GAMAT2 - GERADOR AUTOMÁTICO DE MALHAS TRIANGULARES DE ELEMENTOS FINITOS LINEARES E QUADRÁTICOS

Antonio C. S. Guimarães e Raul A. Feijóo
Laboratório Nacional de Computação Científica
Rua Lauro Muller 455, 22290, Rio de Janeiro, Brasil

RESUMO - O presente trabalho tem por objetivo apresentar o desenvolvimento do programa **GAMAT2** que permite a geração automática de malhas de elementos finitos de tipo triangular linear ou quadrático em regiões bidimensionais arbitrárias multiplamente conexas.

1. INTRODUÇÃO

O Método dos Elementos Finitos (MEF) é uma das ferramentas disponíveis atualmente que se caracteriza por sua versatilidade e por sua capacidade em resolver os problemas, cada vez mais complexos, que surgem na engenharia. Uma das dificuldades no emprego do método reside na preparação dos dados a serem fornecidos aos programas de cálculo. Esta etapa geralmente consome muito tempo, é tediosa e causadora de inúmeros erros.

Uma etapa importante na preparação dos dados é a relacionada com a malha de elementos finitos, definição das incidências e coordenadas dos nós necessários para isto.

Nos últimos dez anos, numerosos pesquisadores vêm concentrando seus trabalhos no desenvolvimento de técnicas automáticas de geração de malha como meio de reduzir o tempo e o esforço necessários para elaborar os modelos de elementos finitos.

Os geradores automáticos de malhas mais conhecidos empregam, entre outras, alguma (ou a combinação) das seguintes técnicas:

- triangularizações automáticas [1-6],
- transformação de coordenadas e
- mapeamentos conformes, isoparamétricos, transfinitos [7-13],
- árvores modificadas [14-16],

Nos trabalhos de Thacker [17], e de Ho-Le [18], pode-se encontrar uma extensa bibliografia, assim como uma classificação e revisão dos trabalhos mais significativos na área.

O presente trabalho tem por objetivo apresentar o desenvolvimento de um programa de geração automática de malhas de elementos finitos de tipo triangular em regiões bidimensionais arbitrárias. Baseado no algoritmo proposto por Lo [21], foi desenvolvido em linguagem C o programa **GAMAT2**. Os aspectos mais relevantes desta implementação assim como algumas aplicações, para mostrar a versatilidade do algoritmo, são apresentados a seguir.

2. GERAÇÃO AUTOMÁTICA DE TRIÂNGULOS

Entre as diferentes técnicas de geração automática de malhas, uma que se apresenta como vantajosa quando comparada às outras é a

geração automática de triângulos. Entre estas vantagens podemos mencionar:

- o domínio pode ser de forma qualquer, simples ou multiplemente conexo,
- os contornos do domínio podem ser também sumamente complexos, sem por isto afetar a performance do processo de geração,
- não é necessário subdividir o corpo em subregiões convexas,
- o número de elementos em torno de um nó comum não está limitado e a posição relativa destes nós não está predeterminada por alguma fórmula ou transformação matemática,
- o tempo necessário para a geracão da triangularização é virtualmente independente da topologia e da geometria do domínio bidimensional que se deseja triangularizar,

Basicamente, a região bidimensional que se deseja triangularizar é particionada em uma ou mais subregiões. No caso particular de uma triangularização homogênea (isto é, todos os elementos tem aproximadamente o mesmo tamanho), pode-se empregar uma só região, porém se recomenda a partição em mais de uma região, já que isto permitirá reduzir os tempos computacionais gastos na geração da malha. Esta partição põe em evidência uma série de contornos que limitam o domínio e permitem caracterizar cada uma das subregiões.

