paper presented a new visualization algorithm based on the direct and automatic identification of isocurves. This algorithm does not require any domain discretization, preserving the basic properties of the BEM. One of the critical aspects of the visualization of domain results in BE analysis is the computational effort required for the evaluation of results in interior points. For the conventional formulation of the BEM, this last procedure demands a relatively high computational effort. On the other hand, the hybrid formulation of the BEM represents a powerful alternative to improve the performance of those evaluations, since the domain results are obtained without evaluation of integrals. This paper discusses some aspects related to the visualization of domain results in BE analysis. In addition, it also compares the performance of the conventional and hybrid formulations of the BEM using different visualization procedures. Some examples show the results of the present study. Finally, the paper discusses on new applications of the visualization algorithms, specially for nonlinear analyses with BEM.*

Keywords: Boundary Element Methods, Scientific Visualization, Contour Plot.

1. INTRODUCTION

In the recent years, due to the increasing number of research and technological advances, Computer Graphics has been expanding rapidly into other fields and areas of knowledge (Foley, 1997) (Farin, 2002). The application of Computer Graphics in engineering and in other scientific and technological areas has been fundamental to the development of new researches on the visualization of scientific data. Nowadays, the computational simulation of continuum mechanics problems demands an extensive use of graphical applications.

Among other numerical methods, the Finite Element Method (FEM) and the Boundary Element Method (BEM) are the most significant techniques for continuum mechanics problems. One of the main advantages of the BEM over the FEM is the low effort of its modeling process. Pure BEMs should represent models only with boundary discretization. However, to accomplish the visualization of domain results, most BEM implementations uses an auxiliary domain discretization. This is a natural approach for the FEM, which uses a mesh over the domain to represent the model. However, this procedure is inadequate for BEMs, since no domain discretization should be necessary. In addition, this procedure can neglect critical results that occur in regions with high gradients, then its efficiency is questionable.

The authors have not find in the literature any specific technique for the visualization in BE analysis. Hence, they developed a new procedure without domain discretization based on the direct and automatic identification of isocurves and isobands. Basically, it consists in a combination of iterative and predictive procedures and spline adjustment techniques.

This paper also discusses some aspects related to the visualization of domain results in BE analysis, comparing the conventional algorithms with auxiliary domain discretization to the new visualization algorithm proposed. It also compares the performance of the conventional BEM to the hybrid formulation of the BEM, an alternative technique which does not require integral calculations to obtain results at interior points.

2. BOUNDARY ELEMENT FORMULATIONS

This section presents a review of the main features of the conventional and hybrid formulations of the BEM, for analysis of planar, isotropic, static and linear elasticity (Brebbia, 1989) (Beer, 2001) (Dumont, 1989) (Souza, 1992). The following developments consider the particular case of an elastic body having a domain Ω enclosed by a boundary Γ which is subjected to tractions $\bar{\mathbf{t}}$ at Γ_t and displacements $\bar{\mathbf{u}}$ at Γ_u , as depicted in Fig. 1a.

In both BE formulations herein presented, the discretization of the boundary Γ uses nodes and elements (Fig. 1b). Only nodal parameters are necessary to the primary representation of the variables involved in the analysis. This representation can be extended to the whole boundary using interpolations over each element as follows:

$$\mathbf{u} = \Phi_j \mathbf{u}^j \quad \text{and} \quad \mathbf{t} = \Phi_j \mathbf{t}^j \quad , \quad (1)$$

where Φ_j are the interpolation functions, \mathbf{u}^j and \mathbf{t}^j are the values of the displacements and tractions at the element nodes. Basically, BE analysis has two steps. The first one consists on finding the results at the boundary. After this, results at any point at the domain can be inquired. Another important feature of BEMs is the use of a special solution of the governing equation of the problem called Fundamental Solution, which is partly responsible for the elimination of domain integrals. For stress analysis, this solution is associated to the problem of a concentrate load applied in a point P of an infinite elastic medium, as Fig. 1c depicts.

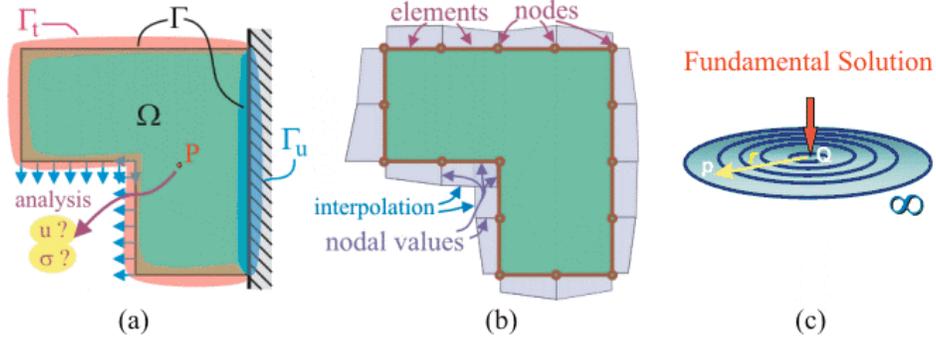


Figure 1 - (a) Structural Model; (b) BE-Discretization; (c) Fundamental Solution.

2.1 Conventional Boundary Element Method (CBEM)

The conventional BEM formulation (CBEM) is based on a relation called Somigliana's Identity, which is an integral equation involving the Fundamental Solutions. This relation is commonly derived from weighted residuals techniques, Betti's reciprocal theorem or Green's third identity. In the case of linear problems, this relation (Brebbia & Domingues, 1989) is

$$u_l^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma, \quad (2)$$

where the Fundamental Solutions u_{lk}^* and p_{lk}^* represent the displacement and traction, respectively, at any point in the k direction when a unit load is applied at point i in the l direction. These Fundamental Solutions are

$$u_{lk}^* = \frac{1}{8\pi\mu(1-\nu)} \left[(3-4\nu) \ln\left(\frac{1}{r}\right) \delta_{lk} + r_{,l} r_{,k} \right] \quad (3)$$

$$\text{and } p_{lk}^* = -\frac{1}{4\pi(1-\nu)r} \left\{ \frac{\partial r}{\partial n} [(1-2\nu)\delta_{lk} + 2r_{,l} r_{,k}] + (1-2\nu)(n_l r_{,k} - n_k r_{,l}) \right\}. \quad (4)$$

Results at the boundary. In order to obtain the nodal values of the variables, the CBEM algorithm evaluates the Somigliana's Identity for each node i of the discretization. With the interpolation defined in Eq. (1), this procedure gives

$$\mathbf{c}^i \mathbf{u}^i + \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{p}^* \Phi d\Gamma \right\} \mathbf{u}^j = \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{u}^* \Phi d\Gamma \right\} \mathbf{p}^j, \quad (5)$$

where the coefficient \mathbf{c}^i depends on the boundary geometry in the vicinity of the node i and on the Poisson's ratio. NE is the number of boundary elements.

