

# UMA BIBLIOTECA COMPUTACIONAL PARA GERAÇÃO DE MALHAS BIDIMENSIONAIS E TRIDIMENSIONAIS DE ELEMENTOS FINITOS

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@tecgraf.puc-rio.br, web page: <http://www.tecgraf.puc-rio.br>

**Keywords:** Finite Element, Mesh generation, Independent API.

**Abstract.** *This paper describes a computational library for finite element mesh generation algorithms that have been developed in the Applied Computer Graphics research group of the Department of Civil Engineering and Tecgraf (Computer Graphics Technology Group) of PUC-Rio. This library presents a consistent Application Program Interface (API) for all the algorithms. It includes two and three-dimensional algorithms of structured and non-structured mesh generation techniques that resulted from a research line of more than ten years of development. The API was devised so that the algorithms are independent of any particular data structure of the client modeler. As a consequence the meshing algorithms may be used in a large number of modelers. In addition, new algorithms may be incorporated in a consistent manner.*

## 1 INTRODUÇÃO

A geração de malhas de elementos finitos consiste, de uma forma geral, dividir um determinado domínio de interesse em sub-domínios. Domínios bidimensionais podem ser divididos em quadriláteros ou triângulos, enquanto que, domínios tridimensionais podem ser divididos em tetraedros, pentaedros ou hexaedros. Essa divisão é feita com a aplicação de um método de geração de malhas. Diversos algoritmos de geração foram desenvolvidos, e ainda são desenvolvidos, para atender a diversas finalidades. Com o crescimento do número de algoritmos torna-se necessário que estes tenham uma interface comum, permitindo assim uma padronização dos dados de entrada e saída. Uma forma de conseguir essa padronização é criar uma biblioteca computacional de geração bidimensional e tridimensional de malhas.

Uma biblioteca computacional neste contexto tem como finalidade integrar os algoritmos de geração de malhas em uma API (*Application Program Interface*) consistente entre os algoritmos e os modeladores de malhas de elementos finitos. Desta forma, cria-se uma biblioteca com algoritmos de geração de malhas que proporciona a incorporação dos algoritmos em diversos modeladores, principalmente tridimensionais, necessitando “apenas” o desenvolvimento de interface com o usuário para coleta de dados necessário.

Este trabalho descreve uma biblioteca computacional que integra os algoritmos de geração de malhas de elementos finitos desenvolvidos no decorrer de anos da linha de pesquisa de Computação Gráfica Aplicada do Departamento de Engenharia Civil e do Tecgraf/PUC-Rio. Os algoritmos considerados são bidimensionais e tridimensionais. Os algoritmos bidimensionais geram malhas estruturadas através de mapeamento transfinito linear, bilinear e trilinear, e também malhas não-estruturadas com utilização de três procedimentos distintos de triangulação arbitrária. Os algoritmos tridimensionais geram malhas estruturadas com as técnicas de *sweep* e mapeamento transfinito tridimensional. Malhas volumétricas não-estruturadas de tetraedros são geradas com utilização de um algoritmo de avanço da fronteira (contração de contorno).

O trabalho está dividido em histórico, descrição da biblioteca, exemplos de aplicação da biblioteca, conclusões e referências utilizadas no trabalho. No histórico descreve-se os programas de análise de elementos finitos e os trabalhos de geração de malhas da linha de pesquisa. Na parte de descrição da biblioteca apresenta-se os objetivos da criação da biblioteca, as funções que a compõem, formatos dos dados de entrada/saída e exemplo da API. Na seção de exemplos de aplicação são citados alguns programas que utilizam a biblioteca. Nas conclusões são feitas as considerações finais sobre o trabalho.

## 2 HISTÓRICO

A linha de pesquisa da Computação Gráfica Aplicada do Departamento de Engenharia Civil e do Tecgraf da PUC-Rio desenvolve há vários anos programas de análise de elementos finitos (FEMOOP<sup>1</sup>), juntamente com pré-processadores gráficos bidimensionais (MTool<sup>2</sup>) e tridimensionais (MG<sup>3</sup>), e pós-processadores bidimensionais (MView<sup>4</sup>) e tridimensionais (POS3D<sup>5</sup>).

Faz-se aqui um breve histórico da linha de pesquisa na área de geração de malhas. O trabalho de Paulino (1988)<sup>6</sup> apresenta o desenvolvimento, implementação e funcionamento de um programa interativo para geração de dados de modelos de elementos finitos de estruturas reticuladas espaciais. Também expõe uma proposta de metodologia para elaboração da arquitetura do programa, os ambientes de desenvolvimento, interface e estrutura de dados. Na mesma filosofia de trabalho de Paulino, Fonseca<sup>7</sup> apresenta um pré-processador de malhas transfinitas tridimensionais de elementos finitos que são criadas a partir de seções transversais do modelo.

O trabalho de Campos (1991)<sup>8</sup> aborda a geração de malhas de elementos finitos bidimensionais baseada em uma estrutura de dados topológica, consistindo de um pré-processador gráfico interativo para discretização numérica de modelos bidimensionais. O trabalho de Vianna (1992)<sup>9</sup>, que aborda a modelagem geométrica estendida para modelos bidimensionais, juntamente com o trabalho de Guimarães (1992)<sup>10</sup> compõem um sistema integrado para problemas de elasticidade plana. Vianna também aborda um gerador automático de malhas bidimensionais baseado na técnica de avanço de fronteira.

Um outro trabalho, de Cavalcante Neto (1994)<sup>11</sup>, descreve um ambiente computacional gráfico interativo para análise integrada auto-adaptativa de modelos bidimensionais de elementos finitos, com aplicação em problemas estruturais mecânicos, baseado em uma técnica de enumeração espacial recursiva. Um gerador de malhas bidimensionais foi desenvolvido, combinando a técnica de *quadtree*<sup>12</sup> e a técnica de triangulação por contração do contorno. O algoritmo gera os elementos no interior da região por *quadtree* e na faixa na fronteira da região por triangulação.

