



GERAÇÃO AUTOMÁTICA DE MALHAS TRIANGULARES POR ANALOGIA MECÂNICA

REIS, ELIEL TAVARES DOS

RIBEIRO, PAULO MARCELO VIEIRA

eliel.reis1@gmail.com

[paulo.vribeiro@ufpe.br](mailto: paulo.vribeiro@ufpe.br)

Universidade Federal de Pernambuco – UFPE

Centro de Tecnologia e Geociências

Departamento de Engenharia Civil

Rua Acadêmico Hélio Ramos, s/n, Cidade Universitária

50740-530

Recife-PE

Abstract. O desenvolvimento de algoritmos de geração automática de malha não estruturada tem se intensificado nas últimas décadas, com aplicação a geometrias de superfície irregular e complexa. Adotar métodos de suavização durante o processo de geração se tornou uma necessidade para produção de malhas de alta qualidade. Das técnicas de suavização, as mais eficientes são as abordagens por analogia mecânica. O objetivo deste trabalho é a elaboração de um gerador automático de malha não estruturada de elementos finitos triangulares no plano, com posterior suavização por analogia mecânica. A geração inicial da malha é estabelecida pela triangulação Delaunay a partir de uma nuvem de pontos randômicos ou de um grid estruturado. As analogias mecânicas utilizadas para geração de malha são baseadas em simples e avançadas estruturas de molas fictícias. A implementação do código gerador é elaborada na linguagem Python e possui parâmetros de entrada: i - geometria do domínio, ii - comprimento mínimo dos elementos. Os dados referentes à geometria são obtidos através de um arquivo de Desenho Assistido por Computador (CAD). Adicionalmente, foram feitos exemplos de validação que confirmam a eficiência em desempenho computacional e em qualidade dos elementos.

Keywords: Gerador de Malha, Suavização, Analogia mecânica, Método dos Elementos Finitos.

1 INTRODUÇÃO

A aplicação do Método dos Elementos Finitos (MEF) a problemas complexos de engenharia tem exigido cada vez mais automatizações nos cálculos e precisão nos resultados. Nas últimas décadas, surgiram estudos aplicados ao desenvolvimento de algoritmos de geração e suavização de malhas, principalmente em domínios irregulares, curvilíneos e com furos.

Os métodos de geração de malha não estruturada progrediram no início dos anos 80, principalmente pelo desenvolvimento de três esquemas populares de geração de malhas: i) Triangulação Delaunay, ii) Método do avanço de fronteira e iii) Decomposição recursiva quadtree e octree.

A primeira abordagem consiste em construir uma triangulação convexa a partir de uma nuvem de pontos. A base teórica da triangulação Delaunay foi estabelecida por Dirichlet (1850), demonstrando que, para um dado conjunto de nós em duas dimensões é possível dividir o plano em células convexas usando um critério de proximidade. Nos anos 30, Delaunay (1934) deduziu uma triangulação através do diagrama de Voronoi (1908) pela dualidade. Essa propriedade é representada por uma característica-chave, chamada de critério de círculo vazio. Os círculos que circunscrevem os triângulos não contêm qualquer outro nó nos seus interiores. Sibson (1978) notou a maximização dos ângulos mínimos dos elementos gerados pela triangulação Delaunay; uma vantagem importante em relação às outras técnicas de triangulação por gerar bons elementos.

O método do avanço de fronteira foi proposta por Lo. S. H. (1985), em que a triangulação é construída progressivamente do contorno para a parte interna do domínio. A cada iteração, a fronteira “ativa” é atualizada na medida em que novas camadas de triângulos irão se formando até que o domínio fique preenchido por completo.

Geradores de malhas por decomposição recursiva se baseiam em estruturas de dados quadtree e octree em domínios bidimensionais e tridimensionais, respectivamente. O algoritmo funciona em subdivisão sucessiva do domínio em células (triângulos). Inicialmente envolve o domínio em uma célula e geram outras por divisões sempre que as estiverem parcialmente dentro do domínio ou algum critério de parada ainda não alcançado.

