

ARANHA: GERADOR DE MALHAS 2D  
PARA ELEMENTOS FINITOS TRIANGULARES DE 3 E 6 NÓS

Eduardo A. Fancello  
Programa de Engenharia Mecânica.  
COPPE/Universidade Federal de Rio de Janeiro.  
Rio de Janeiro - Brasil.  
Antônio C. S. Guimarães.  
Raúl A. Feijóo  
Laboratório Nacional de Computação Científica  
LNCC/CNPq  
Rio de Janeiro - Brasil

RESUMO

Este trabalho apresenta a técnica utilizada no desenvolvimento de um programa destinado a gerar malhas bidimensionais de elementos triangulares de tres ou seis nós, para sua utilização em programas de análise via elementos finitos.

Mostra-se a potencialidade do método utilizado para aplicações em malhas adaptativas, são feitos comentários com respeito às facilidades fornecidas pela linguagem C na implementação deste programa e por último se apresentam alguns exemplos para avaliar a eficiência do mesmo.

ABSTRACT

A technic used in the development of an automatic mesh generation program for finite element analysis is presented in this work. This technic allows the generation of 2D grids of linear or quadratic triangular elements.

The applicability for adaptive remeshing is shown and it's also commented the facilities provided by the C language in the program implementation. Finally some examples are presented to evaluate the program efficiency.

## 1. INTRODUÇÃO

Um dos métodos de cálculo mais utilizados ultimamente em problemas de engenharia tem sido o Método de Elementos Finitos. O constante aumento na capacidade de processamento e memória disponível dos atuais computadores facilita a resolução de problemas de geometria complexa e de alto nível de discretização. Tem-se notado também a tendência de utilizar um maior número de elementos de construção simples (elementos de interpolação linear por exemplo) ao invés de uma quantidade menor de elementos complexos. Estas são as razões pelas quais a construção manual de uma malha de Elementos Finitos razoável em termos de resultados numéricos, é uma tarefa que, dependendo da complexidade do problema, torna-se praticamente impossível.

Surge assim uma linha de pesquisa sobre diferentes metodologias para geração automática de malhas, como também, através de pós-analisadores (analisadores de erro), algoritmos de re-geração de malhas (malhas adaptativas).

Atualmente existe uma extensa bibliografia ao respeito, onde podem identificar-se duas grandes linhas de ação:

a) Determinação de malhas em regiões de geometria simples e posterior passagem ao domínio de interesse via transformação de coordenadas, mapeamentos conformes, isoparamétricos, transfinitos, etc.

b) Métodos que procedem discretizando diretamente sobre a região de interesse: triangulações automáticas, estruturas de árvore e modificações, etc.

Trabalhos como os de BAKER T.J. [1], THACKER W.C. [2] e HO-LE [3] constituem uma revisão e classificação das diferentes técnicas desenvolvidas nesta área, encontrando-se neles uma grande quantidade de referências bibliográficas sobre o tema.

Com base nestas idéias e levando em conta a necessidade comum a todo grupo de pesquisa de contar com software próprio que facilite o desenvolvimento nas diferentes áreas em que este trabalha, foi desenvolvido o programa GAMAT2 [3], e a continuação o programa ARANHA aqui apresentado.

## 2. CRITÉRIO DE GERAÇÃO

Dentre os diferentes métodos possíveis, aqueles que demonstraram características vantajosas são as técnicas de triangularização. São também chamadas técnicas não estruturadas devido à falta de direção preferencial na orientação dos elementos dentro da malha. Caracterizam-se por:

a) Grande capacidade de geração em geometrias complicadas, simple ou multiplamente conexas, com total independência do fato de serem ou não convexas.

b) O tempo de processamento depende quase exclusivamente do número de elementos a ser gerados e não da geometria do

domínio.

Como se apontou anteriormente, o primeiro passo neste sentido foi o programa GAMAT2, que foi desenvolvido tomando como base o algoritmo proposto por LO [4].

A técnica aqui utilizada foi proposta por PERAIRE e MORGAN [5, 6]. As linhas gerais deste método são apresentadas a seguir, juntamente com sugestões e modificações propostas.