Em um trabalho mais recente dos autores, um outro algoritmo de geração de malhas bidimensionais é apresentado baseado na técnica de enumeração espacial recursiva e contração de contorno, mas que incorpora novas idéias desenvolvidas por Cavalcante Neto (1998)<sup>13</sup> em um algoritmo tridimensional de malhas não-estruturadas. Testes foram feitos com outros algoritmos anteriores descrevendo tempo e qualidade de forma do elemento e se demonstra que este algoritmo apresenta melhores resultados tanto no tempo quanto na forma (1999)<sup>14</sup>.

No que se refere à modelagem tridimensional, como mencionado, o trabalho de Fonseca (1989)<sup>3</sup> descreve um pré-processador gráfico interativo para geração de modelos de elementos finitos sólidos através do mapeamento transfinito tridimensional. Gomes Coelho (1998)<sup>15</sup> desenvolveu em seu trabalho o MG (*Mesh Generator*): um modelador de malhas de superfícies com geometria arbitrária, com enfoque na interseção entre retalhos de malhas de superfície distintas. Cavalcante Neto (1998)<sup>13</sup> descreve uma técnica de geração de malhas volumétricas de elementos tetraédricos para domínios tridimensionais arbitrários.

Com o objetivo de escrever a Biblioteca Computacional de Geração de malhas, os autores reescreveram o algoritmo de Fonseca<sup>3</sup>, tornando-o independente do pré-processador. Para isso nossos métodos de construção da malhas foram desenvolvidos (1999)<sup>16</sup>. A mesma idéia foi introduzida na modelagem de sólidos através da técnica de *sweep* onde é feita o “arrasto” de uma seção transversal ao longo de uma trajetória no espaço (1999)<sup>17</sup>.

### 3 DESCRIÇÃO DA BIBLIOTECA

Os objetivos no desenvolvimento desta biblioteca foram:

- (a) disponibilizar os algoritmos para programas que necessitem de geração de malhas de elemento finitos;
- (b) tornar os algoritmos independentes dos programas de pré-processamento;
- (c) permitir a programação dos pré-processadores sem preocupação com os algoritmos específicos;
- (d) padronizar os argumentos de entrada e saída dos algoritmos.

Os argumentos de entrada e saída dos pontos foram padronizados seguindo uma representação por um vetor, como mostra a Figura 1(a). As coordenadas bidimensionais seguem em duplas  $(x,y)$  e os tridimensionais em seguem triplas  $(x,y,z)$ . A conectividade dos elementos é representada por um vetor, Figura 1(b), onde o primeiro argumento representa o numero de nós ( $nn$ ) do elemento e os argumentos seguintes indicam os índices dos nós, no vetor de pontos, que representa o elemento. E assim seqüencialmente para todos os elementos. Apresenta-se um exemplo dos argumentos de entrada e saída na Figura 2, com um pseudo-código similar à linguagem C.

(a) Vetor de coordenadas dos pontos:

$x_0$	$y_0$	$x_1$	$y_1$	$x_2$	$y_2$	$\dots$	$x_{n-1}$	$y_{n-1}$
\	/	\	/	\	/		\	/
(ponto 0)		(ponto 1)		(ponto 2)			(ponto n-1)	

(b) Vetor de conectividade dos elementos:

$nn_1$	$id_1$	$id_2$	$id_3$	$id_4$	$id_5$	$id_6$	$nn_2$	$id_1$	$id_2$	$id_3$	$\dots$
\	\	/	/	/	/	/	\	\	\	\	
Número de nós	conectividade do elemento					número de nós	conectividade do elemento				

---

Figura 1 - Vetores de coordenadas e conectividade.

<b>Msh2DShape</b>			
<b>Argumentos de entrada:</b>			
int n_loops,	/*	número de circuitos	*/
int *loop_segs,	/*	vetor com o número de segmentos em cada	*/
	/*	circuito	*/
double *bdry_pts,	/*	vetor de coordenadas do contorno	*/
int type_mesh,	/*	Tipo de elemento: T3 (3) ou T6 (6)	*/
<b>Argumentos de saída:</b>			
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 2 - Exemplo dos argumentos de entrada e saída de uma função de geração de malha.

A seguir faz-se uma descrição das funções que geram as malhas de elementos finitos. A Figura 3 mostra de forma resumida as funções contidas na biblioteca.

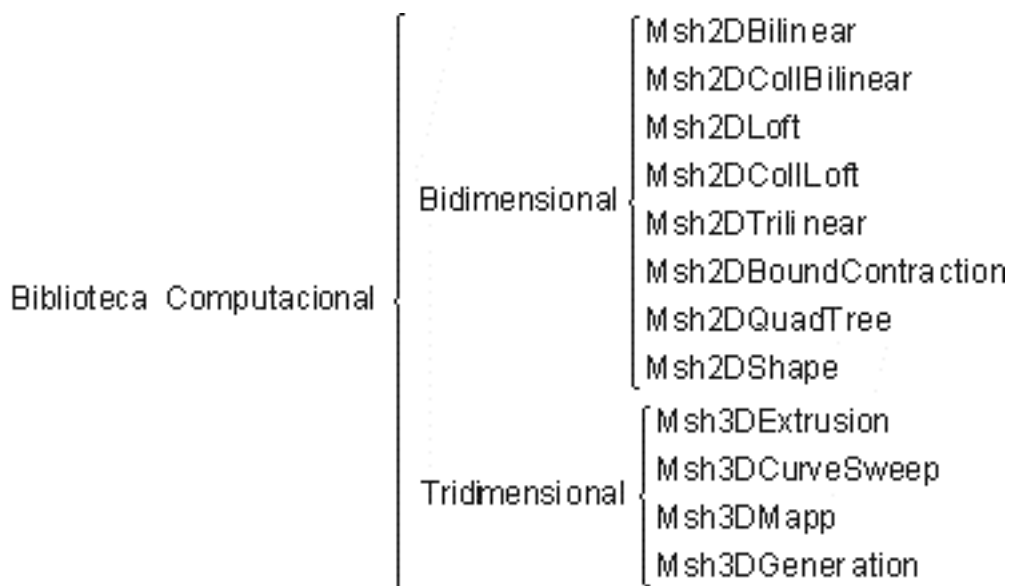


Figura 3 - Resumo da Biblioteca Computacional.

