

MAPEAMENTO TRANSFINITO TRIDIMENSIONAL

Antonio C. de O. Miranda* e Luiz F. Martha*

* Departamento de Engenharia Civil e
Grupo de Tecnologia em Computação Gráfica (TeCGraf)
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
22453-900 - Rio de Janeiro - RJ - Brazil
e-mail: amiranda, lfm@civ.puc-rio.br, web page: <http://www.tecgraf.puc-rio.br>

Keywords: Mesh generation, 3D transfinite mapping .

Abstract. *This paper describes an algorithm for generating structured three-dimensional finite element meshes using transfinite mapping. This method performs a cubic spline blending of discrete cross-section meshes that are arbitrarily placed in space. Corresponding nodes along the cross-section meshes are identified and cubic spline curves are generated interpolating these nodes. The solid mesh is generated by creating nodes along the interpolating spline curves. The cross-section meshes must be equivalent in topology, but may have different size, shape, and orientation. One technique for automatic recognition of nodes and elements on the input cross-section meshes is presented. This technique is important because it provides the independency of the algorithm with respect to the data structure of the pre-processor used for generating the cross-section meshes.*

1 INTRODUÇÃO

Técnicas de construção de malhas de elementos finitos podem ser classificadas como malhas estruturadas e malhas não-estruturadas. Malhas estruturadas são aquelas que apresentam uma ordenação espacial uniforme com um arranjo topológico pré-determinado. Por outro lado, as malhas não-estruturadas têm um ocupação espacial arbitrária. Dentro da construção de malhas estruturadas existem as técnicas de mapeamento que possuem diversas aplicações na modelagem de superfícies e sólidos e na geração de malhas de elementos finitos. Uma das técnicas de mapeamento muito utilizada é o *sweep*. Nesta técnica, a geração de superfícies ou sólidos é feita através do “arrasto” de uma seção transversal bidimensional (ou contorno no caso da geração de superfícies) ao longo de uma trajetória no espaço. O *sweep* mais simples é o translacional, também chamado de *extrusão*, em que o “arrasto” da área é feito ao longo de uma linha reta. De modo idêntico, o *sweep* rotacional é definido pela rotação de uma seção transversal em torno de um eixo. O *sweep* mais genérico pode ser definido por mudanças de tamanho, forma e orientação da seção transversal combinadas com um “arrasto” ao longo de uma trajetória tridimensional [1].

No trabalho de Gomes Coelho [2] são apresentadas algumas técnicas de construção de malhas de superfícies que podem ser criadas a partir de curvas espaciais, incluindo o *sweep*. No contexto do trabalho de Gomes Coelho, o *sweep* para superfícies é definido por duas curvas: uma define as seções transversais em cada passo e outra a trajetória de arrasto. A idéia deste tipo de *sweep* pode ser estendida para a geração de sólidos.

Um outro tipo mais genérico de geração de malhas de superfícies utilizam as superfícies de Gordon [3] que interpolam um conjunto de curvas em duas direções. Um tipo de abordagem semelhante foi utilizado para gerar malhas sólidas por interpolação entre seções transversais, conforme abordado por Fonseca [4], ou anteriormente por Peruchio [5]. A interpolação entre seções transversais utiliza uma *spline* cúbica passando por cada ponto correspondente das seções. Gordon e Hall [6] definem esse tipo de construção de malha sólida como Mapeamento Transfinito Tridimensional (*MTT*) e neste trabalho também será tratado como tal.

Este trabalho apresenta um procedimento de construção automática de malhas sólidas de elementos finitos através de mapeamento transfinito tridimensional, partindo-se de uma seqüência de seções transversais discretizadas que podem ter tamanho, forma e orientação distintas, mas que sejam topologicamente idênticas. Apresenta-se um algoritmo capaz de fazer o reconhecimento automático dos pontos e elementos entre as seções transversais, de modo a torná-lo independente das estruturas de dados de pré-processadores e de como as seções transversais são criadas. A interpolação entre seções transversais é feita através de curvas *spline* cúbica.

Havendo seções transversais topologicamente idênticas, o problema principal da construção da malha sólida consiste na identificação e relacionamento dos pontos em cada seção transversal para que seja aplicada a técnica de mapeamento transfinito tridimensional. A solução desenvolvida neste trabalho foi feita introduzindo um ponto base em cada seção transversal, pertencente a esta, que permite o relacionamento dos pontos restantes. A idéia básica do algoritmo é criar listas de arestas dos elementos de superfície da seção transversal, a

partir do ponto base, guardando os índices inicial e final dos pontos (nós da seção transversal). Se as listas de arestas forem construídas de modo idêntico para todas as seções transversais, é possível fazer a correlação dos pontos entre seções quando as listas de arestas forem percorridas de forma linear.