O contorno de cada uma destas subregiões está caracterizado pela união de segmentos de reta e de círculo (a extensão a outros tipos de curvas planas não apresenta dificuldade alguma) orientados de maneira tal que, quando o contorno é percorrido, a subregião encontra-se sempre á esquerda. Se a subregião é simplesmente conexa o anterior implica em:

- o contorno está formado por um laço fechado simples de segmentos de reta e círculos,
- o contorno será percorrido no sentido anti-horário.

No caso de subregião multiplemente conexa, o contorno estará formado por vários laços fechados, o externo percorrido no sentido anti-horário e os internos (tantos quantas aberturas internas existam na subregião) percorridos em sentido horário.

Sobre cada um destes segmentos de reta ou círculos é gerado um conjunto de pontos. Estes pontos permitem construir a lista inicial dos segmentos que serão lados dos triângulos a serem gerados na subregião. Esta lista de lados é comumente chamada de "front inicial" ou simplesmente de "front".

Posteriormente, o algoritmo gera de maneira simples e eficiente um conjunto de nós que pertenecem ao interior da região. A lista de nós assim gerados será chamada de "nós interiores". Com o "front inicial" e com os "nós interiores" o algoritmo conecta todos estes nós de maneira a gerar triângulos que não se superpõem, tais que a união dos mesmos gere toda a subregião e, finalmente, de forma o mais equilátera possível na medida em que os "nós interiores" o permitam.

3. DEFINIÇÃO DOS CONTORNOS E SUBREGIÕES

Como já mencionamos, o primeiro passo no algoritmo consiste em caracterizar o domínio, que chamaremos de **CORPO**, e cada uma das subregiões em que foi particionado. Temos assim os seguintes atributos para o **CORPO**:

CORPO <contornos; nós; subregiões>

Os contornos ficam definidos através dos seguintes atributos:

CONTORNO <cont, noi, nof, npart, tipo, coorxc, cooryc>

onde:

cont : número do contorno,
noi : número do nó inicial,
nof : número do nó final,
npart : número de partições que permite ao algoritmo gerar a lista "front",
tipo : RETO ou CIRCULAR
coorxc: coordenada x do centro do círculo (somente para tipo CIRCULAR)
cooryc: coordenada y do centro do círculo (somente para tipo CIRCULAR.

Como podemos observar, para definir cada contorno é necessário fornecer o número do nó inicial e final. Assim, para a caracterização destes contornos é necessário ainda fornecer as coordenadas destes nós. Em outras palavras, os atributos dos **NOS** são as suas coordenadas:

NOS <coorx; coory>

Para facilitar a geração dos triângulos, permitindo ter diferentes tamanhos de elementos no corpo, assim como diminuir o tempo gasto na geração, o corpo pode ser particionado em uma ou várias subregiões.

Estas regiões podem ser de tipo SHARP, RADIAL, ANEL ou OUTRAS, dependendo do tipo de refinamento e/ou geração dos nós interiores a estas subregiões (figuras 1a e 1b).

A definição de cada uma destas regiões depende do tipo. Em particular para todas elas é necessário fornecer quais são os contornos que as limitam. Mas, desta vez, os contornos deverão estar orientados, isto é, o sentido noi→nof deve ser tal que a região fique à esquerda (figura 1b).

Por exemplo, a região de tipo OUTRAS (figura 1b) fica definida através dos seguintes atributos:

região OUTRAS <ncont; [cont, noi, nof]$_i$, i = 1, ncont; h; nfix;