Elementos irregulares presentes na malha podem afetar a precisão dos cálculos numéricos, apresentando inconsistência nos resultados. Babuska & Aziz (1976) concluíram que, para obter soluções próximas do real era essencial gerar elementos com ângulos máximos não próximos a 180 graus, ou seja, elementos achatados. Gerar malha com triângulos bem parecidos com equiláteros passaria a ser o desafio dos geradores.

O algoritmo produto deste trabalho foi influenciado por trabalhos publicados sobre analogia mecânica e pelo trabalho de Persson (2005), apresentando uma nova técnica de geração de malhas baseada em equilíbrio de forças em domínios de geometrias específicas com funções implícitas.

Neste contexto, o presente trabalho aborda o desenvolvimento de um algoritmo gerador de malha, enfatizando a técnica de triangulação Delaunay e um processo de suavização por deslocamento nodal análogo a comportamentos mecânicos. Também é apresentada uma evolução do método desde um simples cálculo de média aritmética, como a Suavização Laplaciana, a deslocamentos causados pela Analogia de Mola Semi-torcional.

2 SUAVIZAÇÃO DA MALHA

As técnicas de suavização de malha têm como principal objetivo melhorar as qualidades dos elementos envolvidos sem mudar sua topologia. Os algoritmos de suavização usam técnicas locais de deslocamento nodal em busca de uma melhor posição de cada nó, resultando em uma otimização dos elementos que compartilham o mesmo nó.

Abaixo são apresentados os métodos de suavização de forma evolutiva.

2.1 Suavização Laplaciana

A técnica de suavização Laplaciana consiste em melhorar a malha já gerada deslocando os nós internos i para a posição centroide do polígono formado pelos os nós adjacentes j , ou seja, o cálculo da nova posição de cada nó i é dado pela média aritmética dos nós adjacentes, minimizando a distorção dos elementos gerados. Este processo sem alteração da topologia foi abordado por Hermann (1976).

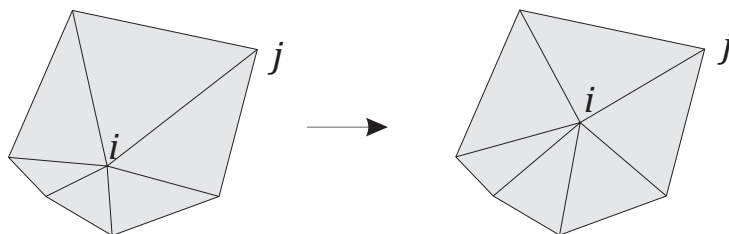


Figura 1 - Algoritmo do método de suavização Laplaciana

$$\vec{P}_{i_{new}} = \frac{1}{n_A} \sum_{j=1}^{n_A} \vec{P}_{j_{old}} \quad (1)$$

Onde n_A é a quantidade vértices adjacentes conectados ao vértice i e $\vec{P}_i$ é o vetor posição.

O quadro abaixo fornece o pseudocódigo do algoritmo utilizado para aplicação da suavização laplaciana na linguagem de programação Python:

Quadro 1 - Pseudocódigo do algoritmo de suavização laplaciana

- Dados de entrada:
 - Matriz das coordenadas dos pontos nodais: $\mathbf{P}$
 - Matriz das conectividades dos elementos triangulares: $\mathbf{CONEC}$
 - Matriz de identificação dos índices dos pontos nodais do contorno na matriz $\mathbf{P}$, pois precisam ser restringidos durante a suavização: $\mathbf{P}_c$
- Número de pontos nodais: n_p

- Cálculo das novas posições nodais $P_{i_{new}}$ dos pontos internos:

$i = 1, 2, \dots, n_p$

Restrição: Se $P_i \in P_c$

Identificação dos pontos adjacentes: P_j

Número de pontos adjacentes: n_{p_j}

$$P_{i_{new}} = \frac{1}{n_{p_j}} \sum_{j=1}^{n_{p_j}} P_j$$

- Fim.

Canann, Tristano & Staten (1988) mostraram que o método poderia resultar em um possível colapso ou inversão dos elementos em regiões côncavas presente no domínio em análise. Observa-se também que é fácil de implementar, entretanto nem sempre move o nó para a posição ideal.