Os elementos das seções transversais devem ser triangulares ou quadrilaterais, gerando, respectivamente, elementos pentaedrais e hexaedrais.

2 DESCRIÇÃO DO ALGORITMO

Em muitas aplicações de modelagem são conhecidas seqüências de seções transversais do objeto que se quer modelar. Este aspecto motiva a criação de procedimentos de construção do modelo a partir de suas seções transversais. Esta seção apresenta um procedimento de construção de malhas sólidas através de *MTT* partindo-se de uma seqüência de superfícies discretizadas que podem ter tamanho, forma e orientação distintas, mas que devem ser topologicamente idênticas. O termo “topologicamente idêntica” significa que as seções devem apresentar uma mesma ordenação espacial e ter o mesmo número de elementos, pontos, arestas internas e externas.

A construção da malha sólida se resume em: dado um conjunto de seções transversais discretizadas, criar seções transversais intermediárias e depois relacioná-las para a geração dos elementos sólidos. Entre cada par de seções transversais fornecidas pode-se ou não gerar um número específico de seções transversais intermediárias.

2.1 Reconhecimento dos Pontos das Seções Transversais

Nesta seção descreve-se o algoritmo para fazer o relacionamento entre os pontos das seções transversais. Os seguintes dados são indispensáveis ao algoritmo (vide Figura 1):

- Uma série de seções transversais que não se interceptam, discretizadas em elementos triangulares ou quadrilaterais.
- Pontos bases conhecidos em cada seção transversal no qual uma curva *spline* irá percorrer. O ponto base deve se situar no contorno da seção transversal.

O primeiro passo do algoritmo consiste em orientar todas as seções transversais em uma mesma direção com respeito à curva *spline*.

A idéia básica do algoritmo é criar *listas de arestas* guardando os índices inicial e final dos pontos (nós) das arestas para cada seção transversal. Se as *listas de arestas* forem construídas de modo idêntico para todas as seções transversais, é possível fazer a correlação dos pontos entre seções quando as *listas de arestas* forem percorridas de forma linear.

A construção de cada *lista de arestas* segue os seguintes passos: (a) identificar o elemento que inicia a construção da *lista de arestas*; (b) adicionar à *lista de arestas* as arestas do elemento inicial; e (c) construir o restante da *lista de arestas* percorrendo os elementos adjacentes ao elemento inicial, seguindo recursivamente para os outros elementos adjacentes.

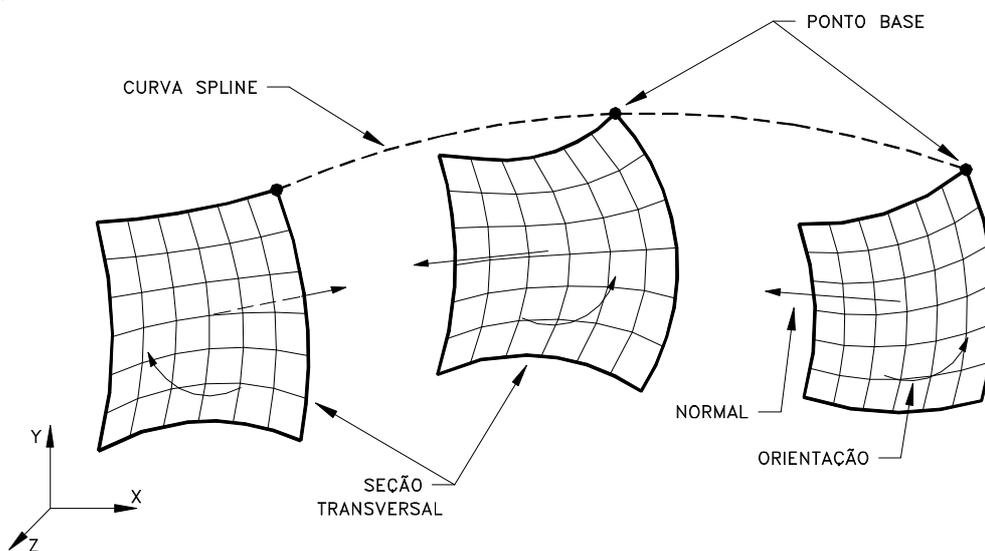


Figura 1 - Seções transversais de um modelo sólido.