The application of Eq. (5) for all boundary nodes, gives the following equation system:

$$\mathbf{H} \mathbf{u} = \mathbf{G} \mathbf{t}, \quad (6)$$

where \mathbf{H} and \mathbf{G} are full and nonsymmetrical matrices, \mathbf{u} and \mathbf{t} the vectors of displacements and tractions, respectively. The solution of Eq. (6) gives the results of \mathbf{u} and \mathbf{t} which can be extended to the whole boundary using the interpolation functions given by Eq. (1).

Results at internal points. After solving Eq. (6), the displacements for any point in the domain Ω can be evaluated using the Somigliana's Identity, Eq. (2). In a similar manner, results for the stress tensor can be obtained at internal points by

$$\sigma_{ij} = \int_{\Gamma} D_{kij} t_k d\Gamma - \int_{\Gamma} S_{kij} u_k d\Gamma, \quad (7)$$

where

$$D_{kij} = \frac{1}{4\pi(1-\nu)r} \left\{ (1-2\nu)(\delta_{ki}r_{,j} + \delta_{kj}r_{,i} - \delta_{ij}r_{,k}) + 2r_{,i}r_{,j}r_{,k} \right\} \quad (8)$$

$$S_{kij} = \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2 \frac{\partial r}{\partial n} \left[(1-2\nu)\delta_{ij}r_{,k} + \nu(\delta_{ik}r_{,j} + \delta_{jk}r_{,i}) - 4r_{,i}r_{,j}r_{,k} \right] + \right. \\ \left. + 2\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k}) + (1-2\nu)(2n_k r_{,i}r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - (1-4\nu)n_k \delta_{ij} \right\}. \quad (9)$$

Extended results. In order to obtain the isocurves of the stress field in Ω , the visualization algorithm proposed in this paper requires the stress tensor derivatives. These derivatives can be evaluated using the following expression:

$$\sigma_{ij,m} = \int_{\Gamma} D_{kij,m} t_k d\Gamma - \int_{\Gamma} S_{kij,m} u_k d\Gamma, \quad (10)$$

where

$$D_{kij,m} = \frac{1}{r} D_{kij} r_{,m} + \frac{1}{4\pi(1-\nu)r} \left\{ (1-2\nu)(\delta_{ki}r_{,jm} + \delta_{kj}r_{,im} - \delta_{ij}r_{,km}) + 2(r_{,im}r_{,j}r_{,k} + r_{,jm}r_{,i}r_{,k} + r_{,km}r_{,i}r_{,j}) \right\} \quad (11)$$

and

$$S_{kij,m} = \frac{1}{r} 2S_{kij} r_{,m} + \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2r_{,m} \left[(1-2\nu)\delta_{ij}r_{,k} + \nu(\delta_{ik}r_{,j} + \delta_{jk}r_{,i}) - 4r_{,i}r_{,j}r_{,k} \right] + \right. \\ \left. 2r_{,m} \left[(1-2\nu)\delta_{ij}r_{,km} + \nu(\delta_{ik}r_{,jm} + \delta_{jk}r_{,im}) - 4(r_{,im}r_{,j}r_{,k} + r_{,jm}r_{,i}r_{,k} + r_{,km}r_{,i}r_{,j}) \right] \right. \\ \left. + 2\nu[n_i(r_{,jm}r_{,k} + r_{,km}r_{,j}) + n_j(r_{,im}r_{,k} + r_{,km}r_{,i})] + (1-2\nu)2n_k(r_{,im}r_{,j} + r_{,jm}r_{,i}) \right\}. \quad (12)$$

The evaluation of results in points at the boundary requires calculation of singular integrals. In addition, calculation for points close to the boundary involves quasi-singular integrals. Both evaluations require a special integration scheme (Dumont & Noronha, 1998). To avoid those effects, this work considers visualization at regions shifted from the boundary.

2.2 Hybrid Boundary Element Method (HBEM)

The hybrid BEM formulation (HBEM) is based on the Hellinger-Reissner variational principle (Dumont, 1989). The Hellinger-Reissner potential Π_R can be expressed as

$$-\Pi_R = \int_{\Omega} [U_0^{*c}(\sigma_{ij}^*) + D_{ij,j}^* u_i] d\Omega - \int_{\Gamma} u_i \sigma_{ij}^* \eta_j d\Gamma + \int_{\Gamma^{\sigma}} \bar{t}_i u_i d\Gamma, \quad (13)$$

where $U_0^{*c}(\sigma_{ij}^*)$ is the complementary strain energy, and σ_{ij}^* is the stress field at the domain, which is given by a linear combination of fundamental solutions as

$$\sigma_{ij}^* = D_{kij} p_k^*, \quad (14)$$

where \mathbf{p}^* is a vector of singular forces.

In a similar manner, the derivatives of the stress field at the domain can be expressed as

$$\sigma_{ij,m}^* = D_{kij,m} p_k^* , \quad (15)$$

where $D_{kij,m}$ is the derivative of the fundamental solution for stresses, Eq. (11).

Results at the boundary. After discretization and application of the principle of stationary potential energy to the Hellinger-Reissner potential, the problem can be expressed as

$$\begin{bmatrix} -\mathbf{F} & \mathbf{H} \\ \mathbf{H}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{p}^* \\ \mathbf{d} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{p} \end{Bmatrix} , \quad (16)$$

where \mathbf{d} is the nodal displacement vector, \mathbf{F} is a symmetric flexibility matrix, \mathbf{H} is the same matrix of the CBEM and \mathbf{p} is the vector of nodal forces equivalent to the boundary tractions.