Para superar essas desvantagens, Lo. S. H. (1992) propôs um esquema restrito de suavização Laplaciana, conhecido por suavização Laplaciana inteligente, ou restrita, em que o deslocamento nodal só aconteceria se houvesse um melhoramento na qualidade geral dos elementos conectados. Este método é eficaz para evitar elementos invertidos, mas o custo computacional aumenta bastante por estar sempre verificando as qualidades dos elementos antes e após a cada iteração.

2.2 Suavização da malha por analogia mecânica

Batina (1990) introduziu um método de suavização análogo a uma rede de tensão de molas longitudinais, isto é, cada aresta da malha é substituída por uma mola fictícia.

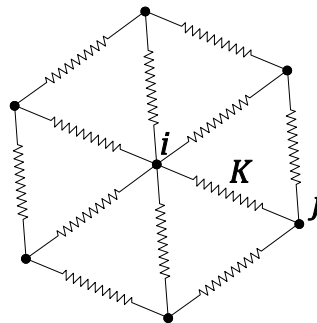


Figura 2 - Analogia de mola longitudinal

2.2.1 Analogia da mola longitudinal de segmento

Nesta seção será apresentada uma breve revisão do método. Dado dois pontos nodais, i e j , o vetor-aresta partindo de i para j é definido como:

$$\vec{V}_{ij} = \vec{P}_j - \vec{P}_i \quad (2)$$

E seu comprimento é dado pela expressão abaixo:

$$L_{ij} = \sqrt{\vec{V}_{ij} \cdot \vec{V}_{ij}} \quad (3)$$

O vetor unitário de $\vec{V}_{ij}$ pode ser escrito:

$$\vec{t}_{ij} = \frac{\vec{V}_{ij}}{L_{ij}} \quad (4)$$

A analogia consiste no equilíbrio das forças provenientes da perturbação da mola em relação ao comprimento não estendido l .

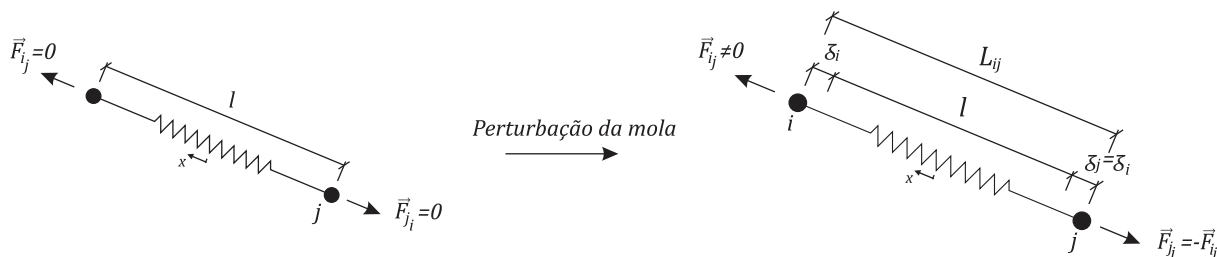


Figura 3 - Demonstração da perturbação da mola

Os deslocamentos dos pontos nodais ou vértices i e j são especificados δ_i e δ_j respectivamente:

$$\delta_i = \delta_j = \frac{l - L_{ij}}{2} \quad (5)$$

As forças em cada vértice e alinhada ao vetor $\vec{V}_{ij}$ podem ser calculadas e escritas pela seguinte expressão:

$$\vec{F}_{ij} = K_{ij}(\delta_i \cdot \vec{t}_{ij} - \delta_j \cdot \vec{t}_{ij}) = -\vec{F}_{ji} \quad (6)$$

Onde K_{ij} é a rigidez da mola análoga.