2.1.1 Escolha do Elemento Inicial

A partir do ponto base é feita a escolha de um elemento que inicia a construção da *lista de arestas*. Quando o ponto base possui apenas um elemento adjacente, esse elemento é escolhido automaticamente. Caso o ponto base tenha mais de um elemento adjacente escolhe-se o elemento que possui uma de suas arestas pertencentes ao contorno da seção com o ponto base identificado como inicial (“i”) nessa aresta. A Figura 2 ilustra um exemplo para a escolha do elemento inicial. Há três elementos adjacentes ao ponto base: “a”, “b” e “c”. Dois elementos têm arestas de contorno (“a” e “c”). Apenas o elemento “c” tem o ponto base identificado como ponto inicial na aresta de contorno do elemento. O elemento “c” é então escolhido como inicial.

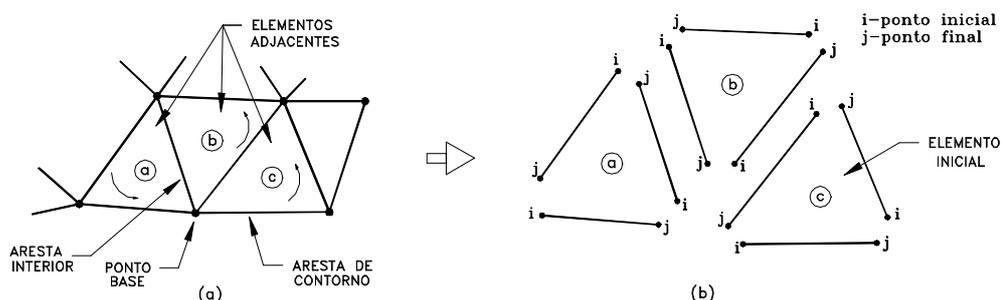


Figura 2 - Escolha do elemento inicial para lista de arestas.

Em seções transversais que são topologicamente idênticas, mas com os pontos base identificados em posições incoerentes geometricamente, é possível que os elementos gerados

sejam inválidos. Neste trabalho é feito um teste para diminuir a possibilidade de elementos inválidos comparando-se o número de elementos adjacentes ao ponto base em cada seção transversal. Caso estes sejam diferentes o processo não segue. A Figura 3 ilustra esse caso. Mesmo considerando seções transversais idênticas topologicamente, os elementos gerados segundo o *MTT* podem ser inválidos geometricamente (auto-interseção, por exemplo). Tais situações deverão ser identificadas com testes *a posteriori*.

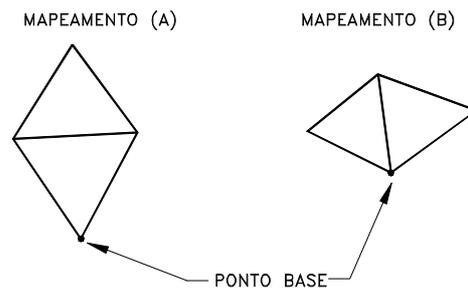


Figura 3 - Pontos base selecionados em posições incoerentes.

2.1.2 Construção da Lista de Arestas

A partir do elemento inicial escolhido em cada seção transversal faz-se a construção da *lista de arestas* para cada seção. A *lista de arestas* é inicializada com as arestas do elemento inicial, segundo a conectividade deste elemento. A aresta que fica no topo da lista deve pertencer ao contorno e possuir o ponto base identificado como inicial. Esta escolha garante a validade topológica dos elementos gerados, conforme descrito mais adiante. A Figura 4 ilustra a construção de uma *lista de arestas* para um elemento inicial.

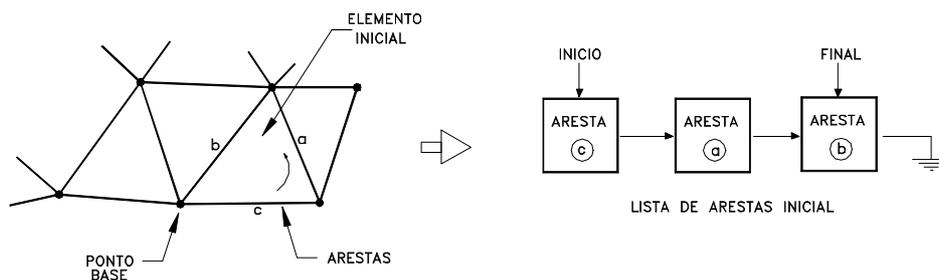


Figura 4 - Lista de arestas para o elemento inicial.