### 2.1. Definição do Corpo

Determina-se primeiramente o domínio de geração através da definição de sua geometria.

Considera-se o corpo composto de uma ou mais regiões limitadas por contornos (Figura 1). Distintos tipos de contorno podem ser previstos (reto, circular, elíptico, B-spline, etc.), dependendo do grau de exatidão que se pretende alcançar na hora de modelar o problema. Um contorno reto, por exemplo, pode estar definido pelas coordenadas do nó inicial e final deste; o circular, por estas coordenadas mais a coordenada do centro do círculo, etc.

A região é definida mediante um número finito de contornos, e o corpo pela união destas regiões. Desta forma, e como notamos acima, se obtém uma definição exclusivamente "geométrica" do domínio, sem vinculação com a discretização sobre este em elementos finitos.

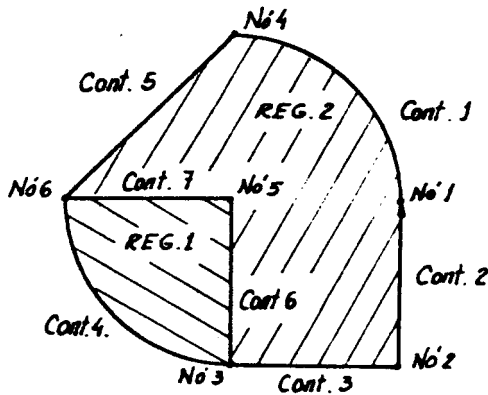


Figura 1

### 2.2. Malha de Parâmetros.

O segundo dado a ser fornecido, é a informação relativa a "forma" da malha a obter. O critério de geração aqui apresentado, baseia-se na capacidade de definição das

características que deverá ter a malha em cada ponto desta. Isto se torna possível através da definição de uma rede simples de elementos triangulares (background grid) ocupando toda a região do corpo. Nesta rede, associam-se a cada nó, valores que representam os parâmetros de interpolação de funções seccionalmente lineares, que permitem conhecer em cada ponto do domínio de definição do corpo o tamanho e uma medida da forma que deverá ter um triângulo ali situado (Figura 2).

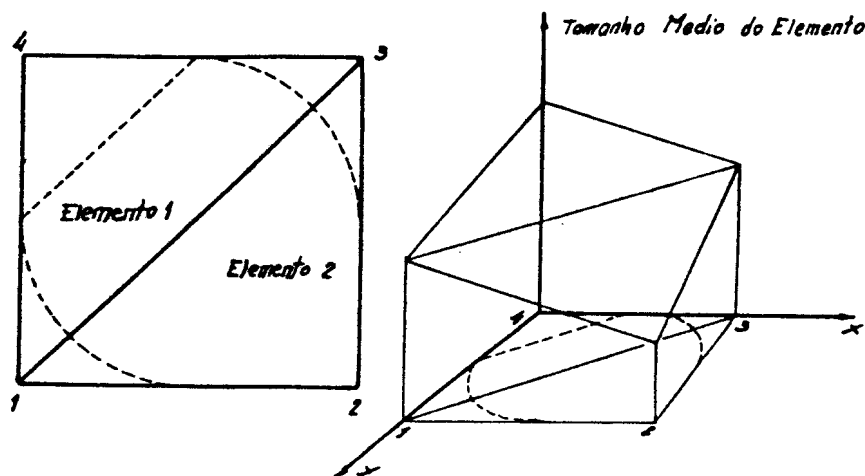


Figura 2

Os parâmetros associados a cada nó são: tamanho médio do elemento ( $\delta$ ), fator de estiramento ( $s$ ), e ângulo em que este estiramento se produz ( $\alpha$ ). Desta forma, quando são interpolados os valores ( $\delta$ ), ( $s$ ) e ( $\alpha$ ) num ponto do domínio, significa que o elemento a gerar nesse ponto deve ter dimensão média ( $s\delta$ ) na direção definida pelo ângulo ( $\alpha$ ) e dimensão média ( $\delta$ ) na direção perpendicular (Figura 3).