Uma particularidade na técnica da mola de segmento, no intuito de impedir a colisão dos pontos nodais conectados, Batina (1990) propôs a rigidez K_{ij} inversamente proporcional ao comprimento do segmento:

$$K_{ij} = \frac{1}{L_{ij}} \quad (7)$$

A nova posição de cada nó é obtida através da expressão de equilíbrio da Lei de Hooke para as forças produzidas por cada aresta adjacente:

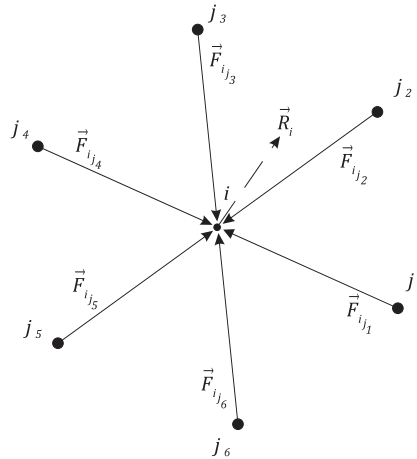


Figura 4 - Equilíbrio de um nó submetido a várias forças (dos vértices adjacentes)

$$\vec{R}_i = - \sum_{j=1}^{n_A} \vec{F}_{ij} = 0 = K_{eq} \cdot \vec{\delta}_{i_{new}} \quad (8)$$

$$\vec{P}_{i_{new}} = \vec{P}_{i_{old}} + \vec{\delta}_{i_{new}} \quad (9)$$

Onde K_{eq} é a rigidez equivalente e n_A o número de arestas conectadas. Uma das maneiras de definir a rigidez equivalente é o somatório das rigidezes das molas conectadas:

$$K_{eq} = \sum_{j=1}^{n_A} K_{ij} \quad (10)$$

Abaixo segue o pseudocódigo geral do algoritmo de analogia de mola:

Quadro 2 - Pseudocódigo do algoritmo de analogia de mola longitudinal

- Dados de entrada:
 - Matriz das coordenadas dos pontos nodais: **P**
 - Matriz das conectividades dos elementos triangulares: **CONEC**
 - Matriz de identificação dos índices dos pontos nodais do contorno na matriz **P**, pois precisam ser restringidos durante a suavização: **P_c**
 - Comprimento não estendido: **l**
- Matriz de conectividade das arestas através da matriz **CONEC**: **BARS = (i j)**
- Número de arestas: **n_B**
- Vetores-arestas: **$\vec{V}_{ij}$**
- Comprimentos das arestas: **L_{ij}**
- Vetores unitários: **$\vec{t}_{ij}$**

- Rigidez das molas análogas: K_{ij}
- Cálculo das intensidades das forças provenientes das perturbações das molas análogas:

$$F_{ij} = K_{ij} \cdot (l - L_{ij})$$

$$F_{ji} = -F_{ij}$$
- Cálculo das reações:

$$\rightarrow k = 1, 2, \dots, n_B$$

$$i, j = BARS(k)$$

$$\vec{R}_i = \vec{R}_i - F_{ij} \vec{l}_{ij}$$

$$\vec{R}_j = \vec{R}_j - F_{ji} \vec{l}_{ij}$$

$$K_{eq_i} = K_{eq_i} + K_{ij}$$

$$K_{eq_j} = K_{eq_j} + K_{ij}$$
- Cálculo dos deslocamentos nodais: $\vec{\delta} = \frac{\vec{R}}{K_{eq}}$
- Cálculo das novas posições nodais:
 - Restrição dos pontos nodais no contorno.

$$\vec{P}_{new} = \vec{P} + \vec{\delta}$$
- Fim.

2.2.2 Analogia de mola longitudinal de vértice

Na comunidade científica são discutidos dois tipos de mola longitudinal: de vértice e de segmento. O grande diferencial entre os dois está no comprimento de equilíbrio da mola.

A analogia de mola longitudinal de vértice tem como variáveis as próprias coordenadas dos nós, ou seja, o seu comprimento não estendido l é considerado zero:

$$\vec{F}_{ij} = K_{ij}(-L_{ij} \cdot \vec{l}_{ij}) = -\vec{F}_{ji} \quad (11)$$

De acordo com as Eq. (8), (9), (10) e (11), observa-se uma particularidade desse método com a Suavização Laplaciana quando a rigidez $K_{ij} = 1$:

$$\vec{\delta}_{i_{new}} = \frac{1}{n_A} \sum_{j=1}^{n_A} L_{ij} \cdot \vec{l}_{ij} \quad (12)$$

2.2.3 Analogia de mola semi-torcional

A rigidez da mola longitudinal apenas considera o efeito de alongamento, ou seja, despreza as áreas e os ângulos dos triângulos conectados, resultando em possíveis mecanismos de colapsos em grandes deformações, como a inversão dos elementos triangulares.

