

# INTERFACE GRÁFICA PARA PROGRAMA DE ELEMENTOS FINITOS

**Ricardo Afonso Angélico<sup>1</sup>, Rodrigo Bresciani Canto<sup>2</sup>, Jonas de Carvalho<sup>3</sup>.**

Universidade de São Paulo - Escola de Engenharia de São Carlos - Depto de Engenharia Mecânica.

Av. Trabalhador Sãocarlense, 400 - Cx. P. 359 - CEP 13566-590 - São Carlos - SP, Brasil -

<sup>(1)</sup>ricardoangelico@hotmail.com, <sup>(2)</sup>canto@sc.usp.br, <sup>(3)</sup>prjonas@sc.usp.br.

**Volnei Tita**

Universidade de São Paulo - Escola de Engenharia de São Carlos - Depto de Engenharia de Estruturas. Av. Trabalhador Sãocarlense, 400 - Cx. P. 359 - CEP 13566-590 - São Carlos - SP, Brasil - voltita@sc.usp.br.

**Neilor Cesar dos Santos**

Centro Federal de Educação Tecnológica da Paraíba – CEFETPB - Depto de Mecânica. Av. Primeiro de Maio, 720 - CEP: 58015-430 - João Pessoa - PB, Brasil - neilor@cefetpb.edu.br

**Resumo.** Este trabalho apresenta o desenvolvimento de uma interface gráfica a ser aplicada a um programa baseado no método dos elementos finitos. Como objetivo principal, busca-se implementar essa interface gráfica para a visualização do modelo a ser solucionado (etapa de pré-processamento) bem como dos resultados obtidos (etapa de pós-processamento). A interface está configurada para elementos planos (quadriláteros e triangulares). A mesma permite, na fase de pré-processamento, a visualização do modelo geométrico, ou seja, dos nós e da malha de elementos e, além disso, das condições de contorno como: restrições e carregamentos aplicados. Na fase de pós-processamento, a interface permite a visualização da malha deformada e dos gradientes de deslocamento, tensão e deformação. Os resultados apresentados pela interface gráfica são comparados com os resultados visualizados através do pré e pós-processador do programa comercial ANSYS®.

**Palavras-chave:** Interface Gráfica, Elementos Finitos, Pré-Processador, Pós-Processador

## 1. INTRODUÇÃO

A década de 50 do século XX foi o ponto de partida para a computação gráfica e as tecnologias CAD (“Computer Aided Design”), CAE (“Computer Aided Engineering”) e CAM (“Computer Aided Manufacturing”) (Anacleto, 1991 e Ferreira, 1990). Na década de 70, trabalhos publicados por Lockheed em 1973 e em 1975 demonstraram a viabilidade econômica da utilização dos sistemas CAD no desenvolvimento de projetos mecânicos (Anacleto, 1991 e Ferreira, 1990). Da mesma forma, os sistemas CAE são ferramentas úteis no auxílio desses projetos.

Os sistemas CAE englobam dentre outros, os programas computacionais com base no Método dos Elementos Finitos (MEF). O MEF é um dos métodos numéricos mais utilizados para a solução de problemas de engenharia.

Muitos programas baseados no MEF, em geral, apresentavam a entrada e a saída de dados por meio de um arquivo no formato texto. Assim, até o surgimento da computação gráfica, uma das maiores dificuldades era a visualização da malha de elementos finitos, das restrições nodais e da disposição dos carregamentos aplicados. Além disso, as rotinas de cálculo forneciam uma enorme listagem de dados de saída, incluindo campos de deslocamentos, tensões e deformações. Tais

valores eram analisados pelo usuário do programa afim de que o mesmo pudesse executar um projeto com êxito. Porém, o tamanho dos arquivos de saída fazia com que por muitas vezes o usuário cometesse erros durante a sua análise, além de dificultar uma visualização mais global do comportamento mecânico da estrutura.

Sendo assim, com a evolução dos recursos de visualização gráfica, surgiram os chamados pré e pós-processadores gráficos que possibilitam uma interação melhor do ambiente computacional com o usuário.

Segundo Carvalho (1989), a geração gráfica passou a ser utilizada com a finalidade de:

- pré-processar a entrada de dados (pré-processadores): inclui a criação de todo modelo geométrico, geração e visualização da malha de elementos finitos e visualização das restrições nodais e dos carregamentos aplicados;
- pós-processar os resultados calculados (pós-processadores): inclui a visualização gráfica dos resultados.

Dessa forma, um programa baseado no método dos elementos finitos passou a ter basicamente três módulos, conforme mostrado no fluxograma da Fig. (1).