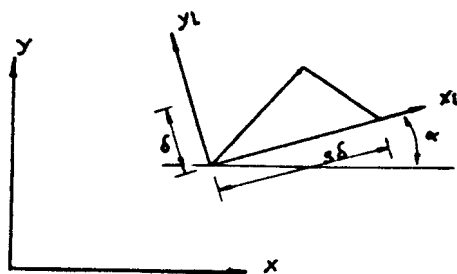


Figura 3

A partir deste pacote de informações, começa a geração propriamente dita, sendo que duas etapas podem ser facilmente distinguidas neste processo: cálculo de nós sobre o contorno e triangularização da região.

### 2.3. Cálculo dos Nós de Contorno

Note-se que os dados fornecidos pela malha de parâmetros tem uma natureza vetorial; dada uma direção existe um tamanho médio de elemento associado à mesma. Desta forma, para determinar o número de nós a inserir no contorno efetua-se uma integração de linha sobre a função  $1/\delta_c$  que determina o número de elementos por unidade de comprimento na direção tangente à curva do contorno. Assim, o número inteiro ( $N_i \approx 1$ ) que mais se aproxima a ( $\lambda_1$ ) dado por:

$$\lambda_1 = \int_0^1 \frac{1}{\delta_c(\lambda)} d\lambda \quad (1)$$

onde  $l$  é o comprimento do contorno, indica a quantidade de "lados" contidos pelo contorno. A posição dos nó  $i$ ,  $1 \leq i \leq N_i - 1$  (os nós  $i = 0$  e  $i = N_i$  correspondem ao nós inicial e final do contorno) é calculada achando o parâmetro ( $\hat{\lambda}$ ) tal que:

$$i = \frac{\lambda_1}{N_i} \int_0^{\hat{\lambda}} \frac{1}{\delta_c(\lambda)} d\lambda \quad (2)$$

Uma vez feito este cálculo, a estrutura de dados está pronta para começar a segunda etapa do problema:

### 2.4. Triangularização do Domínio.

Isto é feito a nível de cada região, tomando esta como um sub-domínio limitado por seus respectivos contornos.

Considere-se o contorno ou *front* da região, agora transformado num conjunto de segmentos onde os extremos de cada um destes são os nós calculados anteriormente. Estes segmentos são denominados *lados do front*, e em geral estão orientados de forma tal que a região a triangularizar esteja sempre a esquerda (ou a direita) do vetor [nó inicial-nó final] (Figura 4).

Assim, este conjunto de lados deve ser passível de atualização no momento que um triângulo seja gerado, modificando a fronteira ativa da região. Por exemplo na Figura 4, quando mediante alguma técnica é gerado o elemento 1, o lado [15→23] deve ser excluído do front e incluídos os lados [15→24] e [24→23]. A triangularização terá acabado quando o front se transformar num conjunto vazio.

Finalmente apresentaremos a metodologia utilizada neste trabalho para produzir a geração de um triângulo.

Baseado em algum critério (aquele adotado será mostrado posteriormente), se escolhe um lado de front e no ponto médio de este são interpolados os valores  $\delta_n$ ,  $s_n$  e  $\alpha_n$ . Com estes parâmetros se estabelece um sistema local de coordenadas através da transformação  $x_l = F x$  :

$$\begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix} = \begin{bmatrix} \cos(\alpha_n) / s_n & \sin(\alpha_n) / s_n \\ -\sin(\alpha_n) & \cos(\alpha_n) \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} \quad (3)$$

Neste novo sistema (estirado na direção  $\alpha_n$ ) se calcula a coordenada de um ponto C a distancia  $\delta_n$  dos nós inicial (A) e final (B) do lado adotado, sendo que  $\delta_n$  é um valor que depende do tamanho médio do elemento  $s_n$  e do comprimento do lado  $\overline{AB}$ :