### 3.1 Geração de Malhas Bidimensionais

#### **Msh2DBilinear**

---

Mapeamento transfinito bilinear para regiões quadrilaterais com qualquer forma para os lados<sup>8, 18</sup>. O número de segmentos em lados opostos devem ser iguais.

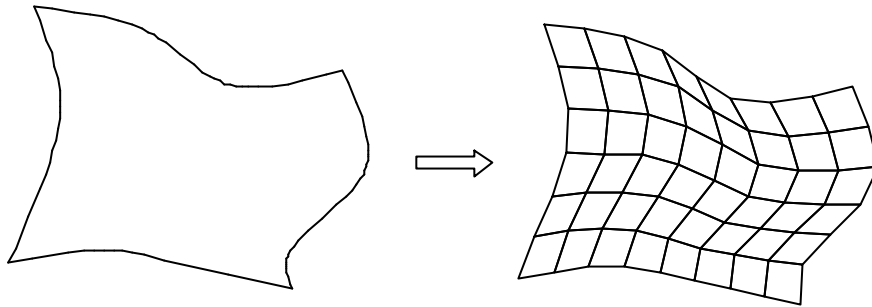


Figura 4 – Exemplo de geração de malha utilizando Msh2DBilinear.

#### **Msh2DCollBilinear**

---

Mapeamento transfinito bilinear para uma região triangular que conceitualmente é formada por uma região quadrilateral com um dos lados colapsado para um ponto<sup>8, 18</sup>. Os três lados podem ter qualquer forma e os lados adjacentes ao lado colapsado devem ter o mesmo número de segmentos.

#### **Msh2DLoft**

---

Mapeamento transfinito linear entre dois lados opostos de uma região quadrilateral<sup>8, 18</sup>. Os outros dois lados devem ser linhas retas. O número de segmentos nos lados opostos devem ser iguais.

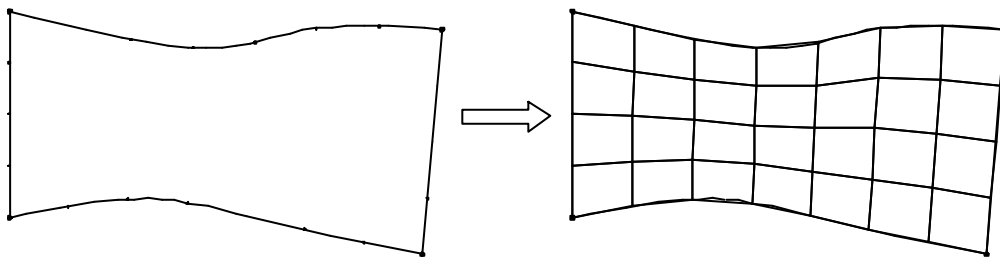


Figura 5 – Exemplo de geração de malha utilizando Msh2DLoft.

#### **Msh2DCollLoft**

---

Mapeamento transfinito linear entre um ponto e uma curva, que forma uma região triangular<sup>8, 18</sup>. Os dois lados da região que são adjacentes ao ponto de *lofting* tem de ser linhas retas e ter o mesmo número de pontos.

### **Msh2DTrilinear**

---

Mapeamento transfinito trilinear para uma região triangular com qualquer forma para os lados<sup>8, 18</sup>. O número de segmentos em todos os lados devem ser iguais.

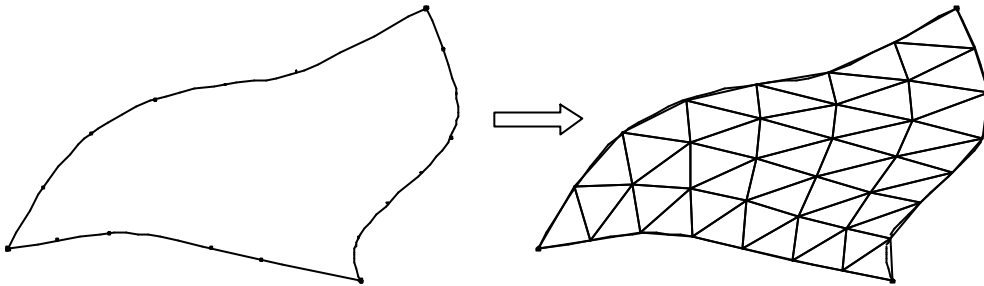


Figura 6 – Exemplo de geração de malha utilizando Msh2DTrilinear.

### **Msh2DBoundContraction**

---

Triangulação genérica aplicável a contornos fechados sem auto-interseção<sup>9</sup>. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. A triangulação é feita usando uma técnica de contração de contorno. Os pontos interiores da triangulação podem ser gerados pelo algoritmo e/ou fornecidos. Dois procedimentos são utilizados para gerar os pontos interiores. O primeiro procedimento utiliza-se de uma estrutura *quadtree* para gerar pontos no centro de cada célula e depois faz-se uma contração do contorno. O outro procedimento gera os pontos interiores em uma grade regular.