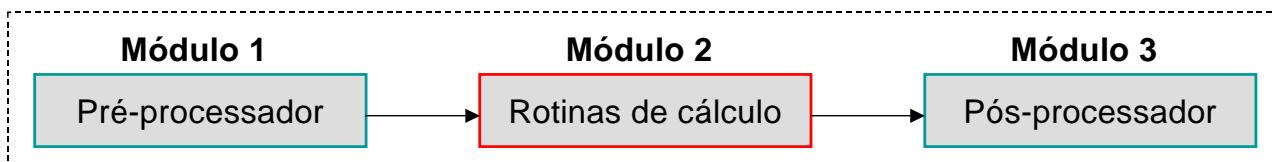


Figura 1 – Fluxograma simplificado da estrutura de um programa de elementos finitos.

As interfaces gráficas foram se aperfeiçoando ao longo do tempo de tal forma que atualmente, as interfaces gráficas são geradas através de uma série de recursos visuais, como por exemplo, bibliotecas de componentes visuais que englobam vários objetos. Esses componentes auxiliam a visão e fazem com que o cérebro possa receber e interpretar informações com referência a tamanho, forma, cor, orientação, movimento e outras características de identificação visual. Os objetos gráficos adotam as formas de janelas, botões, menus de barras horizontais e verticais ou ícones. Esses objetos gráficos são utilizados para apresentar os aplicativos em execução e seqüência de comandos. A representação do sistema pelo emprego de objetos, por ser mais próxima à ótica humana, permite maior facilidade de visualização do funcionamento e do fluxo de informação.

A interface proposta nesse trabalho está sendo desenvolvida utilizando programação orientada a objetos e implementada em C++. Os recursos gráficos são obtidos através da biblioteca OpenGL (Neider, 1993), que vem sendo amplamente empregada em sistemas CAD/CAE atuais. Esta interface é composta de dois módulos: pré e pós-processador.

O módulo de pré-processamento é simplificado, pois não apresenta ferramentas para modelagem e geração de malha, sendo útil somente para a conferência de dados de entrada, como: nós, elementos, condições de contorno, entre outros. Tendo em vista essa limitação, a modelagem e geração de malha podem ser realizadas por um outro programa.

No módulo de pós-processamento, principal objetivo desse trabalho, as principais potencialidades de um pós-processador comercial foram implementadas. Além de proporcionar a exibição dos gráficos de campos de deslocamentos, deformações e tensões, esse módulo possibilita recursos como: listagem de resultados, ferramentas de translação, rotação e zoom, captura de imagens.

## 2. METODOLOGIA

### 2.1. Representação Gráfica na etapa de Pré-processamento

As estruturas em engenharia são geometricamente bem caracterizadas, isso torna mais fácil a operação para criar as entidades primitivas. Por exemplo, para caracterizar um nó de uma estrutura

pode-se exibir um ponto. Para caracterizar elementos, podem-se exibir polígonos definidos pelos nós que constituem o elemento a ser exibido.

Vale ressaltar que, as restrições e os carregamentos devem ser exibidos de forma proporcional ao modelo em estudo, para isso, tomou-se como valor de referência do modelo, a ser discretizado, a diagonal (D) do retângulo envolvendo o mesmo, Fig. (2).

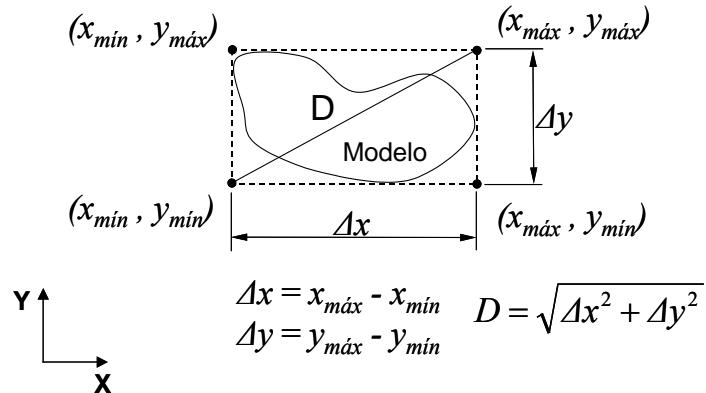


Figura 2 – Esquema da diagonal do retângulo que envolve o modelo discretizado.

As condições de contorno aplicadas aos nós são representadas por triângulos que sugerem as restrições de translação nas direções x e y, Fig. (3).

Inicialmente, deve-se definir uma constante de proporcionalidade R (fator de escala) para que as entidades a serem representadas possuam dimensões coerentes com o modelo.