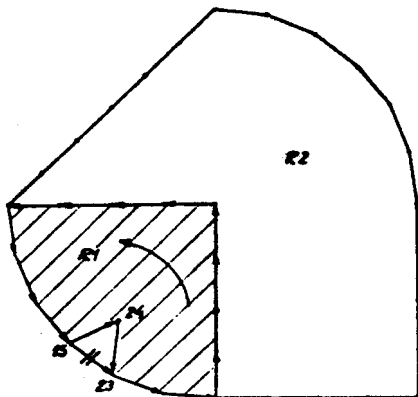


Figura 4

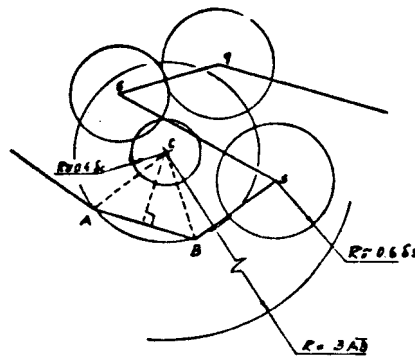


Figura 5

$$\delta_n = \begin{cases} 0.55 \overline{AB} \\ \delta_n \\ 2 \overline{AB} \end{cases} \quad \text{se} \quad \begin{cases} \delta_n < 0.55 \overline{AB} \\ 0.55 \overline{AB} < \delta_n < 2 \overline{AB} \\ 2 \overline{AB} < \delta_n \end{cases} \quad (1)$$

onde  $\overline{AB}$  é a distancia entre os pontos A e B.

Esta limitação é imposta com o objetivo de evitar distorsões excessivas na geração.

O próximo passo é gerar uma lista de nós pertencentes a front que estejam dentro de um círculo de radio R com centro em C, ordenados segundo a distância ao ponto C (o valor recomendado por [5] é  $R = 5 \overline{AB}$ ; os resultados aqui apresentados utilizaram  $R = 3 \overline{AB}$ ). Serão excluídos aqueles pontos situados a direita do vetor  $[A \rightarrow B]$  (ou a esquerda conforme tenha sido o critério adotado para a ordenação dos lados do front). Todos estes nós são candidatos a ser eleitos para gerar o novo triângulo, sendo que o nó C é incluído no início da lista sempre que:

- a) Nenhum lado de front intercepta o círculo de raio

$$r = 0.4 \delta_c \quad (5)$$

e centro em C onde  $\delta_c$  é o valor de tamanho médio de elemento interpolado no ponto C.

b) Nenhum nó  $N_i$  da lista de candidatos interpola um tamanho médio  $\delta_i$  tal que

$$0.6 \delta_i > \overline{N_i C} \quad \text{e} \quad \delta_i > \overline{N_i C} \quad (6)$$

onde  $\overline{N_i C}$  é a distancia entre  $N_i$  e C (Figura 5). Caso algum nó cumpra esta condição, a lista de candidatos começará a partir dele.

Esta série de condições têm como objetivo determinar se a inclusão do nó C como primeiro candidato, produziria incompatibilidades ou distorsões dentro da malha.

Várias diferenças podem observar-se com respeito às condições apontadas em [5]. As presentes foram adotadas por produzir melhores resultados.

A condição a) procura evitar a criação de um novo nó excessivamente próximo a um elemento de front, mesmo que os nós deste estejam longe do novo nó.

A condição b) pretende levar em conta a direção em que se está gerando, sendo por isso que se considera no teste o tamanho médio do elemento dos nós candidatos. Também evita considerar aqueles nós fora do círculo de raio  $\delta_i$  e centro em C, mediante a segunda parte de (6).

Com a lista de candidatos atualizada, se tomará o primeiro nó  $N_i$  que cumpra os seguintes testes:

- 1) A área do triângulo  $ABN_i$  deve ser  $> 0$ .<sup>1</sup>
- 2) O triângulo  $ABN_i$  não contém nenhum outro nó da lista de candidatos.
- 3) Os lados  $AN_i$  e  $BN_i$  não interceptam nenhum lado de front (a procura de possíveis intersecções é feita somente com aqueles elementos de front que possuem ao menos um nó incluído na lista de nós candidatos).

Uma vez eleito o novo nó, as coordenadas são transformadas no sistema original e após gerado o novo elemento, o front é atualizado.

Anteriormente foi apontada a existência de um critério para escolher o lado de front que servirá de base para gerar um elemento. Alguns testes foram feitos e passaremos a comentá-los.