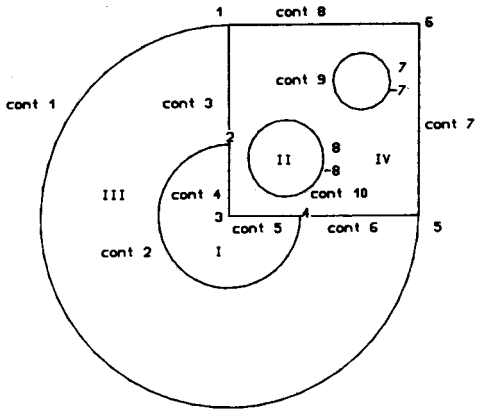
[cx, cy]$_i$, i = 1, nfix>

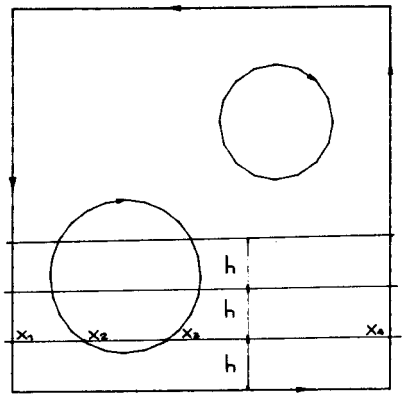
onde:

ncont : é o número de contornos que limitam a região de tipo OUTRAS
cont$_i$: numero do i-ésimo contorno da região
noi$_i$-nof$_i$: nós inicial e final do i-ésimo contorno *orientado*
h : tamanho médio do elemento

nfix : número de nós interiores fixos
cx_i-cy_i : coordenadas x e y do i-ésimo nó fixo

Os nós fixos constituem um recurso disponível ao usúario para que, durante a geração dos triângulos, certos nós estejam já pre-estabelecidos facilitando a modelagem.





(a) (b)

Figura 1: I - SHARP, II - RADIAL, III - ANEL, IV - OUTRAS

4. GERAÇÃO DO "FRONT" INICIAL E DOS "NÓS INTERIORES"

A partir da definição das regiões, o algoritmo procede a geração do que chamamos de "front" e de "nós interiores".

Geração do "front"

A geração do front fica enormemente simplificada ao trabalharmos em linguagem C. Em efeito o algoritmo procede da seguinte forma:

1) para cada contorno $cont_i$ de uma dada região se constróem os $npart_i$ segmentos (de tipo RETO ou CIRCULAR dependendo do tipo do contorno $cont_i$) orientados e a informação relativa a cada uma destes segmentos é inserida em uma lista chamada "front".

Geração dos "nós interiores"

Embora existam diversas técnicas para a geração de nós interiores a uma região [17-20], adotou-se a proposta por Lo [21] por sua simplicidade, por sua versatilidade para adaptar-se a qualquer tipo de domínio e por sua eficiência computacional. Assim o algoritmo procede a:

2) a partir da informação contida em "front" se calcula y_{min} e y_{max} da região,

3) linhas horizontais imaginárias entre y_{min} e y_{max} são construídas, sendo sua separação dada por h, tamanho médio do elemento na região.
4) para cada linha horizontal dada pela equação y = H, se calcula as intersecções com cada um dos segmentos contidos em "front",
5) em cada linha horizontal teremos sempre um número par de intersecções que por sua vez são colocados em ordem crescente, isto é $x_1 < x_2 < x_3 < x_4 <...< x_{2n}$ (figura 1b).
6) asumindo que existem 2n intersecções, e tomando estas intersecções de duas em duas (começando pela primeira e segunda intersecções) são gerados pontos igualmente espaçados na magnitude h. Estes nós serão incorporados a lista "nós interiores" se:
   - a distância a qualquer nó de "front" seja maior ou igual a 0.7 x h,
   - a distância a qualquer nó de tipo interior fixo seja maior ou igual a 0.7 x h,
7) repetir o processo até esgotar todos os pares de intersecções e todas as linhas horizontais imaginárias.
8) incorporar na lista "nós interiores" os nós fixos da região.

As vantagens deste proceso de geração dos nós interiores são óbvias e foram resaltadas por Lo [21]. Assim temos, por exemplo:
- os nós interiores já gerados não precisam ser levados em conta na geração de um novo nó (a distância entre eles será sempre maior que 0.7 x h),
- não é necessário implementar nenhum teste vigoroso de consistência como ocorre geralmente nos procedimentos comumente existentes na literatura [21].