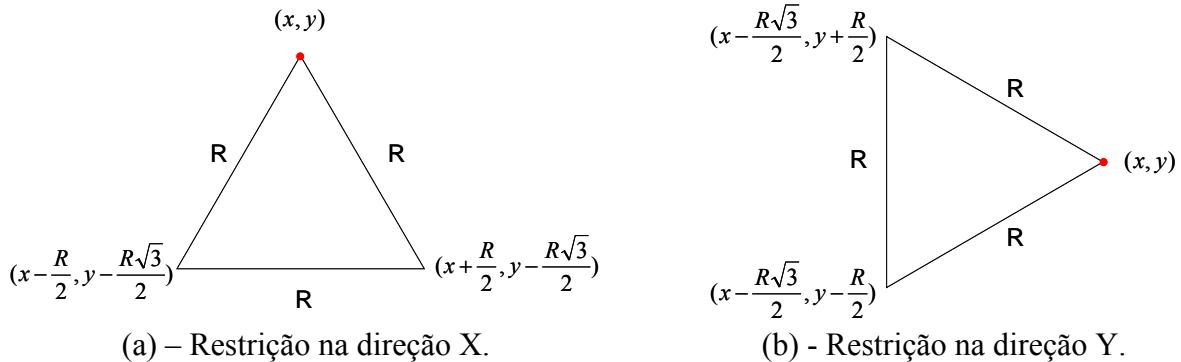


Figura 3 – Representação das restrições aplicadas aos nós em função do valor D.

Os carregamentos são caracterizados por setas. As dimensões dessas setas estão associadas ao valor que a grandeza assume nos nós do elemento, Fig. (4).

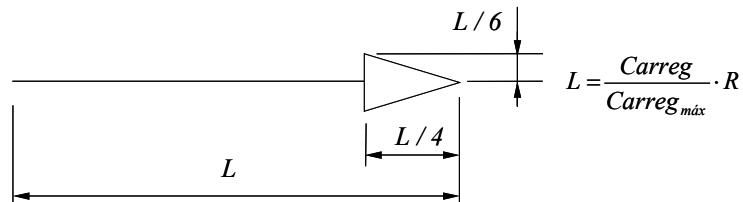


Figura 4 – Esquema da seta para representação de carregamentos.

## 2.2. Representação Gráfica na etapa de Pós-processamento

### 2.2.1. Malha deformada

A visualização da malha deformada está relacionada com os deslocamentos nodais que são obtidos da Eq. (1) através das funções de forma  $N_i$  do elemento (Hinton & Owen, 1979).

$$u^{(e)} = \sum_{i=1}^n N_i^{(e)} x_i^{(e)} \quad \text{e} \quad v^{(e)} = \sum_{i=1}^n N_i^{(e)} y_i^{(e)} \quad (1)$$

Onde:

n: número de nós do elemento

u: deslocamento nodal na direção x

v: deslocamento nodal na direção y

x: coordenada x dos nós

y: coordenada y dos nós

A representação dos valores absolutos dos deslocamentos não é interessante, pois esses deslocamentos podem assumir valores pequenos ou desprezíveis quando comparado às dimensões da estrutura em estudo, dificultando a visualização da estrutura deformada. A fim de criar uma malha deformada que seja perceptível ao usuário, deve-se multiplicar os deslocamentos obtidos (Eq. (1)) por um fator ( $\alpha$ ) e somá-los a posição inicial dos nós dada pela Eq. (2). Assim a posição final dos nós é dada por  $x'$  e  $y'$  conforme a Eq. (3).

$$x' = x^{(e)} + u^{(e)} \cdot \alpha \quad \text{e} \quad y' = y^{(e)} + v^{(e)} \cdot \alpha \quad (3)$$

O fator ( $\alpha$ ) é obtido de uma relação entre o máximo deslocamento absoluto e o valor (D) tomado como referência do modelo (opção padrão da interface) ou a critério do usuário.

### 2.2.2. Gradientes

Para os gradientes de deslocamento, de tensão ou de deformação, utilizam-se cores para representar os valores da grandeza de interesse. Antes de estabelecer como o gradiente é gerado sobre um elemento, é interessante descrever o modelo de cores utilizado pela interface. O modelo de cores utilizado é o RGB (Red – Green – Blue). O modelo é representado pelo cubo da Fig. (5). Nesse modelo as cores são formadas pela ponderação das cores: vermelha, verde e azul como o próprio nome sugere. Esse modelo trata-se de um modelo aditivo, ou seja, uma determinada cor é obtida das contribuições individuais de cada uma das componentes de cores.