As arestas incluídas na lista são “marcadas” para evitar repetições. As inserções seguintes na *lista de arestas* obedecem os seguintes passos:

- (a) A aresta no topo da lista se torna a aresta corrente.

- (b) O elemento adjacente à aresta corrente é o próximo a ser visitado.
- (c) As arestas adjacentes não marcadas por elementos já visitados são incluídas no final da *lista de arestas*.
- (d) A aresta corrente passa a ser a próxima aresta na *lista de arestas* e o processo é retomado a partir de (b). Quando todos os elementos tiverem sido visitados, e todas as arestas incluídas na *lista de arestas*, o processo termina.

O processo é repetido para todas as seções transversais produzindo assim um conjunto de *listas de arestas* construídas de modo idêntico.

A Figura 5 ilustra uma seção discretizada em quatro elementos quadrilaterais. O ponto base é o ponto identificado como 7. O elemento 3 é o inicial. A seqüência de arestas obtida segundo os passos descritos é a seguinte: *k, i, f, h, l, j, g, d, a, c, e e b*.

Com a seqüência de arestas obtida para todas as seções transversais, identificando também se a aresta é de contorno ou interna, é possível fazer um teste de consistência entre seções. Esse teste é feito comparando o número de arestas de contorno e internas entre as seções transversais. Caso não ocorra igualdade entre os números de arestas significa que as seções não são topologicamente idênticas, não sendo possível continuar com o algoritmo. As arestas do contorno de uma malha de seção transversal são identificadas por terem somente um elemento adjacente. As arestas do interior têm dois elementos adjacentes.

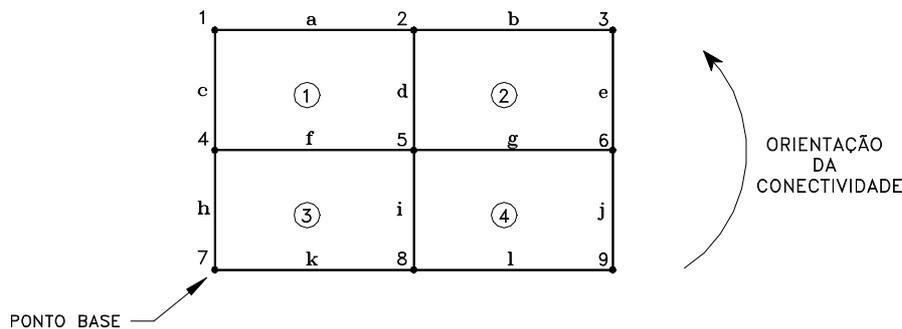


Figura 5 - Exemplo para construção da lista de arestas.

2.1.3 Obtenção dos Pontos a partir da Lista de Arestas

Os índices dos vértices de cada aresta descrita na seção anterior contêm a informação topológica necessária à obtenção da seqüência de pontos de cada seção transversal. A seqüência de pontos pode ser obtida percorrendo-se a *lista de arestas* do início ao fim, pesquisando-se os pontos da aresta corrente que ainda não foram inseridos em um vetor de pontos (ou em outra estrutura que permita armazenar os pontos seqüencialmente). No fim do processo tem-se uma seqüência de pontos para uma determinada seção transversal. Para o exemplo da Figura 5 tem-se a seguinte seqüência de pontos: 7, 8, 5, 4, 9, 6, 2, 1 e 3.

Para todas as seções transversais são obtidos vetores de pontos que são seqüências de pontos montados a partir das listas de arestas. Desta forma, a correlação entre os pontos em cada seção transversal é dada pelo índice no vetor.

2.2 Interpolação

Cada conjunto de vértices identificados pelo índice define uma trajetória poligonal usada para interpolar uma curva *spline* cúbica por trechos [2,7]. Esta representação de curva contínua permite calcular os pontos das seções transversais intermediárias. Esta formulação também permite que se subdivida as curvas da forma desejada, com trechos igualmente espaçados ou com uma razão qualquer definida *a priori*.

3 IMPLEMENTAÇÃO E EXEMPLOS

O algoritmo anteriormente descrito foi incorporado em uma biblioteca computacional que inclui alguns outros algoritmos geradores de malhas desenvolvidos por outros trabalhos. Os objetivos no desenvolvimento desta biblioteca foram:

- Disponibilizar os algoritmos para programas que necessitem de malhas de elementos finitos;
- Tornar os algoritmos independentes dos programas de pré-processamento;
- Permitir a programação dos pré-processadores sem preocupação com os algoritmos específicos;
- Padronizar os argumentos de entrada e saída dos algoritmos.