Solving the first equation in the system of Eq. (16) for \mathbf{p}^* results in

$$\mathbf{p}^* = \mathbf{F}^{-1} \mathbf{H} \mathbf{d} . \quad (17)$$

Substitution of \mathbf{p}^* in the second equation of the system Eq. (16) gives

$$\mathbf{K} \mathbf{d} = \mathbf{p} , \quad (18)$$

where \mathbf{K} is a symmetric stiffness matrix, given by $\mathbf{K} = \mathbf{H}^T \mathbf{F}^{-1} \mathbf{H}$, where \mathbf{F}^{-1} is a generalized inverse of \mathbf{F} . For more details, the reader is referred to (Dumont,1989). Solution of Eq. (18) gives the results of \mathbf{d} and \mathbf{p} at the discretization nodes.

Results at internal points. After solving Eq. (18) and substitution of results in Eq. (17), the stress and the displacements for any point in the domain Ω should be defined through the nodal forces \mathbf{p}^* , except on the points of application of these forces. Then, the stress field at Ω and its derivatives can be obtained directly from Eq. (14) and Eq. (15).

Since no integration is necessary, the evaluation of internal results with the HBEM becomes simpler and faster than any other method. On the other hand, there are also some drawbacks, as the behavior of the results close to the nodes of the discretization, which invariably tends to infinity. Another drawback is the evaluation and inversion of matrix \mathbf{F} , which demand a high computational effort.

3. VISUALIZATION TECHNIQUES

The isocurve plot technique (Fig. 2a) is one important tool that analysts commonly use to visualize data for 2D engineering problems. For a scalar function $f(\mathbf{x})$ where $\mathbf{x} \in R^2$, an isocurve segment is the set of points $\bar{\mathbf{x}}$ which satisfy $f(\bar{\mathbf{x}}) = f_i$ at the interval $[\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_f]$, where f_i is the level or height of the isocurve segment with $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_f$ as limiting points (Figure 2b). For a given point of an isocurve the normal and tangent unit vectors are given by

$$\mathbf{n} = \left\{ \frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \right\}^T / \sqrt{\left(\frac{\partial f}{\partial x_1} \right)^2 + \left(\frac{\partial f}{\partial x_2} \right)^2} \quad \text{and} \quad \mathbf{t} = \left\{ \frac{\partial f}{\partial x_2} \quad -\frac{\partial f}{\partial x_1} \right\}^T / \sqrt{\left(\frac{\partial f}{\partial x_1} \right)^2 + \left(\frac{\partial f}{\partial x_2} \right)^2} . \quad (19)$$

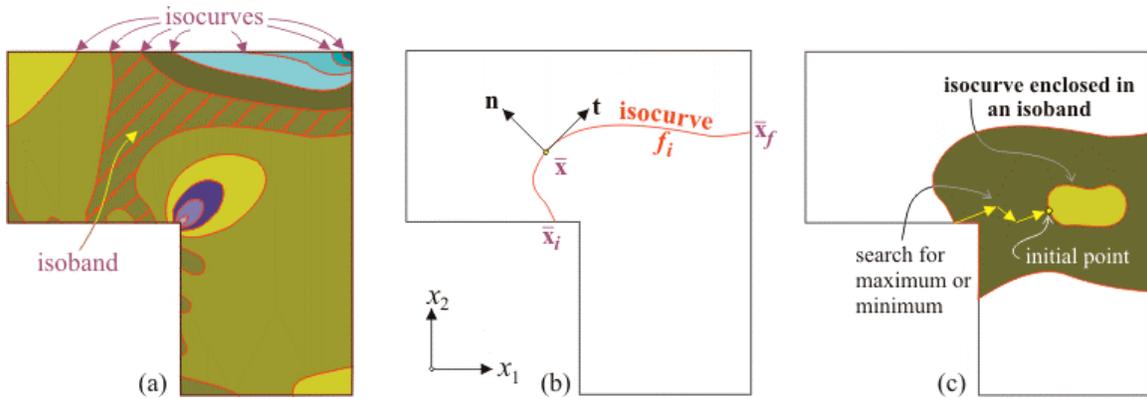


Figure 2 - (a) Isocurve plot; (b) isocurve definitions; (c) particular case of enclosed isocurves.

In these plots, it is useful to use a color palette to display the single connected regions between the isocurves, which are called isobands. The isoband color gives a clear indication of the function level over the model domain. Before presenting the proposed algorithm, this section discusses on the characteristics of the most popular isocurve plot techniques.

3.1 Scan line algorithm

The scan line algorithm is one of the simplest techniques to display isocurve plots. The algorithm consists on the evaluation of the function for each pixel included at the model domain. Therefore, it depends on the scale of the model and on the resolution of the raster display used. The plot is guided by a scan line which moves from the top to the bottom of the model. For each scan line location, the algorithm performs intersection tests to find the initial and final points of the segments over the scan line (Fig. 3a). The next step consist in the sequential evaluation of the function at the center of each pixel on the segment considered. According to the function value, the pixel is displayed using the color palette available (Fig. 3b). Figure 3c displays an isocurve plot for the horizontal stress component using the scan line algorithm for a simulation with the CBEM. For this plot, evaluations of the required function at 120.000 points were necessary. Clearly, this is not adequate for practical uses and should be avoided since it demands a high computational effort. A slight improvement of the performance of the scan line algorithm can be achieved applying a single evaluation to a rectangular group of pixels. Due to the loss of accuracy, the cost-benefit of this last approach makes it inadequate for practical purposes.

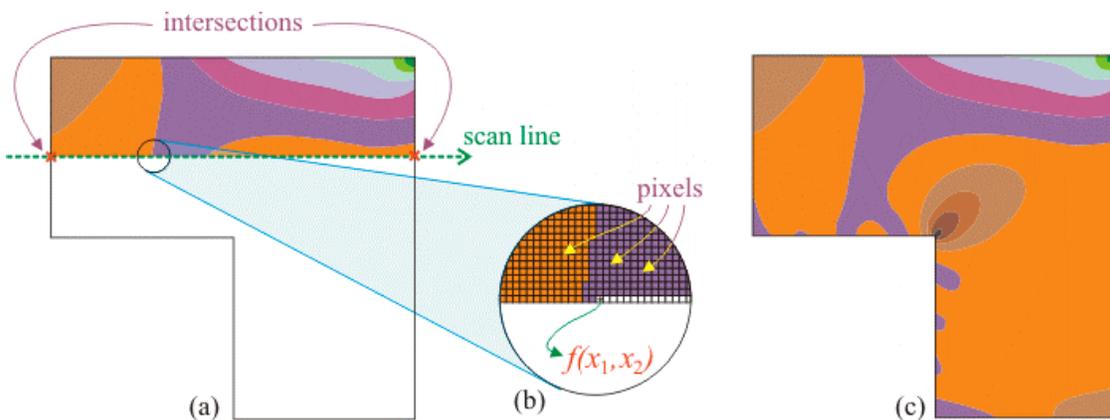


Figure 3 - (a) Scan line algorithm; (b) evaluation at pixels; (c) final plot with 120.000 pixels.