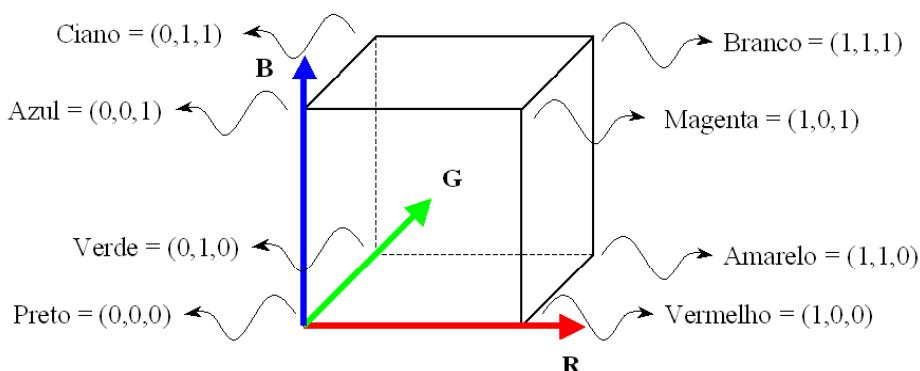


Figura 5 – Modelo de cores RGB (Neider, 1993).

Descreve-se aqui um dos gradientes mais utilizados, o gradiente de nove cores, seguindo o

padrão de programas conhecidos e que auxilia no desenvolvimento da interface. Para classificar os valores nas classes geradas anteriormente devem ser criados valores que limitem uma determinada faixa de cores. Esses valores são determinados através de uma interpolação entre os valores máximo e mínimo da grandeza de interesse. Assim a largura de cada faixa pode ser expressa de acordo com a Eq. (4), sendo GI o valor que representa a grandeza de interesse.

$$\Delta = \frac{GI_{máx} - GI_{mín}}{9} \quad (4)$$

Conhecendo-se a largura de cada faixa e os máximos e mínimos que a estrutura assume no campo em estudo (deslocamento, deformação, tensão) podem-se estabelecer os limites inferiores e superiores para cada faixa de cores conforme a Tab. (1) (Gallagher R. S., 1994).

Os gradientes são gerados elemento a elemento independentemente. O valor da grandeza de cada nó que constitui o elemento pode ser classificado em uma das nove faixas de cores apresentadas na Tab. (1). Os valores das grandezas nos nós do elemento podem ou não assumir diferentes faixas. Isso implica que o elemento pode ser interceptado por isolinhas. As isolinhas de interesse aqui são aquelas que limitam determinada faixa de valores, ou seja, as isolinhas são da forma  $GI_{\min} + i\Delta$ , onde  $i = 1, 2, \dots, 9$ .

Tabela 1 – Classificação dos valores.

<b>Limite inferior</b>	<b>Limite superior</b>	<b>Vermelho</b>	<b>Verde</b>	<b>Azul</b>	<b>Cor</b>
$GI_{min}$	$GI_{min} + \Delta$	0,0	0,0	1,0	Azul
$GI_{min} + \Delta$	$GI_{min} + 2\Delta$	0,0	0,5	1,0	Azul
$GI_{min} + 2\Delta$	$GI_{min} + 3\Delta$	0,0	1,0	1,0	Cyan
$GI_{min} + 3\Delta$	$GI_{min} + 4\Delta$	0,0	1,0	0,5	Green
$GI_{min} + 4\Delta$	$GI_{min} + 5\Delta$	0,0	1,0	0,0	Green
$GI_{min} + 5\Delta$	$GI_{min} + 6\Delta$	0,5	1,0	0,0	Lime
$GI_{min} + 6\Delta$	$GI_{min} + 7\Delta$	1,0	1,0	0,0	Yellow
$GI_{min} + 7\Delta$	$GI_{min} + 8\Delta$	1,0	0,5	0,0	Orange
$GI_{min} + 8\Delta$	$GI_{min} + 9\Delta = GI_{máx}$	1,0	0,0	0,0	Red

Para determinar em qual ponto a isolinha intercepta a aresta de determinado elemento efetua-se uma interpolação entre os pontos da aresta do elemento. Sendo assim, segundo Hinton & Owen (1979), para um elemento linear (2 nós por aresta) a interpolação é linear sobre os nós da aresta do elemento, para um elemento quadrático (3 nós por aresta) a interpolação é de grau dois. A Figura (6) representa a interpolação efetuada para um elemento cuja interpolação sobre suas arestas é linear.