Blom (2000) e Zang & Ethier (2005) introduziram o método de mola semi-torcional. A rigidez da mola longitudinal passa a depender dos ângulos triangulares.

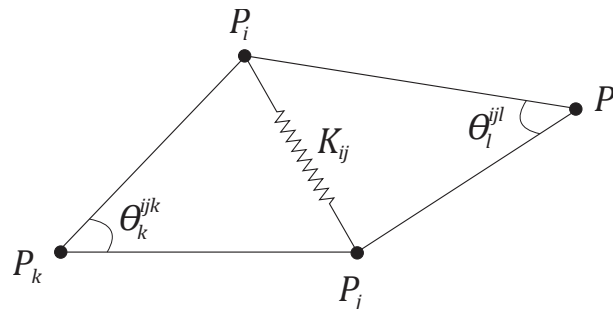


Figura 5 - Algoritmo da analogia de mola semi-torcional

O coeficiente de rigidez introduzido por Zang & Ethier (2005) é dado pelo somatório da rigidez linear com as semi-torcionais voltadas para aresta ij :

$$K_{ij}^{Total} = K_{ij}^L + K_{ij}^{S-T} \quad (13)$$

$$K_{ij}^{S-T} = \sum_{m=1}^{n_E} \frac{1}{\sin^2 \theta_m^{ij}} \quad (14)$$

Onde n_E se refere ao número de elementos que compartilham a aresta ij . Já o coeficiente de rigidez longitudinal K_{ij}^L é o mesmo definido na Eq. (7).

3 ESTRATÉGIA PARA GERAÇÃO AUTOMÁTICA DE MALHA BIDIMENSIONAL

3.1 Estratégia para definição da superfície do domínio

A definição da malha de elementos finitos é a subdivisão do domínio em elementos de comportamento conhecido, geralmente triângulos e quadriláteros, formando a malha. O grande desafio do código é gerar a malha obedecendo ao contorno do domínio, ou seja, inserir pontos nodais e gerar os elementos no interior do domínio.

Para contornar tais problemas, foi desenvolvido um algoritmo de discretização da fronteira em vários segmentos, ou um polígono com várias arestas.

Uma técnica simples de verificar se o ponto está dentro ou fora de um polígono, é testar quantas vezes um raio, partindo do ponto de análise e indo a qualquer direção fixa, cruza as bordas do polígono. Se o ponto estiver fora do polígono, o raio irá cruzar a borda um número de par de vezes, caso contrário, irá cruzar um número de ímpar de vezes.

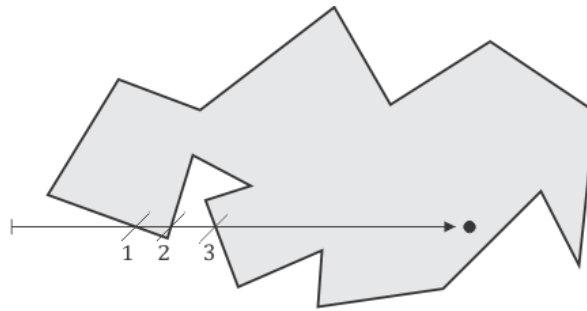


Figura 6 - Algoritmo Ray Casting

Esse algoritmo é conhecido por Ray Casting e já foi mencionado por Shimrat (1962). Entretanto possui alguns cuidados e limitações:

- i) Se caso o raio intercepta a borda do polígono exatamente no vértice, o cruzamento deve ser contabilizado para o segmento mais a direita e abaixo do vértice.
- ii) É possível que o raio coincida exatamente com a aresta do polígono, a solução é ignorar o teste nesses segmentos.
- iii) O método apresenta limitação da não verificação dos pontos inseridos exatamente no contorno.