A Figura 6 mostra a descrição da função que gera a malhas de elementos finitos por *MTT*. Os argumentos destas funções são apresentadas na forma de um pseudo-código similar à linguagem C. A descrição de outras funções que geram malhas de elementos finitos bidimensionais e tridimensionais encontra-se em outro trabalho do mesmo autor [8].

Msh3DMapp (nome da função)

Argumentos de entrada:

```
Int ns,                /* Número de superfícies (seções transversais) */
Int *ids,              /* Vetor com os índices dos pontos bases em cada */
                      /* Superfície */
Int *div,              /* Vetor com o número de superfícies entre */
                      /* superfícies principais */
Int np_sup,            /* Número de pontos em uma superfície */
Double **pts_sup,     /* Matrix (ns x np_sup*3) com as coordenadas dos */
                      /* pontos das superfícies */
Int ne_sup,            /* Número de elementos em uma superfície */
Int **conn_sup,       /* Matrix (ns x ...) com as conectividades dos */
                      /* elementos das superfícies */
```

Argumentos de saída:

```
Int *n_node,          /* Número de nós gerados */
Double **coords,     /* Coordenadas dos nós da malha */
Int *n_elem,         /* Número de elementos gerados */
Int **conn           /* Conectividade dos elementos */
```

Figura 6 - Pseudo-código da função que gera a malha por *MTT*.

Com vista a implementar a função que gera malhas, adotou-se um pré-processador para modelagem tridimensional de elementos finitos que tem demonstrado ao longo dos anos (inclusive remodelado a partir de vários testes-pilotos e de uso em simulações reais) ser capaz de atender eficientemente grande parte das necessidades da modelagem tridimensional: o MG [9].

O MG é um gerador de malhas para modelos tridimensionais de elementos finitos implementado por Gomes Coelho [2] e proveniente do Convênio PUC-Petrobras. Neste trabalho, o MG foi estendido para incorporar a geração de malhas sólidas de elementos finitos.

O MG foi desenvolvido usando linguagem C padrão ANSI seguindo uma disciplina de programação orientada a objetos. O programa utiliza as bibliotecas gráficas G3D e V3D [10] e IUP/LED, um *toolkit* portátil para interface com o usuário [11]. O modelo de interface utilizado no MG é resultado de um trabalho de avaliação feito com testes-pilotos de modelagem, discutidas por Gomes Coelho e de Souza [12]. Detalhes sobre as técnicas desenvolvidas e uso do MG encontram-se no trabalho de Gomes Coelho [2] e no manual do MG [9].

No ambiente do MG, as seções transversais devem estar ligadas por uma curva que permite identificar os pontos bases de cada seção transversal e o número de seções intermediárias entre seções. Os pontos bases são os pontos onde as curvas e seções transversais se interceptam. O número de seções intermediárias entre duas seções transversais é o número de divisões dos trechos da curva que liga essas seções. O reconhecimento da ordem das seções transversais é feito de modo automático, sem que haja necessidade do usuário informar sequencialmente as seções.

A Figura 7 ilustra um gancho construído por mapeamento. A Figura 8 mostra as seções transversais. O modelo tem uma seção transversal inicial circular que muda longitudinalmente de tamanho e forma. A seção circular inicial se torna menor e depois muda para uma seção retangular. Essa seção retangular segue constante e na ponta do gancho a seção transversal torna-se quadrada. Tem-se no total 10 seções transversais que estão ligadas por curvas que definem as seções transversais intermediárias e os pontos bases.

A Figura 9 mostra um modelo de corpo de prova retangular muito utilizado em livros de Resistência dos Materiais e Teoria da Elasticidade para exemplificar zonas de concentração de tensões. O modelo é constituído de uma barra prismática onde há uma diminuição brusca da seção transversal no meio da barra, e é aplicada uma força de tração nas extremidades. As superfícies utilizadas na construção do modelo estão mostradas na Figura 10: são três superfícies construídas com mapeamento transfinito bilinear [13]. As extremidades da barra foram construídas com *extrusão* e o afunilamento da seção transversal através do *mapeamento transfinito tridimensional*. A Figura 11 mostra os elementos gerados internamente. Nesse modelo foram criados apenas elementos hexaédricos (*brick*).