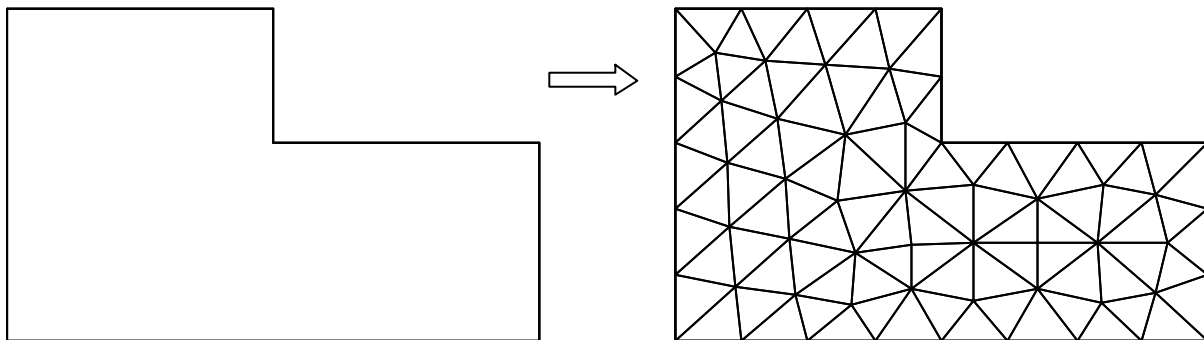


Figura 7 – Exemplo de geração de malha utilizando Msh2DBoundContraction.

### **Msh2DQuadTree**

---

Triangulação genérica aplicável a contornos fechados sem auto-interseção<sup>11</sup>. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. Para elementos Q4 e Q8, o número de segmentos (lados de elementos) deve ser par

em cada circuito. O algoritmo utiliza-se de uma estrutura *quadtree* e constrói os elementos do interior do domínio utilizando padrões de elementos para cada configuração da *quadtree*; para a faixa do contorno gera os elementos por uma técnica de contração de contorno.

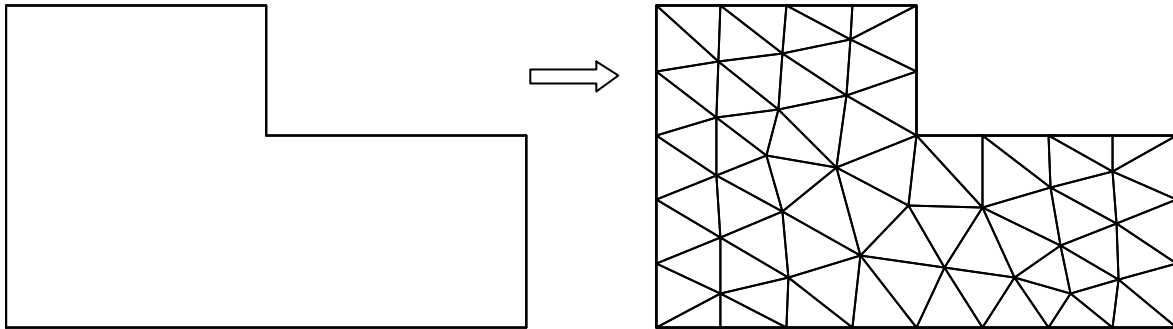


Figura 8 – Exemplo de geração de malha utilizando Msh2DQuadTree.

### **Msh2DShape**

Triangulação genérica aplicável a contornos fechados sem auto-interseção<sup>14</sup>. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. O algoritmo utiliza uma estrutura de *quadtree* para dar uma gradação de transição na geração dos nós interiores. Os pontos do domínio são gerados conjuntamente com o avanço da fronteira e de acordo com o gradiente dado pela árvore *quadtree*. O algoritmo contém adicionalmente um procedimento de melhoria local da malha.

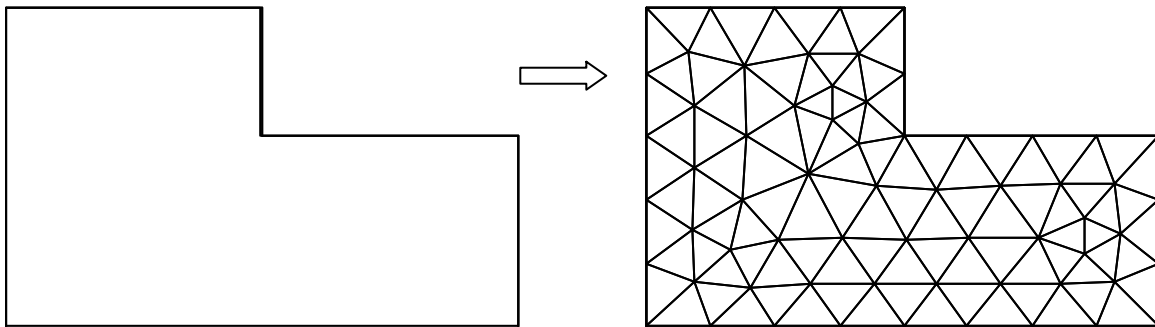


Figura 9 – Exemplo de geração de malha utilizando Msh2DShape.



### 3.2 Geração de Malhas Tridimensionais

#### Msh3DExtrusion

---

*Sweep* de uma seção transversal ao longo de um segmento reto<sup>17</sup>. A superfície geradora deve ter elementos triangulares ou quadrilaterais.