3.2 Comprimento alvo das arestas dos elementos triangulares - h

O grau de refinamento da malha depende do comprimento alvo das arestas dos elementos triangulares - h . Quanto maior for h , menores serão os números de vértices e arestas, no caso oposto, o domínio será discretizado em maiores quantidades de vértices e elementos triangulares. Resultando em uma quantidade suficiente de pontos nodais.

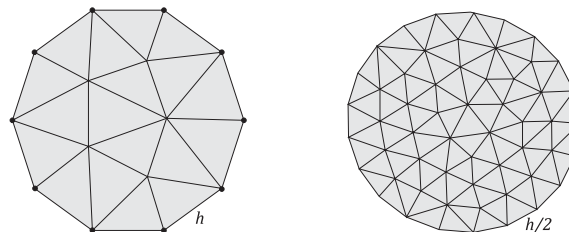


Figura 7 - Comprimento desejado da aresta final

Uma das exigências para o MEF é que as arestas das bordas coincidam com a fronteira do domínio, isso é obrigatório na análise. A estratégia adotada foi inserir os pontos nodais fixos ao longo da fronteira de forma equidistante:

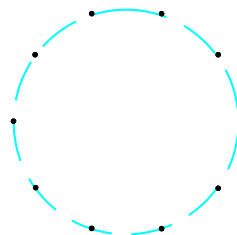


Figura 8 - Pontos nodais na fronteira do domínio

Neste trabalho serão apresentados dois algoritmos distintos na geração da malha inicial, mas ambos utilizam o algoritmo Delaunay para a triangulação. A diferença entre os algoritmos está na inserção dos pontos nodais no interior do domínio de análise, um se dá pela nuvem de pontos randômicos e o outro pelo grid estruturado de pontos.

3.3 Nuvem de pontos randômicos

De ponto de vista bidimensional, a nuvem de pontos é um conjunto de pontos aleatório no plano, geralmente são definidos por coordenadas x e y , comumente têm o objetivo de ser o conjunto de vértices para aplicação da triangulação Delaunay.

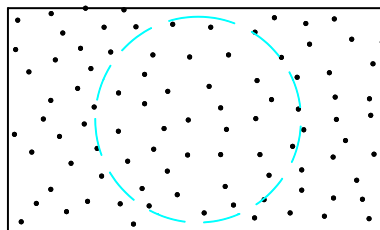


Figura 9 - Nuvem de pontos randômicos inicial

A estratégia utilizada consiste na inserção de pontos obedecendo ao comprimento mínimo desejado das arestas - h . A partir de todos os pontos aleatórios pré-definidos (inicialmente um ponto) na região onde o domínio é inscrito, ao redor desses pontos são processados uma série de pontos randômicos, candidatos, em seguida são aproveitados os pontos no intervalo entre os raios h e $2h$ de cada ponto pré-definido, desses são aproveitados os que obedecem a distância mínima h entre si e por fim são aproveitados os que obedecem a distância mínima com os pontos pré-definidos já existente.

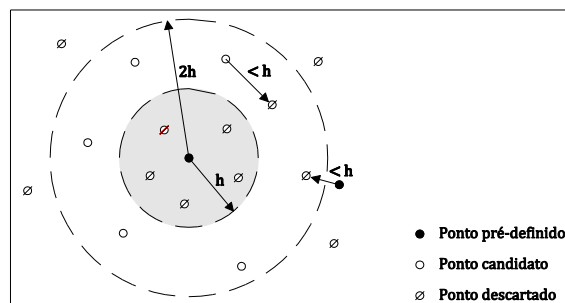


Figura 10 - Inserções de pontos nodais randômicos

O processo de inserção é repetido sistematicamente até que a região onde o domínio está inscrito fique totalmente preenchida. Em seguida aplica o algoritmo Ray Casting é aplicado e são obtidos conjuntos de pontos no interior do domínio de forma sistemática.