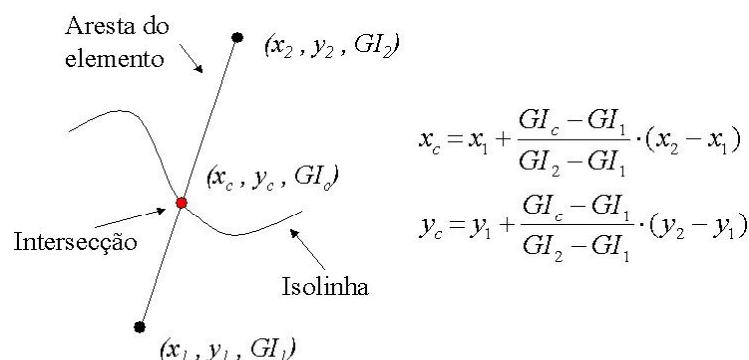
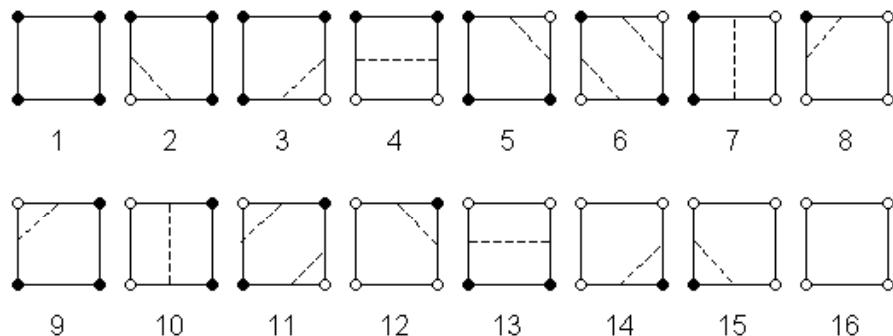


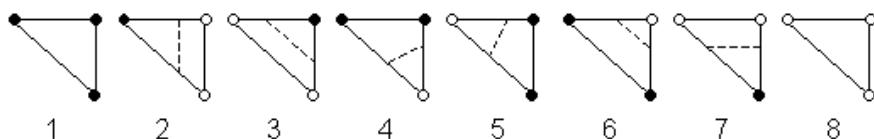
Figura 6 – Interpolacao sobre a aresta de um elemento linear.

A isolinha pode interceptar um determinado elemento de diversas maneiras. A Figura (7) representa as possibilidades para um elemento quadrilátero e um elemento triangular. Os pontos escuros representam um valor acima do valor de uma isolinha genérica, assim como, os pontos claros representam valores abaixo.

Nota-se que no caso do elemento ser quadrilátero ocorre uma situação particular que pode haver duas possibilidades de preenchimento. Esse problema pode ser resolvido conhecendo-se o valor num ponto interno ao elemento. Este ponto pode ser o centro do elemento que é determinado pelo emprego das funções de forma do elemento.



(a) - Elemento quadrilátero



(b) - Elemento triangular

Figura 7 – Possibilidades de uma isolinha interceptar um elemento (adaptado de Angel, 2000 e Earnshaw et al, 1997)

Um exemplo ilustrativo para o preenchimento de um elemento triangular linear pode ser dado por uma combinação de dois casos possíveis daqueles exibidos na Fig. (7b). Deve-se observar inicialmente que ocorre uma diferença de faixas de valores para uma certa grandeza (deslocamento, deformação ou tensão) entre os nós do elemento representado na Fig. (8), assim existem uma ou mais isolinhas que interceptam esse elemento. Os pontos de intersecção das isolinhas com as arestas do elemento são obtidos através das relações encontradas na Fig. (6), pois a interpolação na aresta do elemento é linear. A Isolinha 1 é fronteira entre as faixas amarela e laranja (Tab. (1)). Nesse caso, os valores nos nós k e i estão acima do valor da isolinha 1 e o valor no nó j está abaixo. Esse caso de preenchimento é representado pelo sexto caso da Fig. (7b). A Isolinha 2 é fronteira entre as faixas vermelha e laranja. Para esse caso, os valores nos nós k e j estão abaixo do valor da isolinha 2, enquanto o valor no nó i está acima. Esse caso de preenchimento é representado pelo sétimo caso da Fig. (7b). Como composição desses dois casos obtém-se a configuração apresentada na Fig. (8). É importante ressaltar que a idéia para o preenchimento de um elemento quadrilátero é análoga a de um elemento triangular.