O primeiro critério a ser utilizado foi escolher dentre os elementos do front aquele que tivesse o comprimento menor. Isto produziu uma geração excessivamente aleatória e desordenada, principalmente naquelas malhas com pouca ou nenhuma diferença de tamanho entre seus elementos.

O segundo critério consistiu em tomar o elemento do front ativo mais próximo do ponto do domínio de menor tamanho médio

---

<sup>1</sup> Isto está automaticamente satisfeito pela forma em que foram escolhidos os nós candidatos.

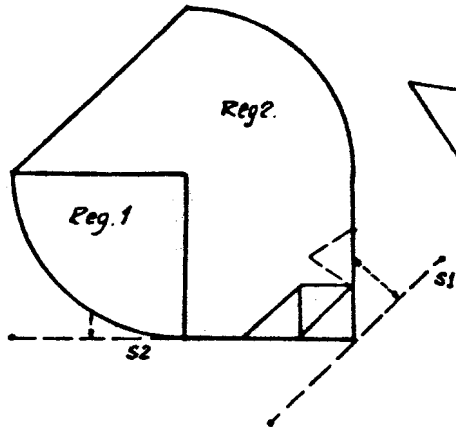


Figura 6

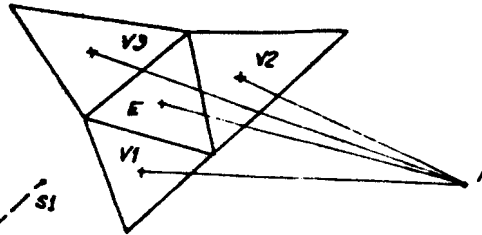


Figura 7

associado. Melhores resultados foram obtidos mas, devido à grande sensibilidade observada no que respeita ao aspecto final da malha, se optou por determinar via entrada de dados um segmento de origem e ordenar e atualizar o front segundo a menor distância a este segmento (Figura 6).

O método adotado em [5] foi escolher o último elemento do front ativo quando a diferença de tamanhos entre os elementos é pobre e o critério de menor lado de front caso contrário.

### 2.5. Algoritmo de procura na Malha de Parâmetros.

A pesquisa sobre a malha de parâmetros é um cálculo relativamente custoso e intensivamente utilizado durante todo o processo. Em [5] se indica um algoritmo que visa reduzir o tempo empregado na procura do triângulo que contém o ponto que se pretende interpolar. Este se baseia na determinação dos elementos vizinhos (Figura 7) do último elemento pesquisado, e escolher o elemento que encontra-se a menor "distância" do ponto de interesse P (isto é, aquele vizinho que produz o menor valor de coordenada de área em relação ao ponto P ou aquele cujo centro de gravidade esteja mais próximo de P). Este algoritmo dá excelentes resultados sempre que utilizado com malhas de parâmetros de geometria convexa. Quando não se dá este caso, o procedimento deve contemplar a possibilidade de bloqueio na procura, tomando atitudes alternativas.

### 3. EXTENSÃO A MALHAS ADAPTATIVAS

Como é possível observar em [5, 6], e através de uma simples análise da metodologia proposta, a aplicação destas idéias à área de malhas adaptativas é extremamente simples e inclusive independente do tipo de problema que se pretende



solucionar. Basta a criação de uma interfase (pós-processador) que tome os dados entregues pelo analisador e mediante um critério de erro, gere novos parâmetros de controle (tamanho médio do elemento, direção e estiramento) sobre a malha utilizada na última análise. Desta forma, a última malha gerada converte-se em malha de parâmetros para a geração subsequente.

Os diferentes critérios adotam a forma de módulos facilmente intercambiáveis, permitindo então concentrar esforços na elaboração destes.

#### 4. IMPLEMENTAÇÃO

Uma linguagem que está abrindo um espaço considerável nas áreas que utilizam a programação como ferramenta de cálculo, é a linguagem C. Não se pretende aqui fazer uma análise das qualidades desta linguagem, mas sim ressaltar algumas das vantagens que oferece na hora de contruir um programa:

- a) Possibilidade de criação de novos tipos de dados (assim como o tipo ponto flutuante, inteiro, booleano, etc.) através da montagem de estruturas.
- b) Capacidade para alocar memória de forma dinâmica.
- c) Possui a nível de compilação um controle estrito das variáveis, bloqueando a maioria dos erros derivados de desatenções na programação e reduzindo, por tanto, os tempos empregados na produção de software.