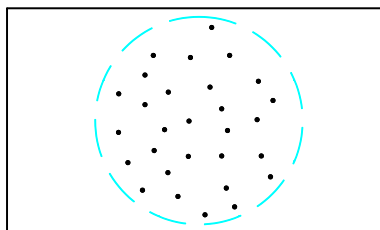


Figura 11 - Nuvem de pontos randômicos finais

3.4 Grid estruturado de pontos

Grid de pontos estruturados é um conjunto de pontos organizados no plano de modo que as ligações aos pontos adjacentes formem triângulos equiláteros. Mais uma vez o comprimento mínimo desejado h aparece como o comprimento dos lados dos triângulos. Essa ideia foi baseada no algoritmo de Persson (2005).

Novamente pelo algoritmo Ray Casting, elimina-se os pontos fora do domínio:

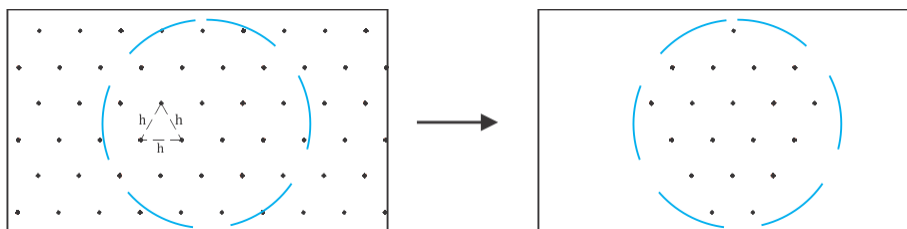


Figura 12 - Grid de pontos estruturados

3.5 Triangulação Delaunay

Delaunay (1934) mostrou que a questão chave para a maioria dos algoritmos de triangulação Delaunay estar no seguinte lema geral: Seja uma triangulação arbitrária, para cada par de elementos triangulares, a condição da esfera vazia é válida, ou seja, uma condição global para definição de uma triangulação Delaunay. Em outras palavras, se todas as arestas da triangulação T do conjunto de pontos P for localmente Delaunay, então a triangulação T é uma triangulação Delaunay.

A aresta é localmente Delaunay testando os dois triângulos que a compartilham. Por exemplo, suponha que a aresta $\overline{AB}$ da triangulação T seja compartilhada pelos triângulos $\triangle ABC$ e $\triangle ABD$, chama-se a aresta $\overline{AB}$ localmente Delaunay se o ponto C estiver fora do círculo que passa através dos pontos A, B e D .

Este trabalho usou o algoritmo Edge-flipping, já definido na biblioteca Scipy do Python, na aplicação da triangulação Delaunay. Primeiro é construído uma triangulação arbitrária do conjunto de pontos e depois o modificando localmente até que todas as arestas passem a ser localmente Delaunay. A ideia é procurar por arestas não localmente Delaunay e virá-las, conforme ilustrado na figura abaixo:

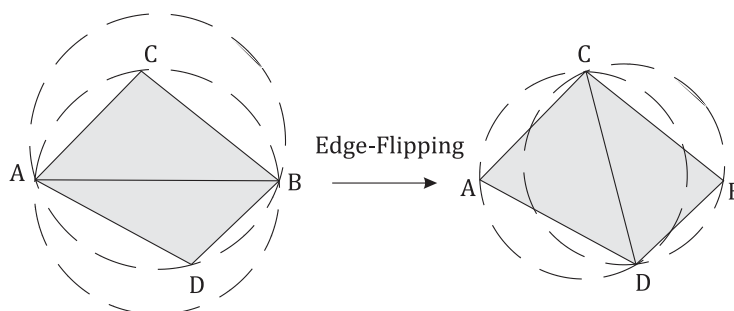


Figura 13 - Algoritmo de Triangulação Delaunay (Edge - Flipping)

Aplicando a triangulação Delaunay nos pontos nodais definidos anteriormente, obtêm-se a malha inicial. Entretanto, a malha gerada não é considerada de alta qualidade, precisando ser suavizada.