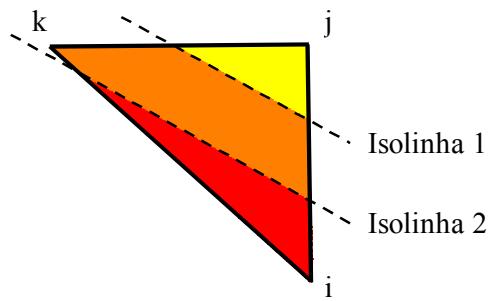
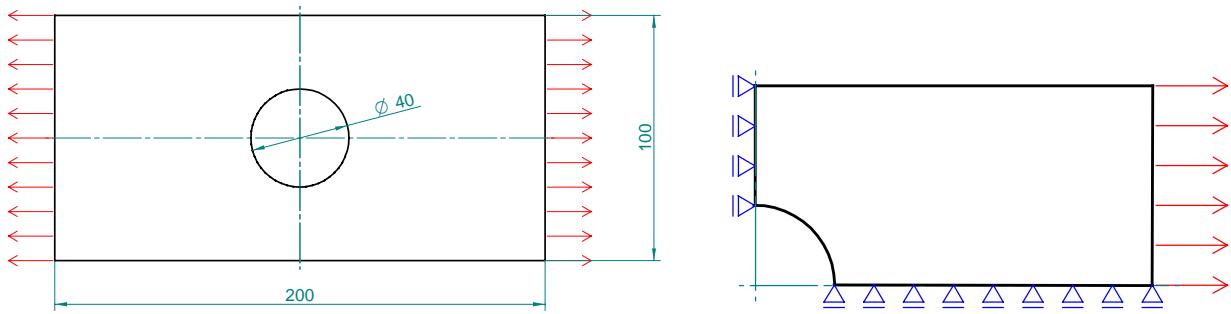


Figura 8 – Exemplo de configuração de um elemento triangular linear

### 3. ESTUDO DE CASO

Com o objetivo de validar a interface desenvolvida, estabeleceu-se como estudo de caso o problema de uma chapa com um orifício central sob a ação de um carregamento externo distribuído uniformemente nas extremidades (50 MPa), conforme exibido na Fig. (9a). Para a análise utilizou-se da simetria do problema, Fig. (9b). Os parâmetros de material (isotrópico) adotados foram: Módulo de elasticidade  $E = 2 \times 10^{11} \text{ N/m}^2$ , e coeficiente de Poisson  $\nu = 0,27$ .



(a) – Geometria da chapa com orifício central sob a ação de carregamento distribuído (Unidade: mm)

– Simetria utilizada na análise do problema

Figura 9 – Estudo de caso, chapa com orifício central

### 4. RESULTADOS

Nas figuras a seguir são mostrados os resultados comparativos entre a interface desenvolvida e os resultados gráficos obtidos através do programa ANSYS®. Nesse estudo, os dados do ANSYS® foram utilizados, com o propósito de validar a interface desenvolvida. A Fig. (10) é mostra a malha de elementos que constituem a chapa em sua configuração inicial.

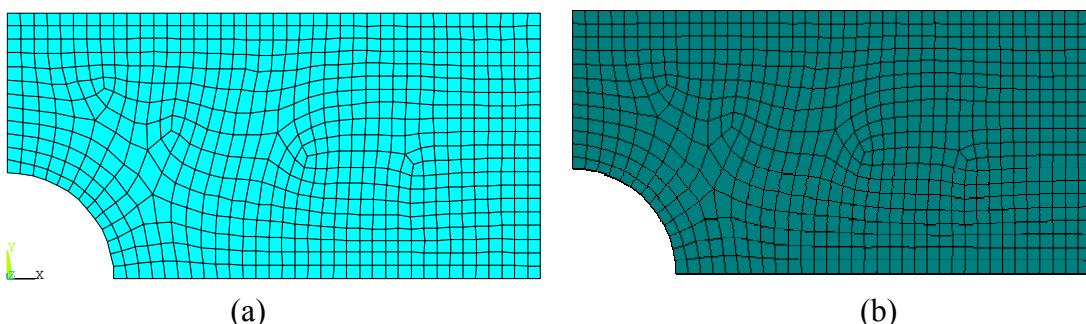


Figura 10 – Malha de elementos do estudo de caso proposto: (a) ANSYS® (b) Interface gráfica

Na Fig. (11) é mostrado o gradiente de deslocamento na direção x. A Fig. (12) traz a solução elementar para a tensão na direção x.