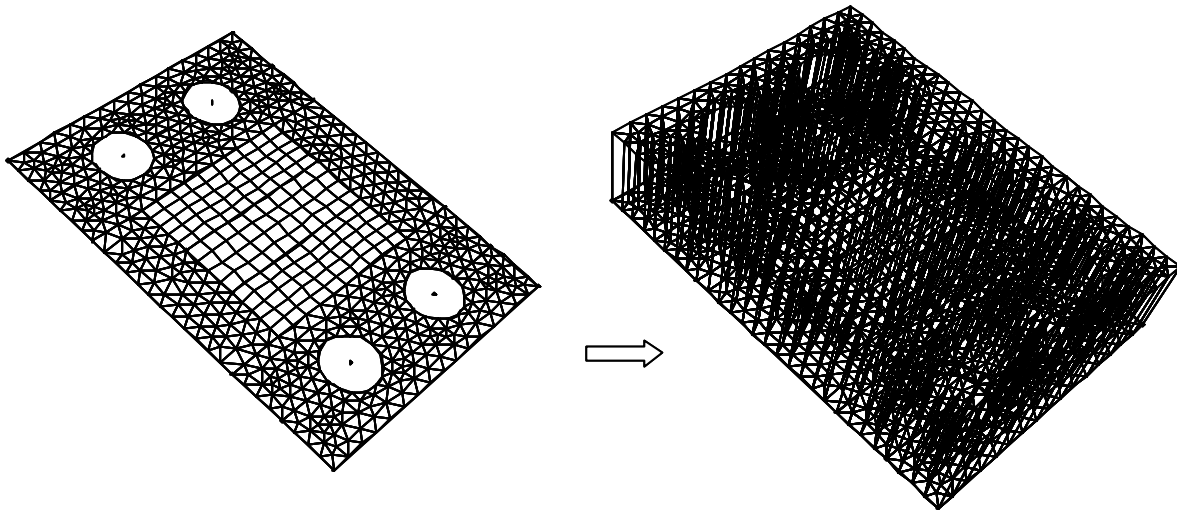


Figura 10 – Exemplo de geração de malha utilizando Msh3DExtrusion.

#### Msh3DCurveSweep

---

*Sweep* de uma seção transversal ao longo de uma curva no espaço<sup>17</sup>. Os vetores  $u$ ,  $v$  e  $w$  dever estar normalizados e ortogonais entre si. O vetores tangentes a trajetória do *sweep*,  $w_i$ , devem seguir a mesmo sentido da trajetória do *sweep*.

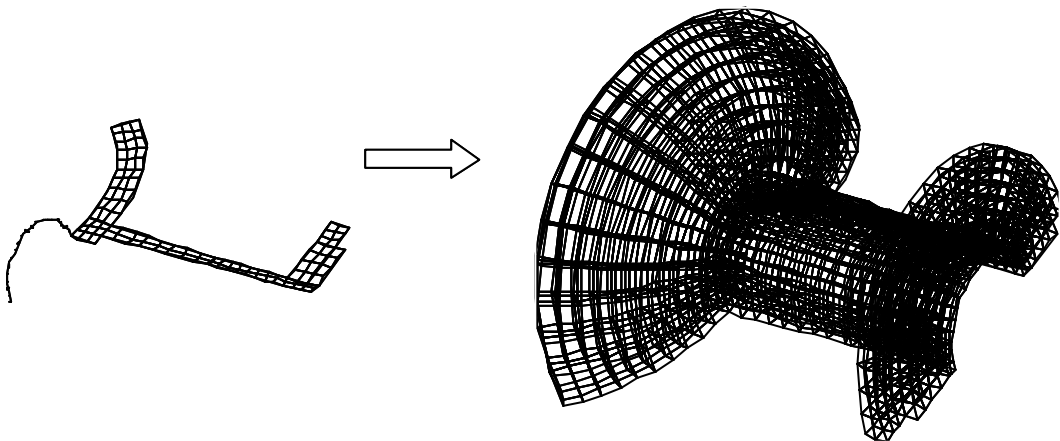


Figura 11 – Exemplo de geração de malha utilizando Msh3DCurveSweep.

### **Msh3DMapp**

Mapeamento transfinito tridimensional a partir de superfícies que representam as seções transversais de um modelo<sup>16</sup>. As seções transversais são interpoladas com uso de uma curva *spline* cúbica. As seções transversais devem ter elementos triangulares e quadrilaterais e serem topologicamente idênticas. Os pontos bases são os pontos conhecidos em cada seção transversal onde uma curva *spline* irá percorrer. O ponto base deve situar-se no contorno da seção transversal.

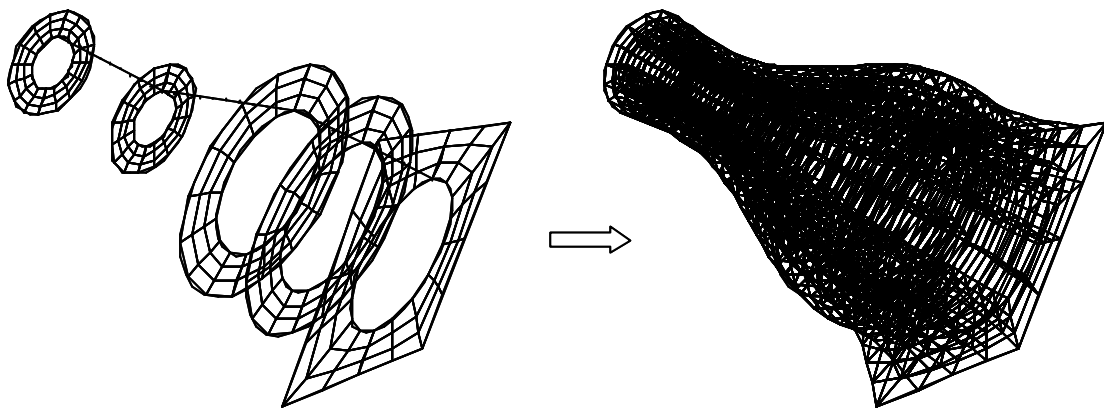


Figura 12 – Exemplo de geração de malha utilizando Msh3DMapp.