3.2 Interpolation over an auxiliary domain discretization

A better approach for the isocurve plot consists in a technique based on interpolation over an auxiliary domain discretization. Probably, this is the most common procedure used by commercially available programs. There are different versions of this algorithm. The simplest one is similar to the scan line algorithm. It requires a simple discretization of the model domain and the set of results at each node of this discretization. For each unity of the discretization, results are interpolated from the nodes using similar techniques of the FEM interpolation (Fig. 4a). This approach works well for the FEM, since a domain mesh and interpolation functions are already available. However, it works against the basic properties of the BEM, which should use only boundary discretization. In addition, the accuracy of this procedure depends on the discretization level used and can display results with insufficient quality, due to the jagged appearance of the isocurves. Another even worse aspect is that it can ignore critical isocurves in regions with high gradients of the function (Fig. 4b).

For practical purposes, this technique requires at least an adaptive discretization in order to deal with the regions with high gradients. Therefore, for the BEM the efficiency of this alternative is questionable, since the task of mesh generation at the domain attains the same complexity of the FEM modeling process.

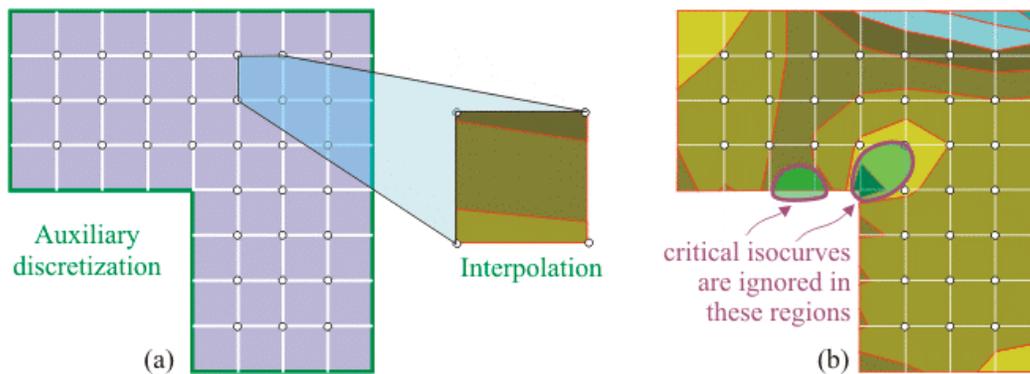


Figure 4 - (a) Domain discretization and interpolation; (b) accuracy problems.

3.3 Proposed Algorithm

In order to overcome the drawbacks of the two previous techniques, the authors of this paper developed a new algorithm for the isocurve plot in analysis with BEMs. The proposed algorithm uses the specific features of the BEM in a rational manner, preserving its basic characteristics. The central idea is to accomplish the automatic and direct identification of the isocurves. For a smooth representation of isocurves, it adjusts segments of cubic splines (Foley, 1997). To obtain the control points of the splines, the algorithm uses the predictive and iterative techniques commonly used in non-linear analysis (Crisfield, 1991). In order to restrain the computational effort, the governing parameters of the incremental process (step size, convergence criteria) should result in evaluations at a small number of internal points. For the sake of simplicity, the authors suggest a heuristic adjustment of these parameters.

The present implementation deals only with the most frequent case of isocurves emerging from the boundary. The case of isocurves enclosed in an isoband will be discussed in future works, since it requires an auxiliary algorithm to find an initial point on the isocurve (Fig. 2c).

The first step of the algorithm consists in finding the limiting points at the boundary of the model, storing them in a list according to their occurrence (Fig. 5a). For isocurves

representing the displacement field, these points should be obtained using the interpolation functions of Eq. (1). On the other hand, isocurves for the stress field need a special treatment to the hypersingular integrals present in Eqs. (7) and (10). A simple alternative to this is the so-called “traction-recovery” method (Beer, 2001).

Starting from a initial point $\bar{\mathbf{x}}_i$, the proposed algorithm consists in finding a new point lying on the isocurve, tracking its path in an incremental manner (Fig. 5b). For this task, the algorithm combines two different procedures. The first one is a prediction estimate, which locates the new point on the tangential direction using a scalar step size Δt as follows:

$$\bar{\mathbf{x}}_p = \bar{\mathbf{x}}_i + \Delta t \cdot \mathbf{t} \quad . \quad (20)$$

An adequate choice for the step size Δt will be presented later on in this section.

For curved paths, point $\bar{\mathbf{x}}_p$ will drift from the isocurve. The second procedure of the algorithm consists in using iterative techniques to find the new point $\bar{\mathbf{x}}_n$ on the isocurve. Most iterative techniques uses the Newton-Raphson method (N-R) in conjunction with constraint conditions that define the returning direction for the point $\bar{\mathbf{x}}_p$. In the last two decades, the arc-length methods (Crisfield, 1991) have been successfully used for non-linear applications.

An alternative approach to the arc-length methods is the “normal flow” method, which is related to the proposed algorithm (Ragon, 2002). This method is based on the “Davidenko flow” concept, which represents the set of perturbations to a path (Fig. 5b). For the present application, these perturbations are the set of isocurves adjacent to f_i . The normal flow method uses the normal direction to the Davidenko flow as the returning path for the N-R iterations. For the proposed algorithm this is the direction of the unit normal vector defined on Eq. (19). Thus, in order to find the new point $\bar{\mathbf{x}}_n$, successive iterations are evaluated by

$$\bar{\mathbf{x}}_j = \bar{\mathbf{x}}_{j-1} + \frac{|f_i - f(\bar{\mathbf{x}}_{j-1})|}{\sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2}} \cdot \mathbf{n}_{j-1} \quad , \quad \text{for } j = 1, 2, \dots \quad , \quad (21)$$

where the first point corresponds to the result of the prediction phase ($\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}_p$).