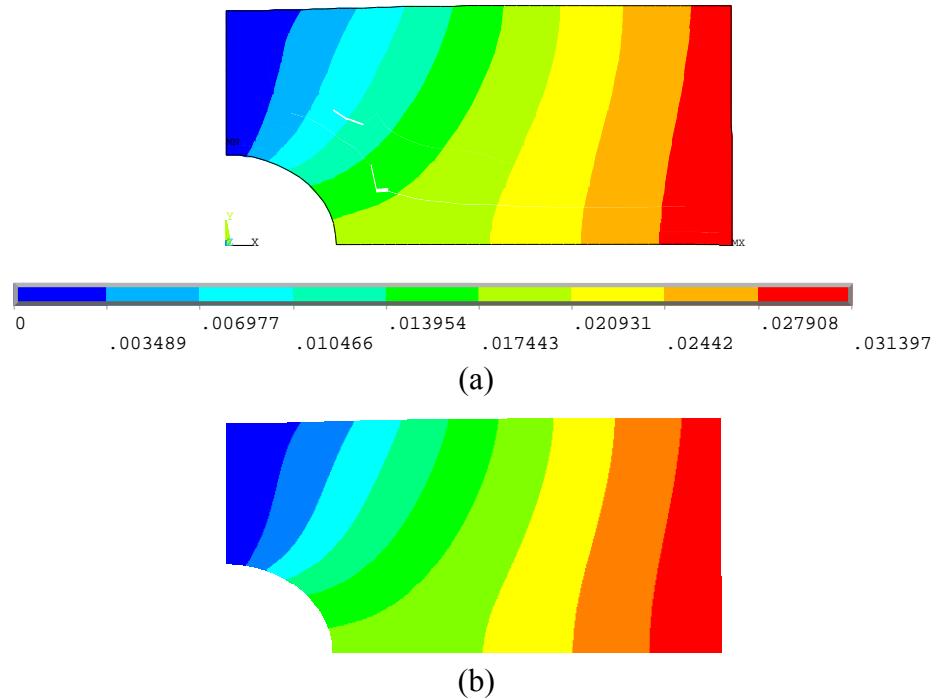


Figura 11 – Gradiente de deslocamento na direção x: (a) ANSYS<sup>®</sup> (b) Interface gráfica

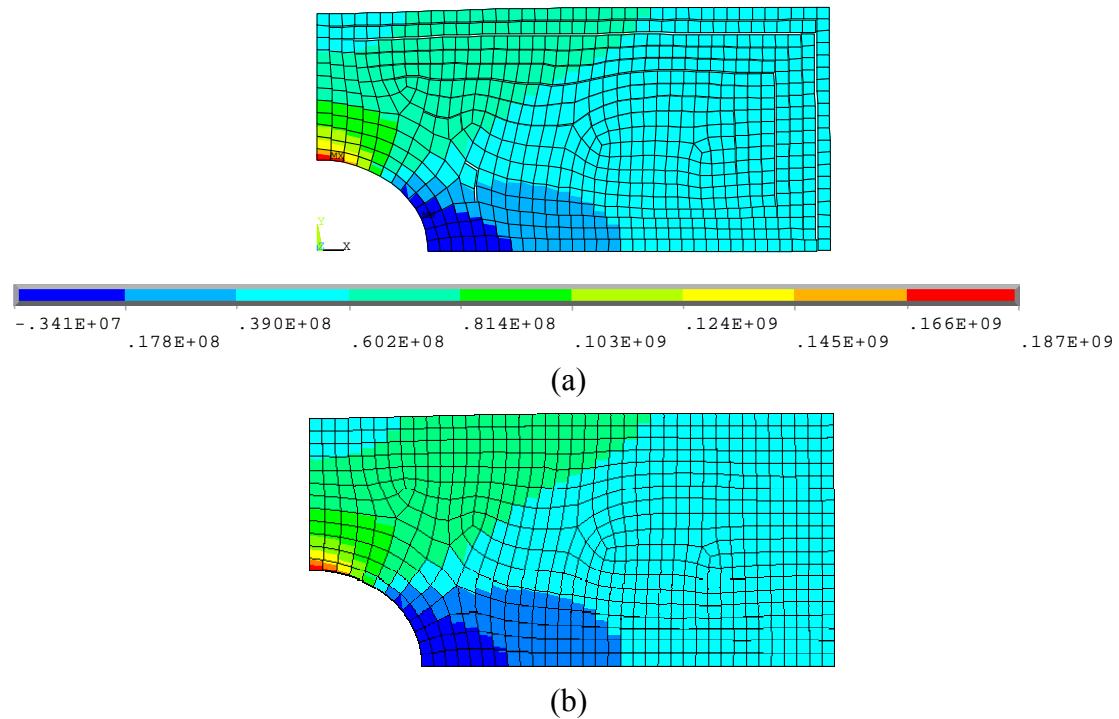


Figura 12 – Gradiente de tensão na direção x (solução elementar): (a) ANSYS<sup>®</sup> (b) Interface gráfica

Além desses resultados, a interface permite a visualização dos gradientes de deslocamento, tensão e deformação para as direções x e y. Estas opções estão dispostas para elementos quadriláteros e triangulares. A interface também exibe os resultados tanto para a solução nodal quanto para a solução elementar.