Explorando todas estas facilidades e dentro do sistema ACDP - Ambiente Computacional para Desenvolvimento de Programas [7], foi desenvolvido o programa ARANHA. Se mostrará o tratamento dado por este último a sua estrutura de dados.

##### 4.1. Estrutura de Dados

Para ordenar a informação de trabalho, foram criadas uma série de estruturas ou tipos contendo a informação relativa a um elemento. Por exemplo:

Informação respectivo à malha de parâmetros.

Como já se indicou, a cada nó está associado um tamanho médio de elemento, um estiramento, e uma direção na qual o estiramento se produz. A estrutura criada para os nós é:

---

<sup>2</sup> Isto pode variar com a versão do compilador.

```
struct STNOMAPAR
{
    float x, y;           /* coordenadas */
    float tammedel;      /* tamanho medio dos elementos */
    float alfa;          /* direção */
    float stretch;      /* estiramento */
};
dypedef TNOMAPAR struct STNOMAPAR;
```

Os elementos da malha de parâmetros contém além da incidência nodal, informação dos elementos que o rodeiam e coordenadas do centro de gravidade:

```
struct STELMAPAR
{
    int inci[3];         /* incidencia nodal ou elem. vizinhos */
    int viz[3];          /* incidencia elemental */
    float cg[2];        /* coordenadas do centro de gravidade */
};
```

Dependendo do número de nós e de elementos da malha de parâmetros são alocados dinamicamente vetores de elementos STNOMAPAR e STELMAPAR onde a informação é colocada.

O tratamento para elementos que contém informação relativa a nós do corpo, incidência, contornos, regiões, etc. é similar.

Uma estrutura interessante de ressaltar, é a estrutura de LISTA. Se utiliza para criar "pilhas" de informação onde cada elemento possui os dados da lista (ou o endereço destes dados) e o endereço de memória do próximo elemento (Figura 8). Esta ideia é muito útil para trabalhar com o conceito de front (2.4), dado que é necessário modificar constantemente o tamanho do conjunto de informação: inserir em forma ordenada (segundo algum critério), deletar elementos, pesquisar, etc. O sistema ACDP [7] fornece ao programador uma série de procedimentos para a manipulação (isto é, inserção, eliminação, procura, etc.) em listas genéricas, bastando para isto fornecer a função de comparação.

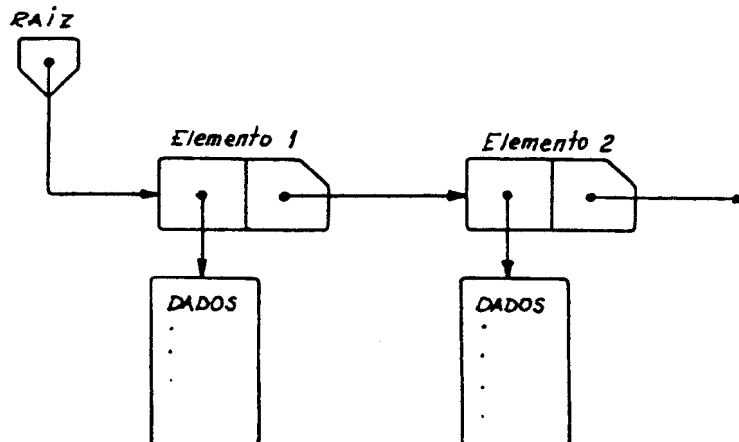


Figura 8

#### 4.2. Tratamento da malha após geração

Tres tipos de tratamentos pos-geração foram implementados: regularização da malha (smooth), geração de nós intermediários lados dos triângulos (triângulos quadráticos) e renumeração dos nós.

Não obstante todos os cuidados tomados na geração para evitar a formação de triângulos excessivamente distorcidos, podem aparecer eventualmente alguns problemas isolados. Nestes casos, a utilização de smooth, produz grandes melhorias, dando inclusive um aspecto mais agradável à malha gerada. O tipo de regularização aqui implementado é a chamada regularização Laplaciana que reposiciona o nó no centroide do polígono formado pelos elementos finitos a ele associados. Outras técnicas podem encontrar-se em [8].