O anterior se traduz em um algoritmo muito eficiente, tanto do ponto de vista de sua implementação como do ponto de vista da sua eficiência no referente a tempo computacional gasto na geração destes nós interiores.

A geração de nós interiores nas regiões tipo SHARP, RADIAL e ANEL segue um procedimento algo diferente ao anteriormente descrito, que procura gerar um refinamento progressivo.

Considere dois arcos de círculos concêntricos e sejam d_e e d_i o espaçamento entre os nós em cada um destes arcos, novos nós interiores são gerados de maneira a estarem em um outro arco de círculo concêntrico aos anteriores e caracterizado pelas seguintes expreções (Figura 2):

$$\lambda = \frac{S + d_e}{S + d_i}, \qquad d = \frac{\lambda - 1}{\lambda^N - 1} S, \qquad d_i = \frac{d_e}{exp\ (N \cdot \log \lambda)}$$

onde N é o numero de camadas de triângulos a serem gerados entre ambos os arcos de círculo.

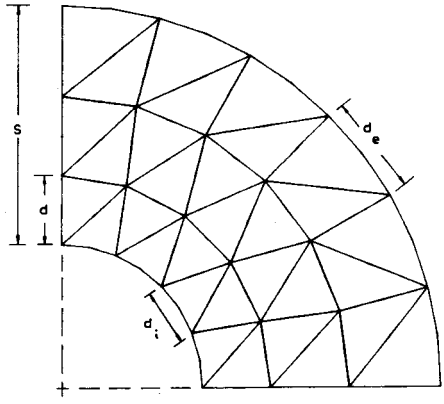


Figura 2.

5. GERAÇÃO DOS TRIÂNGULOS

Tendo já construídas as duas lista "front" e "nós interiores", passamos a descrever como o algoritmo procede a interconectar estes nós de maneira a gerar a triangularização.

Como já mencionamos e devido a orientação dos contornos, ao percorrer no sentido noi→nof qualquer segmento pertencente a "front", a região a ser triangularizada sempre estará a esquerda.

Embora o contorno da região sempre seja o mesmo durante todo o processo de triangularização, o mesmo não ocorre com o "front". Na etapa inicial ambos são exatamente iguais mas, a medida que um novo triângulo é gerado, o "front" é modificado, assim como a lista de "nós interiores". O precedimento continua até que as listas "front" e "nós interiores" não tenham nenhum elemento, isto é, estejam vazias.

Para gerar um triângulo se procede da seguinte forma:

1) da lista "front" se constrói uma outra lista contendo os nós do front que chamaremos "nofront".
2) Se a lista "front" não está vazia, se toma um elemento da mesma, por exemplo o primeiro da lista que chamaremos de segmento AB. O método consiste em escolher entre os nós existentes em "nofront" e "nós interiores" o nó C tal que se encontre a esquerda do segmento AB e tal que o triângulo ABC seja "ótimo" em algum sentido que explicitaremos mais adiante.
   A escolha do nó C entre os nós de "nofront" ou de "nós interiores" permite eliminar todo teste de consisténcia associado à possíveis intersecções com a fronteira da região ou com os triângulos já gerados, o que garante uma grande eficiência computacional do algoritmo. O único teste está relacionado com a colocação do nó C em relação ao segmento AB. Isto é facilmente estabelecido exigindo-se que $\Delta ABC > 0$ (ΔABC = 0.5 . AB × BC, área do triângulo).

3) Escolhido o nó C, o segmento AB é eliminado da lista "front" e as listas "front", "nofront" e "nós interiores" são atualizadas.
4) O processo se repete até que as listas "front" e "nós interiores" fiquem vazias, o que deve acontecer simultaneamente (figura 3).