### **Msh3DGeneration**

Algoritmo para geração de malhas volumétricas não-estruturadas de tetraedros para domínios arbitrários<sup>13</sup>. A entrada de dados deve ser feita por representação de fronteira, por faces triangulares. O modelo deve estar fechado e as orientações das normais das faces da fronteira devem apontar para dentro do modelo

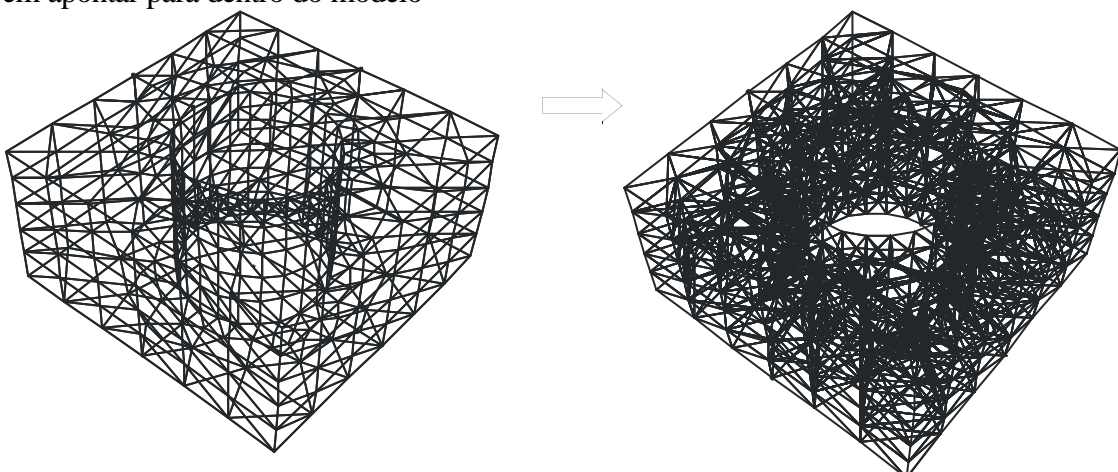


Figura 13 – Exemplo de geração de malha utilizando Msh3DGeneration.

### 3.3 Funções Auxiliares

#### **Msh2DFreeNodes e Msh3DFreeNodes**

---

Libera a memória do vetor de coordenadas dos pontos passada como argumento de saída pelos algoritmos bidimensionais e tridimensionais, respectivamente.

#### **Msh2DFreeConn e Msh3DFreeConn**

---

Libera a memória do vetor de conectividade dos elementos passada como argumento de saída pelos algoritmos bidimensionais e tridimensionais, respectivamente.

## 4 EXEMPLOS DE USO DA BIBLIOTECA

Esta seção apresenta alguns exemplos de programas que se utilizaram da biblioteca computacional. Como mencionado anteriormente, há apenas a necessidade de obtenção dos dados de entrada para as funções e em seguida é retornada uma malha. Desta forma não somente pré-processadores de elementos finitos utilizam as funções. Os seguintes programas utilizam-se das funções de geração de malhas:

- MTool<sup>2</sup> (Mesh Tool) - Programa completo para análise bidimensional de elementos finitos.
- Quebra2D<sup>19</sup> - Programa bidimensional para análise de fadiga e propagação de trincas.
- MG<sup>3</sup> (Mesh Generator) - Pré-processador tridimensional de malhas de elementos finitos.
- FRANC3D<sup>20</sup>, “Fracture Analysis Code 3D” - Programa tridimensional de análise de propagação de trincas.
- RECON<sup>21</sup> - Simulador do processo reverso de seções geológicas.
- MM<sup>22</sup>, “Multi-Mesher” - Programa para geração de malhas tridimensionais para geologia estrutural.

## 5 CONCLUSÃO

Este trabalho descreve uma biblioteca computacional de geração de malhas de elementos finitos que inclui algoritmos desenvolvidos pela linha de pesquisa da Computação Gráfica Aplicada do Departamento de Engenharia Civil e do Tecgraf/PUC-Rio, em uma API genérica. Estão incluídos algoritmos de geração de malhas estruturas e não-estruturadas bidimensionais e tridimensionais. Apresenta também alguns exemplos de aplicação da biblioteca em programas de análise e pré-processamento.

Os argumentos de entrada e saída são padronizados em vetores, existentes em qualquer linguagem de programação não utilizando assim tipos específicos, que torna a idéia aplicável em outras linguagens. Também, não há vínculos entre a geração da malha e as estruturas de dados existentes nos pré-processadores. Logo a primeira consequência da biblioteca é que o usuário programador pode trocar uma função por outra, de acordo com sua finalidade, sem a necessidade de mudança nos argumentos de entrada e saída. Outra consequência é que permite incluir novos

algoritmos de geração de malhas na biblioteca, que é importante para continuidade da linha de pesquisa. Apesar que a biblioteca ter sua principal utilização em elementos finitos, ela pode ser utilizada em qualquer programa que necessite uma discretização do domínio. A principal consequência deste trabalho é disponibilizar de imediato uma ferramenta para geração de malhas com muitas facilidades, como já mencionado no corpo do trabalho.

Um fato importante neste trabalho é que ele só foi possível com a continuidade de trabalhos anteriores dentro da linha de pesquisa. Isso fica demonstrado na seção histórico deste trabalho que começa com algoritmos simples, como mapeamento transfinito bidimensional, até algoritmos mais complicados como geração de malhas genéricas tridimensionais. Portanto, de fato os autores deste trabalho são todos aqueles que contribuíram com os seus algoritmos para a biblioteca computacional de geração de malhas.