The new point $\bar{\mathbf{x}}_n$ is obtained by recursive evaluations of Eq. (21) until the fulfillment of a convergence criteria, which should be a simple error norm as

$$\varepsilon = \frac{|f(\bar{\mathbf{x}}_j) - f_i|}{\Delta f_i} \quad , \quad (22)$$

where Δf_i is the isocurves level interval. The present implementation considered $\varepsilon = 0.001$. The choice of the normal direction as the returning path results in a unique minimum norm solution for the N-R iterations. Due to this feature, it is expected that a small number of iterations (1 or 2) will satisfy Eq. (22).

The algorithm proceeds with new prediction and iterative steps, using point $\bar{\mathbf{x}}_n$ of the N-R iterations as the initial point ($\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_n$), until it finds the final point $\bar{\mathbf{x}}_f$ at the boundary. Then, it removes $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_f$ from the list of limiting points. If this list is not empty, the algorithm goes further to the next isocurve.

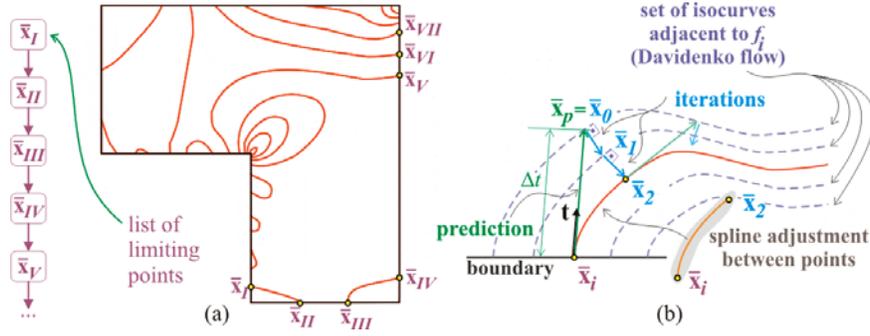


Figure 5 - (a) Limiting points; (b) predictive and iterative phases.

The present algorithm requires the partial derivatives of $f(\mathbf{x})$ during the predictive and iterative processes. This additional information allows a suitable approximation of the curve between the points \bar{x}_i and \bar{x}_n using cubic splines, since the tangent vectors are promptly available. Normally, just few spline segments will represent the isocurve with good accuracy.

The step size Δt used for the prediction phase depends on several factors, e.g. the isocurve distribution, the model geometry and the required degree of accuracy. There are a myriad of ways to combine these factors, making the choice of an appropriate step size Δt a relatively difficult task. The present implementation used a heuristic approach to determine the step size Δt . It is a simple attempt to avoid isocurve intersections, setting Δt as the half of the distance between two points adjacent to the initial point (Fig. 6a). These adjacent points are either the initial points of other isocurves or the model corners.

For each prediction estimate, in order to limit the deviation from the actual isocurve level to 20%, the algorithm evaluates a new step size Δt_{new} as follows:

$$\Delta t_{new} = \frac{0.2 \Delta f_i}{|f(\bar{x}_p) - f_i|} \Delta t \quad (23)$$

To avoid strong deviations, a range is established for the new step size, varying between half and double of the Δt value.

If the current values of $f(\bar{x}_p)$ differs more than 20% from f_i , the algorithm performs new predictions using Δt_{new} , before the iterative phase, until the error norm of Eq.(22) is less than 20%. On the other hand, Δt_{new} will be used only for the prediction phase of the next segment of the isocurve, thus providing an automatic adjustment for the step size (Fig. 6b).

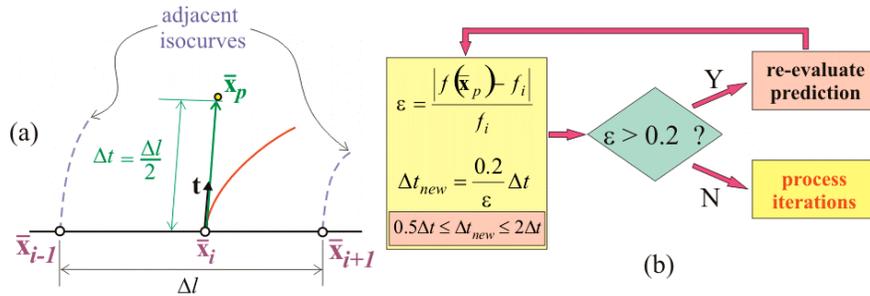


Figure 6 - Step size: (a) initial determination; (b) automatic adjustment.

4. IMPLEMENTATION

Recently, the authors have been involved in the development of a computational platform for BEMs. The platform consists in an integrated system with an analysis core and graphical pre- and postprocessors, using the features and benefits of the object-oriented programming (OOP) with Java (Booch, 1999) (Ammeraal, 1998).

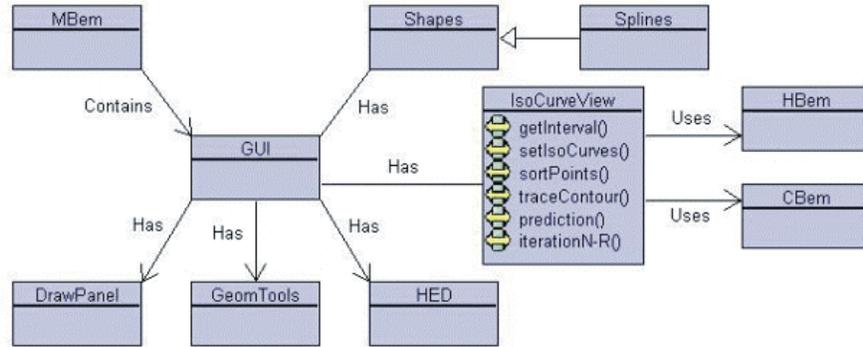


Figure 7 - Class diagram for the postprocessor with the new class *IsocurveView*.

In other to implement the proposed algorithm, the application of object-oriented analysis and design processes resulted in the inclusion of the new class *IsoCurveView* in the existing postprocessing class architecture. Figure 7 displays the class diagram using UML Notation (Booch, 1999). This new class uses cubic splines, Half-edge (HED) data structure (Mäntylä, 1988) and several graphical algorithms (intersections, point location) implemented in the postprocessor. Due to the HED, the implementation allows the automatic identification of the isobands. Additionally, the class *IsocurveView* is associated to classes *CBem* and *HBem*, which represent the conventional and hybrid BEM, respectively, allowing the evaluation of displacements and stresses (and their derivatives) in interior points with both methods.