A colocação de novos nós para possibilitar o trabalho com elementos de interpolação quadrática foi feito tomando cuidado de acompanhar a geometria dos contornos (Figura 9).

Finalmente, no que diz respeito à renumeração dos nós, se implementou uma técnica que procura otimizar a semi-largura de banda da matriz de rigidez montada pelo MEF, considerando resolvidores de tipo Gauss ou Choleski [9].

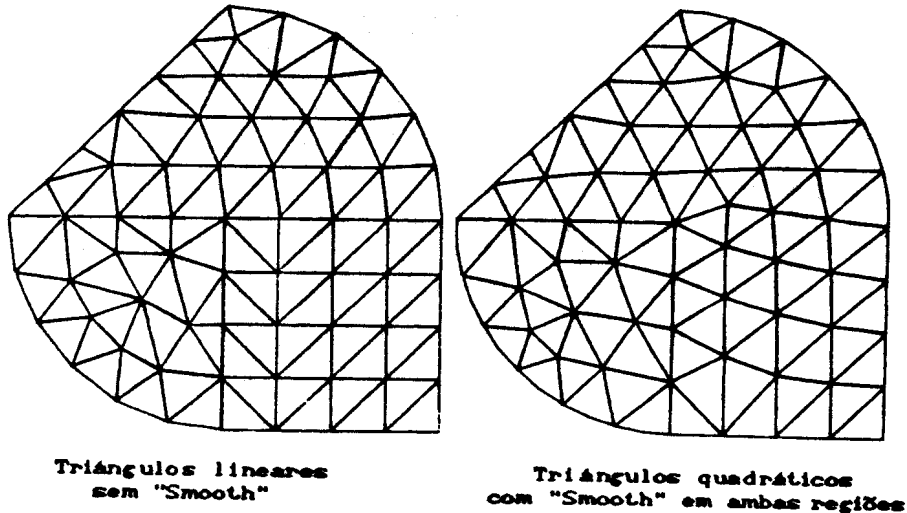
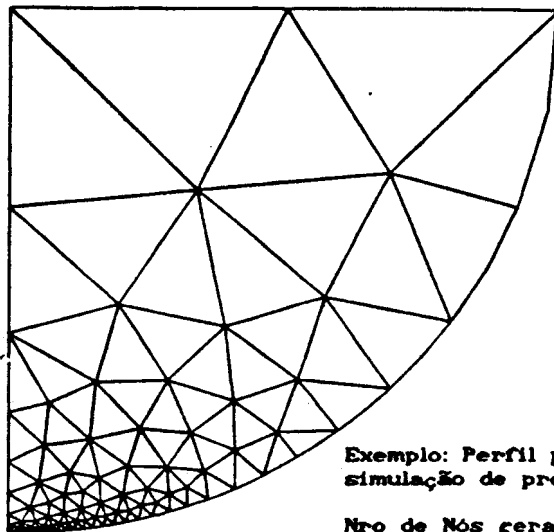


Figura 9

#### 5. EXEMPLOS DE APLICAÇÃO.

Mostraremos alguns gráficos obtidos a fim de observar a potencialidad del presente programa. Se indica numero de elementos gerados e tempo de geração em duas máquinas: microcomputador tipo IBM AT com um clock de 12 Mhz, e uma estação de trabalho SUN SPARCstation 01.



Exemplo: Perfil para  
simulação de problema de contato.

Nro de Nós gerados = 316  
Nro de Elem. gerados = 139

Tempo Tot. de CPU Micro (seg) = 31.05  
Tempo Tot. de CPU Sun (seg) = 2.70



Exemplo: Perfil de pistão.