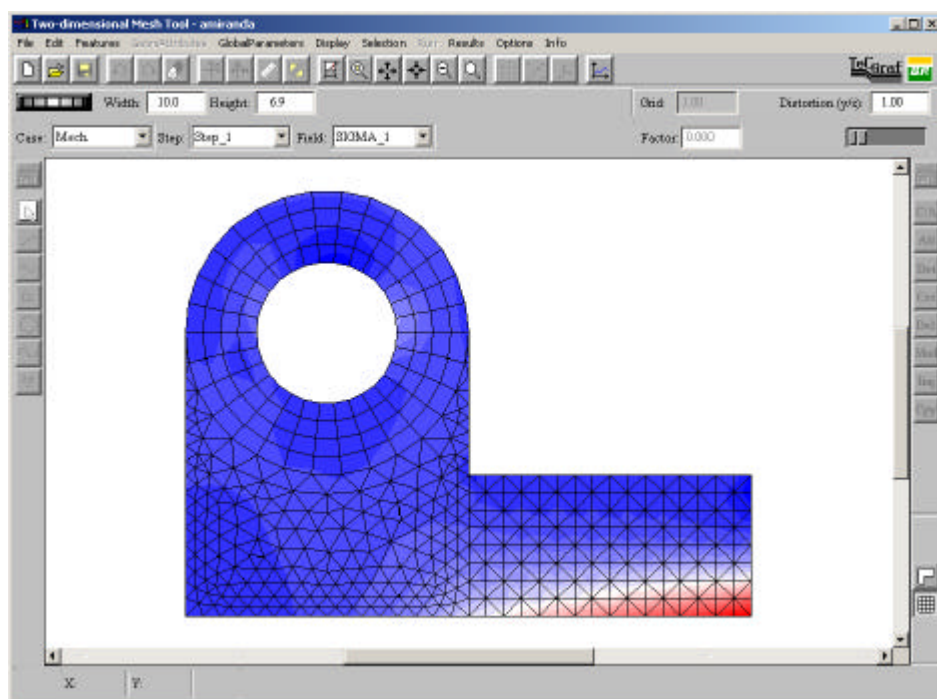


Figura 14 – Exemplo do uso da biblioteca no MTool.

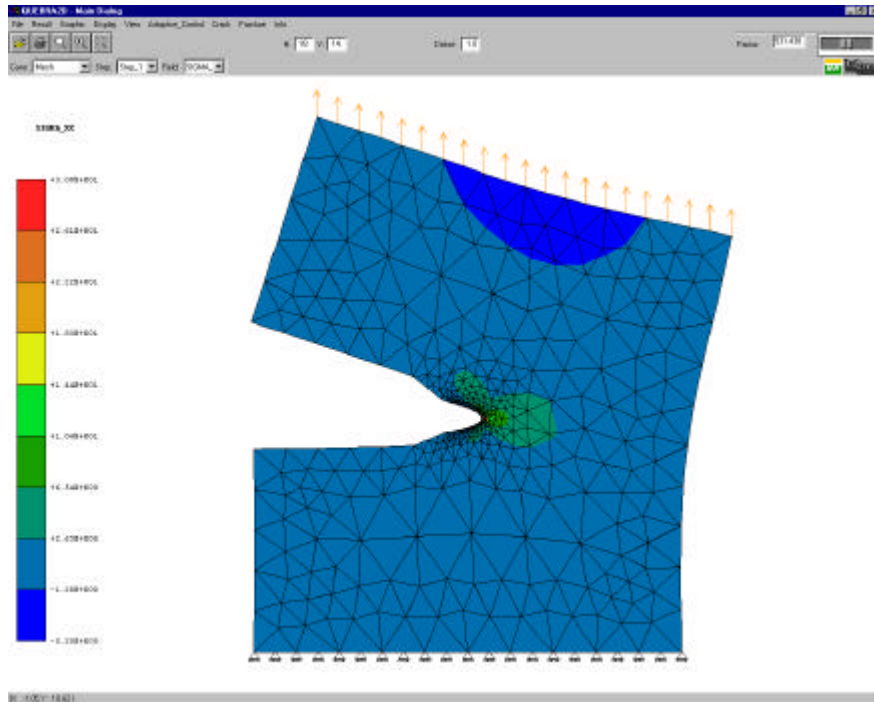


Figura 15 – Exemplo do uso da biblioteca no Quebra2D.\

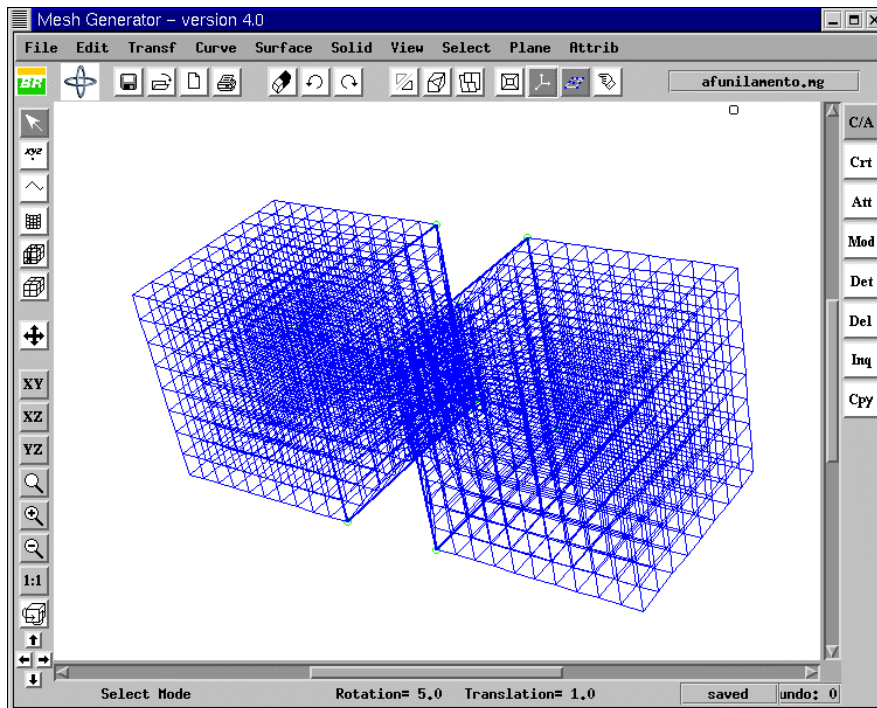


Figura 16 – Exemplo do uso da biblioteca no MG.