Escolha do nó C

A escolha do nó C se realiza da seguinte maneira:

1) Se determinam os nós C_1 e C_2 tais que

$$\Delta ABC_1 > 0 \text{ e } \Delta ABC_2 > 0$$

$$(AB^2 + BC_1^2) \leq (AB^2 + BC_2^2) \leq (AB^2 + BD^2)$$

$$\forall\ D \in \text{"nofront" e "nós interiores"}$$

2) A escolha entre os nós C_1 e C_2 é feita de maneira a escolher o triângulo mais equilátero possível para o qual as seguintes expreções são calculadas (Lo [21]):

$$\alpha_1 = \Delta ABC_1 \,/\, (AB^2 + BC_1^2 + C_1A^2)$$

$$\alpha_2 = \Delta ABC_2 \,/\, (AB^2 + BC_2^2 + C_2A^2)$$

$$\beta_1 = \Delta C_1BC_2 \,/\, (C_1B^2 + BC_2^2 + C_2C_1^2), \quad \beta_2 = -\beta_1$$

$$\vartheta_1 = \Delta AC_1C_2 \,/\, (AC_1^2 + C_1C_2^2 + C_2A^2), \quad \vartheta_2 = -\vartheta_1$$

$$\lambda_1 = max\,(\beta_1, \vartheta_1), \quad \lambda_2 = max\,(\beta_2, \vartheta_2)$$

Com estes valores o nó C_1 é escolhido se $\alpha_1\lambda_1 > \alpha_2\lambda_2$ e vice-versa.

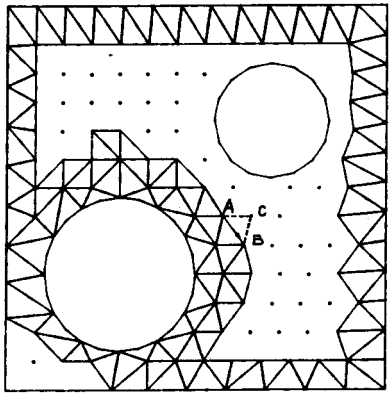


Figura 3. Estágio intermediário da geração de triângulos

6. REGULARIZAÇÃO DOS TRIÂNGULOS E NUMERACÃO ÓTIMA DOS NÓS

Durante o processo de geração automática de elementos finitos, alguns destes elementos podem ser gerados com uma forma inadequada. Nestes casos, técnicas de regularização que permitam melhorar a malha resultam sumamente úteis. Uma das técnicas de regularização mais popular é a "regularização Laplaciana", adotada no presente trabalho, na qual se procura estebelecer a nova posição dos nós de maneira que os nós internos fiquem no centróide do polígono formado pelos elementos finitos a eles associados. Esta reposição geralmente é feita de maneira iterativa, embora esquemas diretos existem na literatura.

Em alguns casos, a "regularização Laplaciana" pode proporcionar ainda elementos de forma inadequada. Herrmann [7] propõe nestes casos uma reposição do nó interno dada pela expressão:

$$P_j = \frac{1}{N\,(2 - w)} \sum_{n=1}^{N} (P_{nj} + P_{nl} - w\,P_{nk})$$

onde N é o número de elementos ao redor do nó "i" e w (0 ≤ w ≤ 1) é um coeficiente que caso tome o valor w = 0 a expressão anterior correspondera à regularização Laplaciana. Se w = 1 teremos a técnica chamada de "isoparamétrica".

Outro aspecto muito importante no processo de geração automatica é o relacionado com a numeração dos nós e/ou elementos. Esta numeração de nós e elementos deve ser feita de maneira a reduzir os tempos computacionais necessários na montagem e resolução do sistema de equações que surge ao empregar o MEF. No caso de adotarmos uma técnica de resolução do tipo frontal, a correta numeração dos elementos é importante. Nos outros casos, tais como técnicas de Gauss ou Choleski, é importante a numeração dos nós.

Na presente versão do programa GAMAT2 foi prevista a otimização da semi-largura de banda do sistema de equações, isto é, os nós são numerados de maneira a minimizá-la [8].