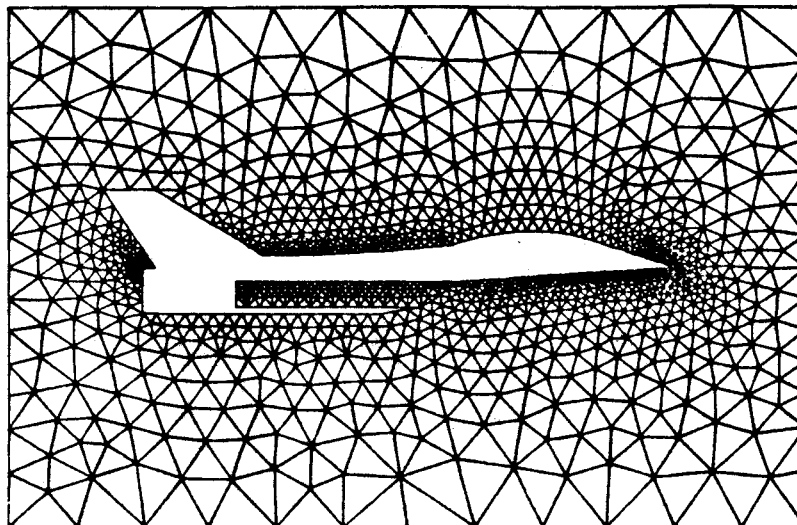
Nro de Nós gerados = 366  
Nro de Elem. gerados = 583

Tempo Tot. de CPU Micro (seg) = 216.77  
Tempo Tot. de CPU Sun (seg) = 19.24

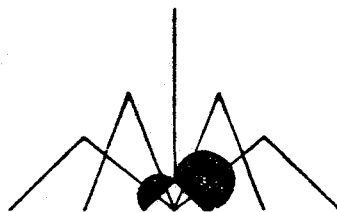
ARANHA

Malha para perfil de pistao.

24/05/1990



Exemplo: Perfil de avião para análise de fluidos  
Nro de Nós = 1156      Nro de Elem. = 2068  
Tempo Total de CPU Micro (em seg) = 855.83  
Tempo Total de CPU Sun (em seg) = 45.67



Exemplo: Aranha  
Nro de Nós = 949      Nro de Elem. = 1558  
Tempo Total de CPU Micro (em seg) = 518.85  
Tempo Total de CPU Sun (em seg) = 40.32

**BIBLIOGRAFIA**

- [1] BAKER, T.J.; "Developments and trends in three-dimensional mesh generation", Applied Numerical Mathematics, Nro. 5, 1989, pp 275-304.
- [2] THACKER, W.C.; "A brief review of techniques for generating irregular computational grids"; Int. J. Numerical Methods Eng., Vol 15, 1980, pp 1335-1341.
- [3] HO-LE, K.; "Finite element mesh generation methods: a review and classification"; Computer Aided Design, Vol 20, Nro 1, 1988, pp 27-38.
- [4] LO, S.H.; "A new mesh generation scheme for arbitrary planar domains"; Int. J. Numerical Methods Eng., Vol 21, 1985, pp 1403-1426.
- [5] PERAIRE, J.; PEIRO, J.; MORGAN, K.; ZIENKIEWICZ, O.C.; "Adaptive remeshing for compressible flow computations", J Comp. Phys., Nro 72, 1987, pp 449-466.
- [6] PERAIRE, J.; PEIRO, J.; FORMAGGIA, L.; MORGAN, K.; ZIENKIEWICZ, O.C.; "Finite element euler computations in three dimensions", Int. J. Numerical Methods Eng., Vol 26, 1988, pp 2135-2159.
- [7] GUIMARÃES, A.C.S.; FEIJÓO, R.A.; "O sistema ACDP", Anais do X Congresso Ibero-Latino-Americano sobre Métodos Computacionais para Engenharia, Set. 89, Porto, Portugal.
- [8] HERMANN, L.R.; "Laplacian isoparametric grid generation scheme", J. of the Mechanics Division of the American Society of Civil Engineers, Vol 120, Nro EM5, 1976, pp 49-756.
- [9] VENERE, M.J.; "Generador de redes ENREDO", Publicação interna Centro Atómico Bariloche, Argentina, 1987.
- [10] GUIMARÃES, A.C.S.; FEIJÓO, R.A.; "GAMAT2 - Gerador automático de malhas triangulares de elementos lineares e quadráticos", Publicação Interna LNCC/CNPq