A Figura 12 mostra o modelo cortado longitudinalmente com os campos de tensões normais à seção transversal representados graficamente por faixa de cores. É possível observar a alta concentração de tensões, representado em vermelho, estando aparentemente de acordo com a literatura da Teoria da Elasticidade.

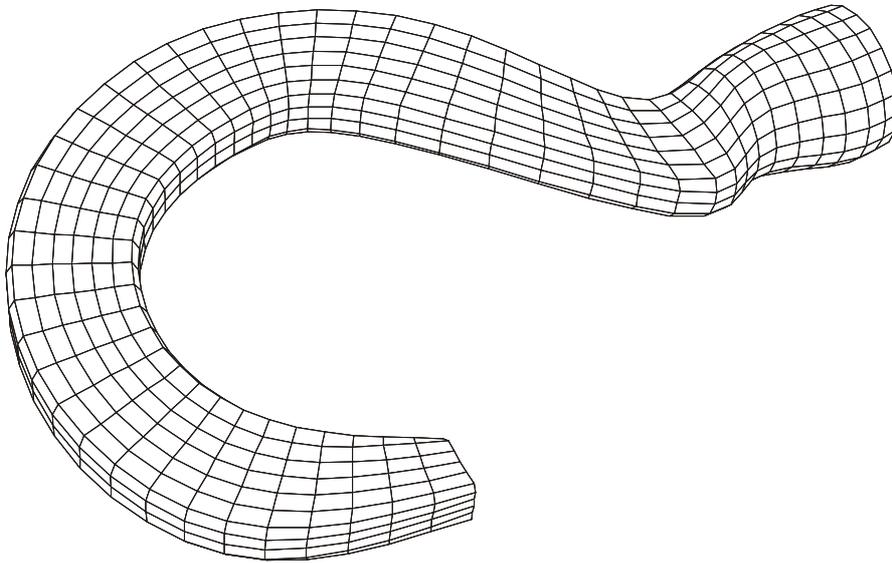


Figura 7 - Gancho construído por mapeamento transfinite tridimensional.

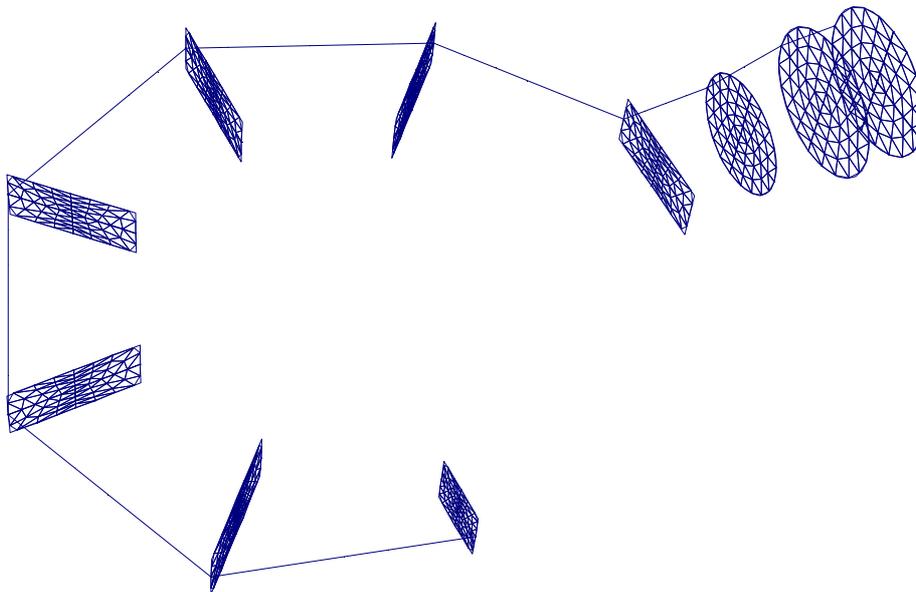


Figura 8 - Seções transversais do gancho.