7. APLICAÇÕES

EXEMPLO 1

O exemplo 1 apresenta a triangularização antes e após o processo de "smooth" do corpo empregrado na figura 1. O número total de nós gerados é de 869 e o número de elementos quadráticos de 412 (figuras 4a e 4b).

EXEMPLO 2

A figura 5 apresenta em escala log-log os tempos gastos na geração de triângulos por número de elementos gerados. Os equipamentos empregados são um micro computador XT com clock de 8 Mhz e um AT com clock de 12 Mhz. A curva "1 Reg. XT" corresponde à geração de 50, 200 e 800 elementos tomando-se apenas uma região. As curvas "4 Reg. XT" e "4 Reg. AT" correspondem à geração de 200, 800 e 3200 elementos tomando-se 4 regiões com 50, 200 e 800

elementos respectivamente. Como se pode observar, é conveniente a partição da região em subregiões, já que isto acelera o processo de geração.

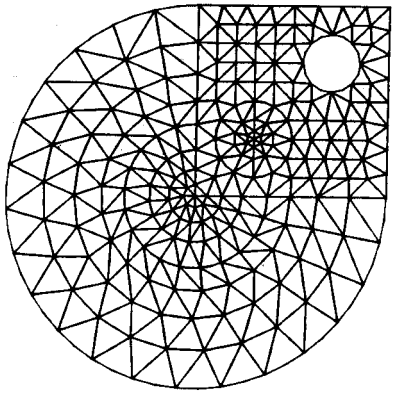


Figura 4a: Sem SMOOTH

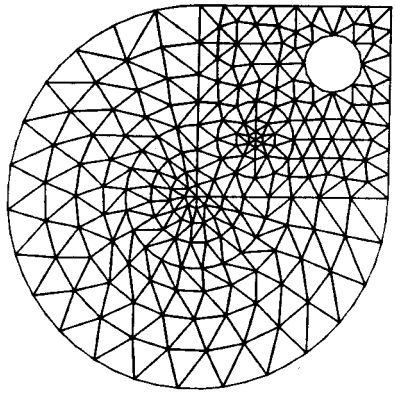


Figura 4b: Com SMOOTH

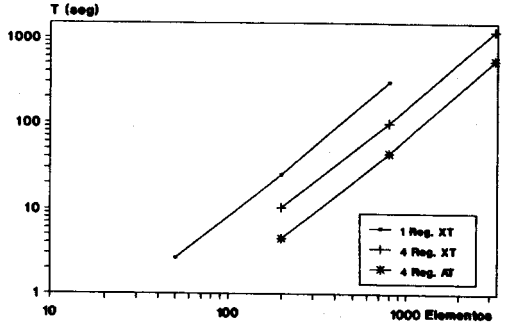


Figura 5. Performance

BIBLIOGRAFIA

[1] CAVENDISH, J.C.; "Automatic Triangularization of Arbitrary Planar · Domains for the Finite Element Method", Int. J. Numerical Methods Eng., Vol. 8, No 4, 1974, 679-696.

[2] TRACY, F.T.;"Grafics Pre- and Post-Processor for Two-Dimensional Finite Element Programs"; Computer Grafics, Vol. 11, No 2, 1977, 8-12.

[3] BYKAT, A.; "Automatic Generation of Triangular Grid: I-Subdivision of General Polygon into Convex Subregions. II-Triangulation of Convex Polygons"; Int. J. Numerical Methosd Eng., Vol. 10, No 6, 1976, 1329-1342.

[4] SADEK, E.A.; "A Scheme for the Automatic Generation of Triangular Finite Elements"; Int. J. Numerical Methods Eng., Vol. 15, No 12, 1980, 1813-1822.

[5] CAVENDISH, J.C., FIELD, D.A., FREY, W.H.; "An Approach to Automatic Three-Dimensional Finite Element Mesh Generation";

Int. J. Numerical Methods Eng. Vol 21, 1985, 329-347.