The necessary components for the implementation of the stresses derivatives according to the expressions given by Eqs (10) and (15) were introduced in classes *CBem*, *HBem*, *FundSol* and *Elastostatic*. Figure 8 shows the class diagram of the analysis core and the corresponding classes involved in the stress derivatives implementation.

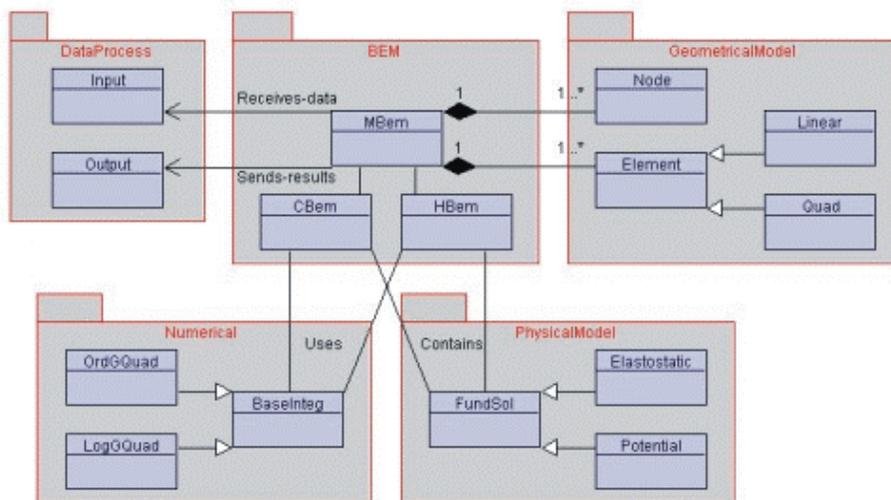


Figure 8 - Class diagram for the analysis core and the implementation of derivatives.

The sequence of steps of the new visualization algorithm might be represented using a Finite State Machine, a valuable tool in the object-oriented analysis and design. Figure 9 displays the state transition diagram which indicates the dynamic behavior of class *IsoCurveView* using UML Notation (Booch, 1999).

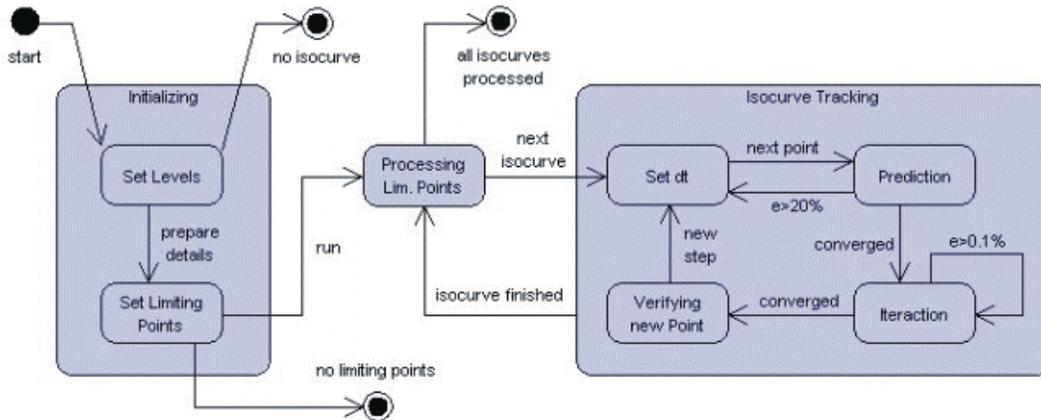


Figure 9 - State transition diagram of class *IsoCurveView*.

5. NUMERICAL RESULTS AND EXAMPLES

In order to illustrate the different aspects related to the visualization techniques discussed in this article, this section presents results using the proposed algorithm, the interpolation over domain discretization and the scan line technique.

5.1 Results with the proposed algorithm

This section presents two test examples which demonstrate the proposed algorithm accuracy and efficiency in analyses with the CBEM. The first example consists in a cantilever beam with a load applied to its free end. For this analysis, a regular discretization with 48 linear elements was used. To avoid quasi-singular integrals, the present implementation used an auxiliary boundary inside the real boundary. The distance between both is half of the element size. Figure 10 shows the results for the σ_{xx} stress field using a set of 11 equally spaced isocurves. This plot required evaluations at 43 interior points, using just one N-R iteration to correct the prediction estimate. The resulting isocurves agree very closely with the analytical results from the theory of elastic bending of beams.

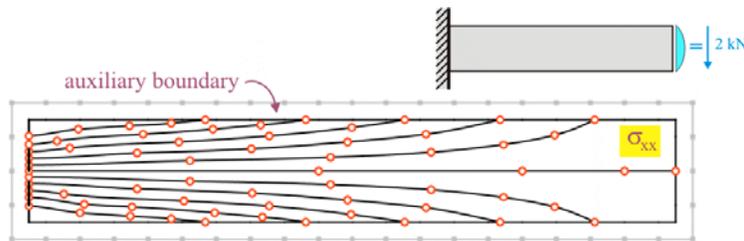


Figure 10 - Application of the algorithm for the analysis of a cantilever beam.

A comparison between the proposed algorithm and the interpolation over a domain discretization has been made using analysis with the FEM. As Figure 11 depicts, the results of the proposed algorithm using 43 interior points were more accurate than a mesh using 80 finite elements of type Q9, which has 369 interior points. Additionally, in contrast with the domain discretization, the proposed algorithm resulted in smoothly continuous isocurves.

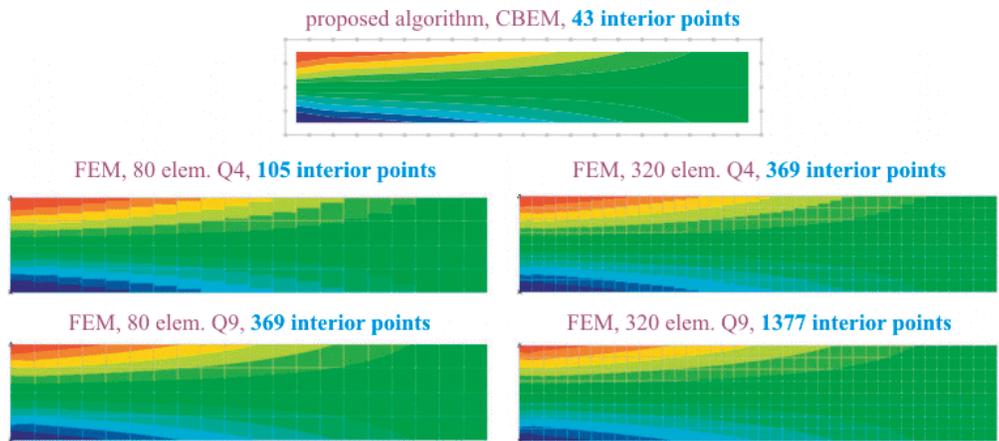


Figure 11 - Proposed algorithm and the interpolation over a domain discretization.