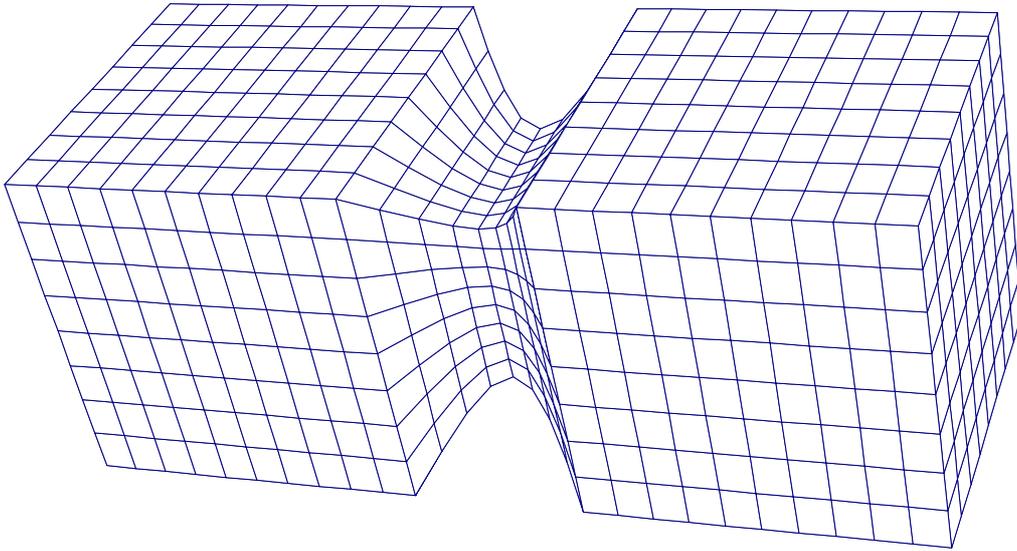


Figura 9 - Modelo de um corpo-de-prova.

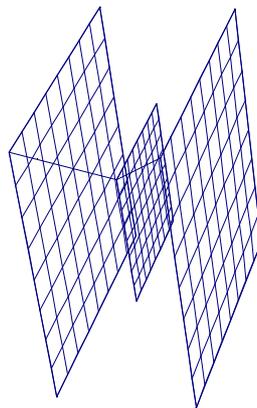


Figura 10 - Superfícies bases para o corpo-de-prova.

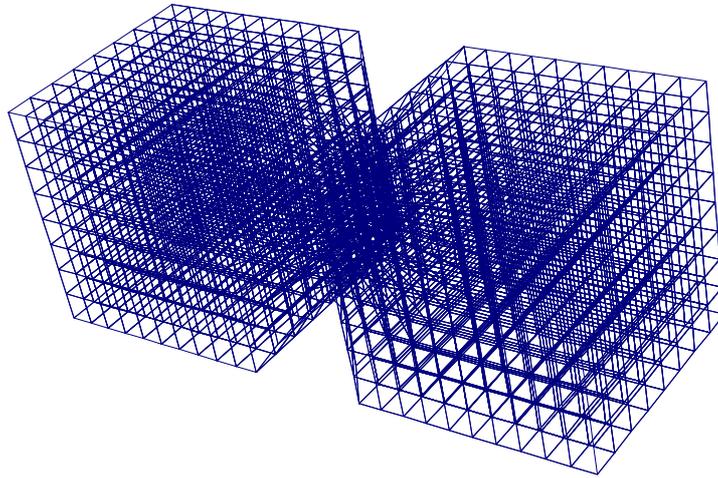


Figura 11 - Malhas geradas para o corpo-de-prova.

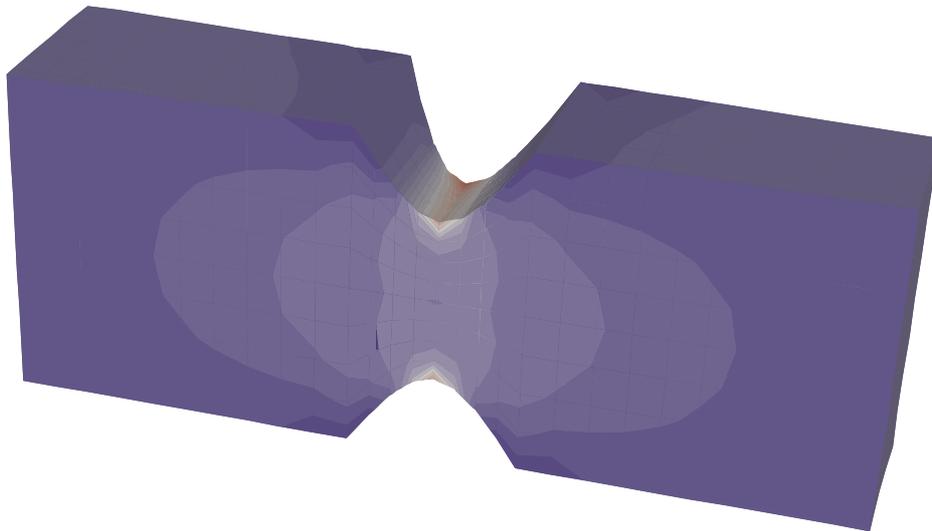


Figura 12 - Campos de tensões normais à seção transversal do corpo-de-prova.