## **5. CONCLUSÕES**

Primeiramente, tem-se a importância que uma interface gráfica representa para programas de engenharia, como aqueles baseados no MEF que propiciam um número consideravelmente grande de dados. O usuário deste tipo de programa procura resultados que forneçam uma visão global do comportamento que a estrutura assume. Porém, a interface não deve somente oferecer informações de um modo geral, pois, muitas vezes, o usuário deseja saber o comportamento de uma região específica da estrutura.

Em relação aos resultados da interface para o estudo de caso proposto, verifica-se a coerência dos resultados obtidos quando comparados com os resultados gráficos do programa comercial ANSYS®. Isso demonstra que a metodologia utilizada é adequada para ser aplicada no pré e pós-processamento de problemas baseados no MEF.

Por fim, estando a interface validada, esta pode ser aplicada a programas de elementos finitos desenvolvidos pelo próprio usuário, permitindo assim o desenvolvimento de soluções próprias, e proporcionando assim uma certa independência tecnológica. Atualmente programas comerciais importados exigem o pagamento de uma licença de uso muitas vezes com duração de apenas um ano. Somado a esta questão econômica, tem-se que muitas vezes, esses programas comerciais não atendem aos requisitos para a análise, pois os mesmos não contemplam formulações desejadas e também não permitem a implementação de novas formulações.

## **6. AGRADECIMENTOS**

Os autores agradecem ao Programa Institucional de Bolsas de Iniciação Científica – PIBIC do CNPq pela concessão da bolsa e ao Laboratório de Mecânica Aplicada e Computacional (LAMAC) da EESC/USP pela concessão do programa ANSYS®.

## **7. REFERÊNCIAS**

Anacleto R. C., 1991, “Aumento da produtividade dos sistemas CAD através de parametrizados”, Dissertação (Mestrado), São Carlos, Escola de Engenharia de São Carlos, Universidade de São Paulo.

Ferreira, A.B., 1991, “CAD/CAM: Conceitos e aplicações em projetos mecânicos e critérios para seleção e utilização em engenharia”, Dissertação (Mestrado), São Paulo, Escola Politécnica. Universidade de São Paulo.

Carvalho, J., 1989, “Projeto e implementação de pré/pós-processador gráfico para o método dos elementos finitos”, Dissertação (Mestrado), São Carlos, Escola de Engenharia de São Carlos, Universidade de São Paulo.

Neider, J., 1993, “OpenGL programming guide: the official guide to learn OpenGL”, Addison-Wesley.

Gallagher, R. S., 1994, “Computer Visualization: graphics techniques for scientific and engineering analysis”, CRC Press

Hinton E., Owen, D. R. J., 1979, “An Introduction to Finite Element Computations”, Swansea, Pineridge Press Limited.

Angel, E., 2000, “Interactive computer graphics: a top-down approach with OpenGL”, Addison-Wesley.

Earshaw, R. et al., 1997, “Visualization & modeling”, Academic Press.

## **8. DIREITOS AUTORAIS**

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.

# GRAPHICAL INTERFACE FOR A FINITE ELEMENT CODE

**Ricardo Afonso Angélico<sup>1</sup>, Rodrigo Bresciani Canto<sup>2</sup>, Jonas de Carvalho<sup>3</sup>.**

Universidade de São Paulo - Escola de Engenharia de São Carlos - Depto de Engenharia Mecânica.

Av. Trabalhador Sãocarlense, 400 - Cx. P. 359 - CEP 13566-590 - São Carlos - SP, Brasil -

<sup>(1)</sup>ricardoangelico@hotmail.com, <sup>(2)</sup>canto@sc.usp.br, <sup>(3)</sup>prjonas@sc.usp.br.

**Volnei Tita**

Universidade de São Paulo - Escola de Engenharia de São Carlos - Depto de Engenharia de Estruturas. Av. Trabalhador Sãocarlense, 400 - Cx. P. 359 - CEP 13566-590 - São Carlos - SP, Brasil - voltita@sc.usp.br.

**Neilor Cesar dos Santos**

Centro Federal de Educação Tecnológica da Paraíba – CEFETPB - Depto de Mecânica. Av. Primeiro de Maio, 720 - CEP: 58015-430 - João Pessoa - PB, Brasil - neilor@cefetpb.edu.br

**Abstract.** This work presents the development of a graphic pre/post-processor for finite element analysis packages. The main objective is the development and implementation of the graphic interface for the visualization of geometric model to be solved (pre-processing stage) as well as for the results (post-processing stage). In this first implementation the interface was developed for plane elements (quadrilateral and triangular). The results obtained with the use of the graphic interface are validate in a case study developed in the commercial package ANSYS®. One additional advantage of having this interface is to allow engineers and researches to develop their own finite element routines and solvers, which enables them to implement analysis features and formulations not present in the commercial packages.

**Keywords:** Graphic interface, Finite Element, Pre/post-processors