The second example consists in a L-shape body with stress concentration. The model representation used a regular discretization with 40 linear elements. Figure 12a presents the algorithm results. This plot required evaluations at 115 interior points. To correct the prediction estimate, 58 interior points required just one N-R iteration, 54 points required 2 iterations, 2 points required 3 iterations and just 1 point required 4 iterations. Figure 12b shows an enlarged view of an area of the isocurve plot to closely inspect the predictive and iterative processes.

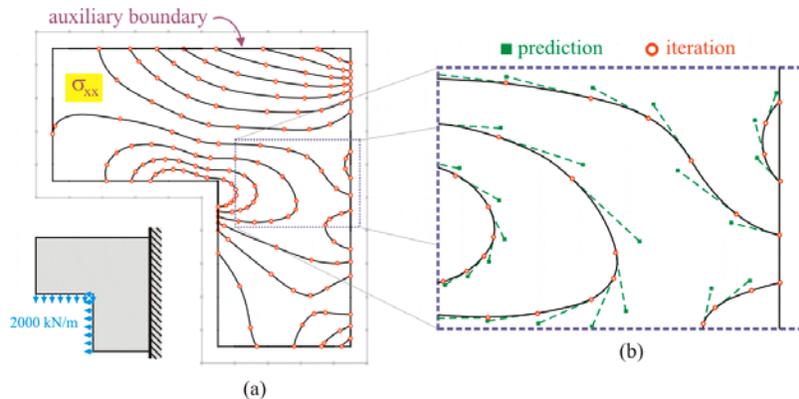


Figure 12 - (a) Isocurves for a L-Shape; (b) close view of the incremental process.

The comparison using the interpolation over domain discretization using the FEM is depicted in Fig. 13. The results with the proposed algorithm using 115 interior points were more accurate than the plot for a domain discretization with 320 finite elements of type Q9, which has 1281 interior points.

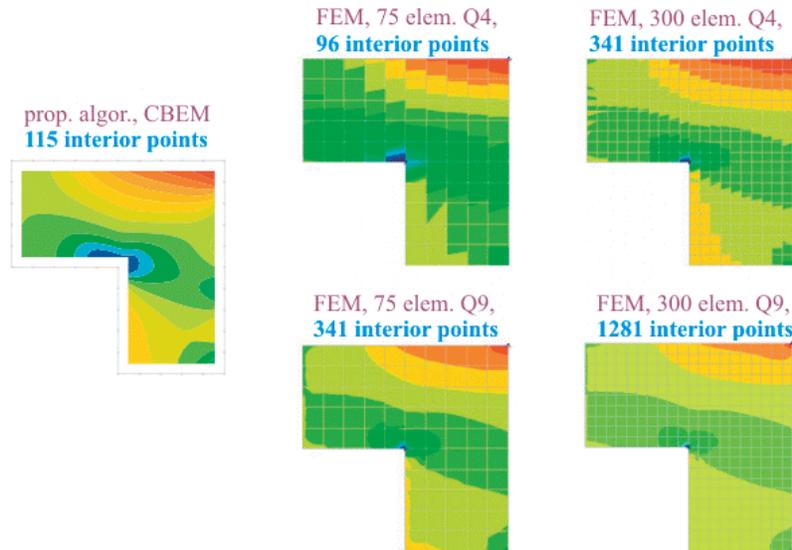


Figure 13 - Comparison for the L-Shape.

5.2 Results with the scan line algorithm for CBEM and HBEM.

As discussed in section 2, evaluation of results in interior points using the HBEM is much simpler and faster than the procedure required by the CBEM. This section presents results for the scan line algorithm applied to both methods, comparing the computing time required only for the visualization. Figure 14 displays results for a simply supported beam represented by 48 linear boundary elements.

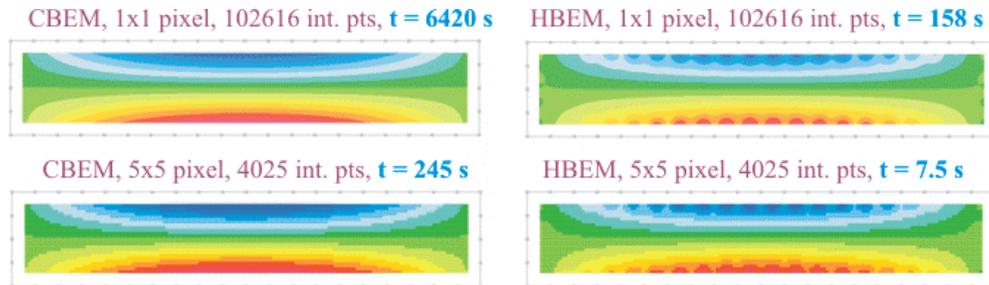


Figure 14 - Comparisons using scan line for a simply supported beam.

The scan line algorithm was applied considering evaluations at a single pixel (1x1) and at a group of 5x5 pixels as well. Due to the integral calculations, the CBEM required a much higher computing time (about a factor of 35) than the HBEM. On the other hand, problems with jumps and lack of accuracy in regions close to the boundary were more pronounced in the HBEM. These effects can be reduced with a more refined discretization. Figure 15 shows results with the HBEM for a mesh with 96 boundary elements. Now the solution behavior is practically the same of the CBEM, and the computing time for these visualizations was only twice that of the previous examples with the HBEM.

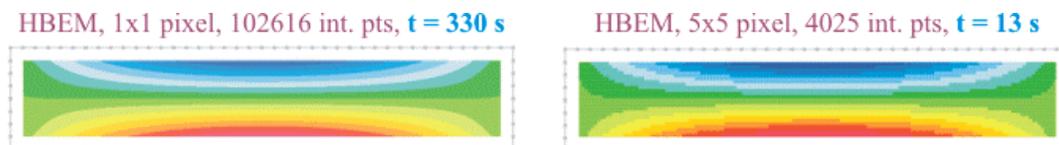


Figure 15 - Results using a more refined discretization.