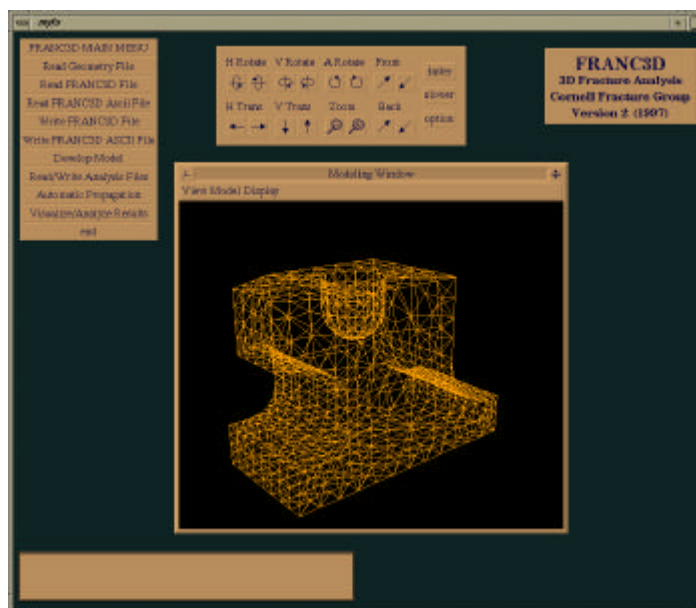


Figura 17 – Exemplo do uso da biblioteca no FRANC3D.

## REFERÊNCIAS

- [1] Martha, L.F.; Menezes, I.F.M.; Lages, E.N.; Parente Jr., E.; Pitangueira, R.L.S., “An OOP Class Organization for Materially Nonlinear Finite Element Analysis”, *Join Conference of Italian Group of Computational Mechanics and Ibero-Latin American Association of Computational Methods in Engineering, Padua, Italia, Set. 1996*, pp. 229-232, 1996.
- [2] MTool - *Bidimensional Mesh Tool* - Manual do Usuário, PUC-Rio, 1992.
- [3] MG – *Mesh Generation* , Manual do Usuário, versão 3.0, PUC-Rio, 1996.
- [4] MVIEW - *Bidimensional Mesh View*, (Versão 3.0) - PUC-Rio, 1997.
- [5] Celes Filho, W.; Martha, L. F. e Gattass, M. - “An Efficient Data Structure for Solid Finite Element Analysis Post-Processor (in Portuguese)”, *XI Brazilian Congress of Mechanic Engineering*, pp. 105-108, 1991.
- [6] Paulino, G. H. - “Pré-Processamento de Estruturas Reticulares Espaciais”, *Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1988*.
- [7] Fonseca, G L. - “Editor Gráfico de Malhas Transfinitas Tridimensionais para Elementos Finitos”, *Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1989*.
- [8] Campos, J. A. P. - “Geração de Malhas de Elementos Finitos Bidimensionais Baseada em uma Estrutura de Dados Topológica”, *Dissertação de Mestrado, Departamento de Engenharia Civil, 1991*.
- [9] Vianna, A. C. - “Modelagem Geométrica Estendida para Modelos Bidimensionais de Elementos Finitos”, *Dissertação de Mestrado, Departamento de Engenharia Civil, 1992*.

- [10] Guimarães, L. G. S. - “Disciplina de Programação Orientada a Objetos para Análise e Visualização Bidimensional de Modelos de Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1992.
- [11] Cavalcante Neto, J. B. - “Simulação Auto-Adaptativa Baseada em Enumeração Espacial Recursiva de Modelos Bidimensionais de Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1994.
- [12] Baehmann, P. L.; Wittchen, S. L.; Shephard, M. S.; *et al* - “Robust Geometrically Based, Automatic Two-Dimensional Mesh Generation”, *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 1043-1078, 1987.
- [13] Cavalcante Neto, J. B. - “Geração de Malha e Estimativa de Erro para Modelos Tridimensionais de Elementos Finitos com Trincas”, Dissertação de Doutorado, Departamento de Engenharia Civil, PUC/Rio, 1998.
- [14] Miranda, A.C. de O.; Cavalcante Neto, J.B.; Martha, L.F., “An Algorithm for Two-dimensional Mesh Generation for Arbitrary Regions with Cracks”, *SIBGRAPI'99 – XII Brazilian Symposium on Computer Graphics, Image Processing and Vision, UNICAMP/SBC, Campinas, SP, Out. 1999*, IEEE Computer Society Order Number PRO0481, ISBN 0-7695-0481-7, Eds.: J. Stolfi & C. Tozzi, pp. 29-38, 1999.
- [15] Gomes Coelho, L. C. - “Modelagem de Cascas com Interseções Paramétricas”, Dissertação de Doutorado, Departamento de Informática, PUC/Rio, 1998.
- [16] Miranda, A.C. de O.; Martha, L.F., “*Mapeamento Transfinito Tridimensional*”, 20th Iberian Latin-American Congress on Computational Methods in Engineering, 1999.
- [17] 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.
- [18] 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.
- [19] Araújo, T.D.P.; Cavalcante Neto, J.B.; Carvalho, M. T. M.; Bittencourt, T.N.; Martha, L.F., “Adaptive Simulation of Fracture Processes Based on Spatial Enumeration Techniques”, *International Journal of Rock Mechanics and Mining Sciences*, vol. 34, no. 3/4, p. 551 (abstract) and CD-rom (full paper), 1997.
- [20] Carter, B. J.; Wawrzynek, P. A.; Ingraffea, A. R.- “Automatic 3-D Growth Simulation”, *International Journal For Numerical Methods in Engineering*, vol. 47, pp. 229-253, 2000.
- [21] Martha, L.F.; Carvalho, P.C.P.; Bender, A.A., “A Topological Data Structure as a Framework for Balancing of Geological Cross-Sections”, *Proceedings of the 8th International Conference of the International Association for Computer Methods and Advances in Geomechanics*, Edited by H.J. Siriwardane, A.A. Balkema, pp. 477-482, 1994.
- [22] Cavalcanti, P. R. and U. T. Mello, 1999, Three-dimensional constrained Delaunay Triangulation: a minimalist approach, 8th International Meshing Roundtable Proceedings, p. 119- 129.