4 CONCLUSÃO

Foi descrito um procedimento de construção automática de malhas sólidas de elementos finitos através de mapeamento transfinito tridimensional partindo-se de uma seqüência de seções transversais discretizadas que podem ter tamanho, forma e orientação distintas, mas que sejam topologicamente idênticas. Apresenta-se um algoritmo capaz de fazer o reconhecimento automático dos pontos e elementos entre as seções transversais, de modo a torná-lo independente das estruturas de dados de pré-processadores e de como as seções transversais são criadas. A interpolação entre seções transversais é feita através de curvas *spline* cúbicas.

O problema da identificação e relacionamento dos pontos em cada seção transversal foi resolvido introduzindo em cada seção transversal um ponto base que permite fazer o relacionamento dos pontos restantes. A partir dos pontos bases são criadas listas de arestas, que guardam os índices inicial e final de cada ponto, em cada seção transversal e que torna possível fazer a correlação dos pontos entre seções transversais.

O algoritmo foi incorporado em uma biblioteca computacional de algoritmos de geração de malhas de elementos finitos e utilizado em um pré-processador tridimensional. Dois exemplos são apresentados para mostrar as potencialidades do algoritmo.

O algoritmo que faz o reconhecimento das malhas das seções transversais independe de como estas malhas são criadas, desde que as seções sejam topologicamente equivalentes. Uma consequência do algoritmo, já que trabalha com topologia, é a possibilidade de se trabalhar com seções transversais não planas. Entretanto, ainda é necessário desenvolver testes para verificar a validade dos elementos gerados pois as curvas *splines* podem sofrer auto-interseção e os elementos podem resultar distorcidos.

REFERÊNCIAS

- [1] Hearn, D. e Baker, M. P. - *Computer Graphics - C version*, 2nd. Edition, Prentice Hall, Inc. , 1997.
- [2] Gomes Coelho, L. C. - “Modelagem de Cascas com Interseções Paramétricas”, Dissertação de Doutorado, Departamento de Informática, PUC-Rio, 1998.
- [3] Farin, G. - “Curves and Surfaces for Computer Aided Geometric Design”, *Academic Press*, 2nd. Edition, ISBN: 0-12-249051, pp 464, 1990.
- [4] Fonseca, G L. - “Editor Gráfico de Malhas Transfinitas Tridimensionais para Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC-Rio, 1989.
- [5] Peruchio, R.; Ingraffea, A. R e Abel, J. F. - “Iterative Computer Preprocessing for Three-Dimensional Finite Element Analysis”, *International Journal For Numerical Methods in Enginnering*, vol. 18, pp. 909-926, 1982.
- [6] Gordon, W. J. e Hall, C. A. - “Construction of Curvilinear Co-ordinate Systems and Aplication to Mesh Generation”, *International Journal For Numerical Methods in Enginnering*, vol. 7, pp. 461-477, 1973.
- [7] T.A. Foley and G.M. Nielson - “Knot Selection for Parametric Spline Interpolation”, *Mathematical Methods in CAGD*, L. Schumaker, Academic Press, pp. 445-467, 1989.

- [8] Miranda, A. C. de O. - “Integração de Algoritmos de Geração de Malhas de Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC-Rio, 1999.
- [9] MG, Manual do Usuário, versão 3.0, PUC-Rio, 1996.
- [10] L. F. Martha, B. Castier e M. Gattass, “Uma Taxonomia para Manipulação Interativa e Visualização de Objetos 3D”, *Anais do VII SIBGRAPI, VII Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, Curitiba – PR – Brasil, pp. 149-156, 1994.
- [11] Levy, C. H. - “IUP/LED: Uma Ferramenta Portátil de Interface com Usuário”, Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 1993.
- [12] Gomes Coelho, L. C. E de Souza, C. S. - “Comunicação de Problemas e Soluções Geométricas em uma Interface 3D”, *Anais do VIII SIBGRAFI*, pp 233-240, 1995.
- [13] Haber, R.; Shephard, M. S.; Greenberg, D. P.; Abel, J. F. and Gallagher, R. H. - “A General Two-dimensional, Graphical Finite Element Preprocessor”, *International Journal For Numerical Methods in Engineering*, vol.17, pp. 1015-1044, 1981.