The results of the application of the scan line algorithm to the L-Shape also presented the same characteristics from the previous example (Fig. 16). Again, the computing time required by the CBEM was about 35 times higher than the analysis with the HBEM.

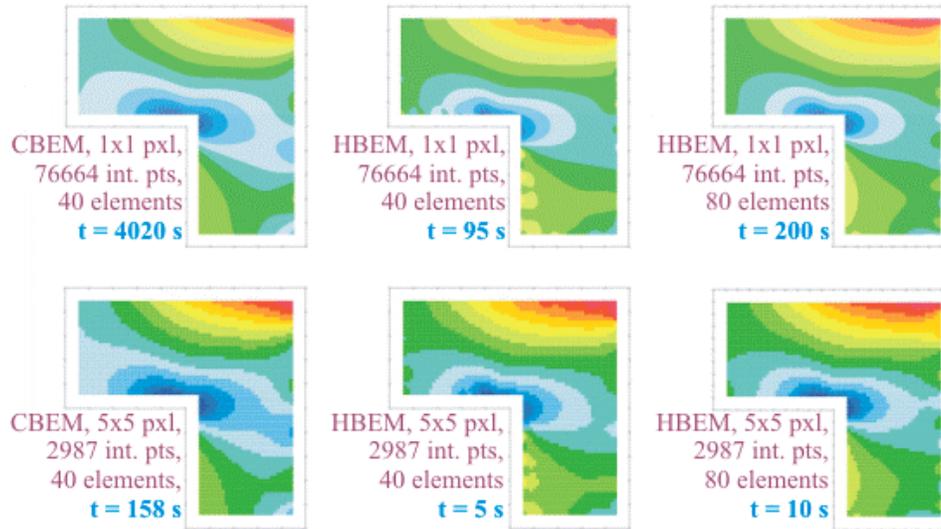


Figure 16 - Comparison for the L-Shape.

It is worth to emphasize that the results presented in this section are not conclusive about the characteristics of the CBEM and the HBEM in visualization of results. The computing time necessary for the first part of the analysis (solution at the boundary) has also to be considered. In addition, an adequate integration scheme for the CBEM has to be used, allowing the visualization of the results at the whole domain.

6. CONCLUSIONS

This article addressed the task of visualization of domain results in analysis with BEMs. In the last two decades, much research has been done on the development and implementation of BEMs. However, the advances in the visualization algorithms were insignificant. After an extensive literature review, the authors have found that the visualization task for BEMs practically uses the same techniques for the FEM. Therefore, they proposed a new algorithm which makes a convenient use of the specific BEM properties, as the flexibility of inquiring results and their derivatives at interior points. Several examples using different visualization techniques demonstrate the accuracy and efficiency of the proposed algorithm. In addition, this work also carried out performance tests of two different formulations of the BEM.

The tests with the scan line algorithm show that this technique requires a high number of evaluations, but its usage is not prohibitive for analysis using the HBEM, since it involves a low cost operation to evaluate results at interior points. However, further investigations are necessary, considering the computing time required to obtain the solution at the boundary, which normally is considerable in analysis with the HBEM. Another important aspect is the problem with jumps and lack of accuracy at regions close to the nodes of the discretization.

The visualization task using interpolation over domain discretizations requires much less evaluations at interior points than the scan line algorithm. However, this technique requires a fine domain discretization to give results with satisfactory accuracy. Another drawback of this technique is that it might ignore important results in regions with high gradients. In addition, the necessity of a domain discretization makes this approach unattractive for the BEM.

Compared to the other two techniques, the algorithm proposed in this work presented more accurate results and required less evaluations at interior points. This algorithm makes a good usage of the BEM features, preserving its basic characteristics. In addition, it makes possible the identification of isocurves and isobands in a straightforward manner. This fact permits the extension of the proposed algorithm to other applications, e.g. non-linear analysis with BEMs. Further investigations are also necessary to improve the proposed algorithm, in order to adjust the governing parameters, to deal with enclosed isocurves and to better determine its accuracy, convergence and stability features.

Acknowledgements

The authors acknowledge the support of Brazilian agencies FAPESP and of the University of São Paulo.

REFERENCES

- Ammeraal, L., 1998, Computer Graphics for Java Programmers, Wiley, Engl.
- Beer, G. & Watson, J., 1995, Introduction to Finite and Boundary Element Methods for Engineers, Wiley, Chichester.
- Beer, G., 2001, Programming the Boundary Element Method, Wiley, Chichester.
- Booch, G.; Rumbaugh J. and Jacobson, I., 1999, The Unified Modeling Language -User Guide, Addison Wesley.
- Brebbia, C. A. & Domingues, J., 1989, Boundary Element Method: An Introductory Course, Comp. Mech. Publications, Southampton.
- Crisfield, M. A., 1991, Non-linear Finite Element Analysis of Solids and Structures, Wiley, England.
- Dumont, N. A., 1989, The Hybrid Boundary Element Method: An Alliance Between Mechanical Consistency and Simplicity, Applied Mechanics Reviews, vol. 42, n 11, part 2, pp. S54-S63.
- Dumont, N. A. & Noronha, M., 1998, A Simple Scheme for the Numerical Evaluation of Integrals with Complex Singularity Poles, Comp. Mech. 22, pp. 42-49.
- Farin, G.; Hoschek, J. and Kim, M-S., 2002, Handbook of Computer Aided Geometric Design, Elsevier, Netherlands.
- Foley, J. D.; van Dam, A.; Feiner, S. K. and Hughes, J. F., 1997, Computer Graphics, Addison Wesley, USA.
- Gomes, J. & Velho, L., 1998, Computação Gráfica, Volume 1, Série Computação e Matemática, SBM/IMPA.
- Mäntylä, M., 1988, Introduction to Solid Modeling, Comp. Science Press.
- Ragon, S. A.; Gürdal, Z. and Watson, L.T., 2002, A Comparison of Three Algorithms for Tracing Nonlinear Equilibrium Path of Structural Systems, International Journal of Solids and Structures, vol. 39, pp. 689-698.
- Souza, R. M. de, 1992, O Método Híbrido dos Elementos de Contorno para a Análise Elastostática de Sólidos, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC-Rio, Rio de Janeiro.
- Wolff, R. & Yaeger, L., 1993, Visualization of Natural Phenomena, Springer Verlag, New York.