[6] BOUBEZ, T.I., FUNNELL, W.R.J., LOWTHER, D.A. PINCHUK, A.R. and SILVESTER, P.P.; "Mesh Generation for Computational Analysis"; Compt.-Aided Eng. J., 1986, 190-201.

[7] HERMANN, L.R.; "Laplacian-Isoparametric Grid Generation Scheme"; J. of the Mechanics Division of the American Society of Civil Engineers, Vol 102, No EM5, 1976, 749-756.

[8] VENERE, M.J.; "Generador de Redes ENREDO"; Publicação Interna Centro Atómico Bariloche, Argentina, 1987.

[9] BROWN, R.P.; "A Non-Interactive Method for the Automatic Generation of Finite Element Meshes Using the Schwarz-Christoffel Transformation"; Computer Methods Appl. Mechanics an Eng., Vol. 25, No 1, 1981, 101-126.

[10] ZIENKIEWICZ, O.C., PHILIPS, D.V.; "An Automatic Mesh Generation Scheme for Plane and Curves Surfaces by Isoparametric Co-ordinates"; Int. J. Numerical Methods Eng., Vol. 3, No 4, 1971, 519-528.

[11] COOK, W.A.; "Body Oriented (natural) Co-ordinates for Generating Three-Dimensional Meshes"; Int. J. Numerical Methods Eng., Vol. 8, No 1, 1974, 27-43.

[12] GORDON, W.J., HALL, C.A.; "Construction of Curvilinear Co-ordinates Systems and Applications to Mesh Generation"; Int. J. Numerical Methods Eng., Vol. 7, No 4, 1973, 461-477.

[13] HABER, R.B., SHEPHARD, M.S., ABEL, J.F., GALLAGHER, R.H. and GREENBERG, D.P.; "A General Two-Dimensional Finite Element Preprocessor Utilizing Discrete Transfinite Mappings"; Int. J. Numerical Methods Eng., Vol. 17, No 7, 1981, 1015-1044.

[14] YERRY, M.A., SHEPHARD, M.S.; "A Modified Quadtree Approach to Finite Element Mesh Generation"; IEEE Computer Grafics Appl., Vol. 3, No 1, 1983, 39-46.

[15] YERRY, M.A., SHEPHARD, M.S.; "Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique"; Int. J. Numerical Methods Eng., Vol. 20, 1984, 1965-1990.

[16] KELA, A., SAXENA, M., PERUCCHIO, R.; "A Hierarchical Structure for Automatic Meshing and Adaptative FEM Analysis"; Eng. Comput., Vol. 4, 1987, 104-111.

[17] THACKER, W.C.; "A Brief Review of Techniques for Generating Irregular Computational Grids"; Int. J. Numerical Methods Eng., Vol. 15, 1980, 1335-1341.

[18] HO-LE, K.; "Finite Element Mesh Generation Methods: a Review and Classification"; Computer Aided Desing, Vol. 20, No 1, 1988, 27-38.

[19] McGIRR, M.B., KRAUKLIS, P.; "The Automatic Generation of Arrays of Nodes with Varying Density"; Computational Techniques and Aplications, CTAC-83, 1984, 229-235.

[20] FUKUDA, J., SUHARA, J.; "Automatic Mesh Generation for Finite Element Analysis"; em *Advance in Computational Methods in Structural Mechanics and Designs* (J.T.Oden,R.W.Clough and Y.Yamamoto eds.), UAH Press, Huntsville, 1972.

[21] LO, S.H.; "A New Mesh Generation Scheme for Arbitrary Planar Domains"; Int. J. Numerical Methods Eng., Vol. 21, 1985, 1403-1426.

AGRADECIMENTOS

Os autores agradecem a Fundação do Banco do Brasil pelo auxílio outorgado ao Projeto RIP-PROCOM onde este pregrama está inserido.