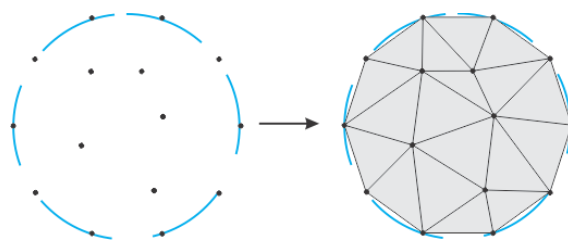


Figura 14 - Triangulação Delaunay

Em alguns casos, a triangulação Delaunay forma triângulos fora do domínio. Se o centroide de um triângulo estiver fora do domínio, esse triângulo é removido.

3.6 Método de suavização adotada

Neste trabalho foi adotada a analogia de mola de segmento na suavização da malha inicial. Já mencionado, essa técnica possui limitações em grandes deslocamentos, ocorrendo colapsos, ou seja, os elementos tendendo a ficarem achatados ou até mesmo a inversão dos elementos:

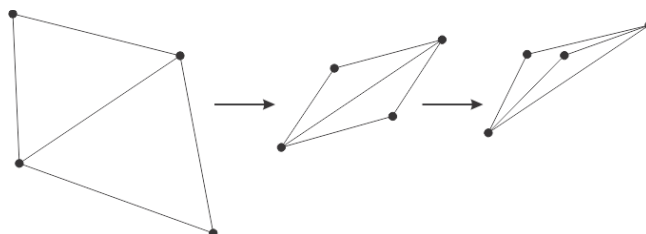


Figura 15 - Problema de inversão dos triângulos

Segundo Blom (2000), esse problema ocorre pela ausência de artifícios impedindo a colisão entre pontos nodais opostos.

A estratégia para contornar esse problema neste trabalho foi a mesma que Persson (2005) publicou: A cada iteração com a suavização, observa-se o quanto obteve de deslocamento acumulativo em relação à malha inicial, ou seja, se o deslocamento acumulativo de qualquer ponto nodal ao longo das iterações for maior que 10% de h , isso indica um deslocamento significativo e aplica-se novamente a triangulação Delaunay para uma outra topologia.

Se o objetivo é obter o comprimento h como o comprimento final das arestas, é preciso uma perturbação nas molas análogas adotando um fator escala F_s sobre h . Segundo Persson (2005) e por tentativas exaustivas realizadas neste trabalho revelam que o F_s adequado em malha bidimensional é de 20% a mais do valor de h , ou seja, o comprimento não estendido da mola - l :

$$l = F_s \cdot h = 1.2 \cdot h \quad (15)$$

4 PROCEDIMENTOS COMPUTACIONAIS

4.1 Python

Essa linguagem possui programação orientada a objetos de alto nível, e não é necessário compilar o código para que a máquina entenda. Por ser uma linguagem interpretada, ela roda em qualquer sistema que possui interpretador Python. Quanto à “estabilidade” da linguagem no mercado, possui uma imensa comunidade ao redor do globo. Ou seja, probabilidades altas de existir bibliotecas complexas e funcionais aos problemas em questão.

Para construção dos algoritmos, foram utilizadas as seguintes bibliotecas: i) Numpy, ii) Scipy, iii) Sympy, iv) Ezdxf e v) Numba.

- i) A biblioteca Numpy possui ferramentas úteis em operações matriciais, ou seja, suporta arrays e matrizes multidimensionais, e um enorme repertório de funções matemáticas.
- ii) Já a biblioteca Scipy foi desenvolvida para trabalhar com a biblioteca Numpy. Fornece códigos prontos de algoritmos já publicados na comunidade científica como, por exemplo, a técnica de triangulação Delaunay.
- iii) Sympy é uma biblioteca para computação simbólica. O pacote possui recursos do simples ao avançado termos algébricos, incluindo as operações trigonométricas.
- iv) O pacote Ezdxf foi construído para ler e manipular arquivos DXF. Através de operações em conjunto com outras bibliotecas, consegue fazer a leitura da geometria do domínio e gerar a malha.
- v) Numba é um compilador de otimização. Foi utilizado para deixar mais eficiente o processo de geração da malha no código.

4.2 Critério de convergência

Na literatura há critérios de convergência de acordo com o método de suavização aplicada. Com a Suavização Laplaciana, por exemplo, o critério de parada está na quantidade de iterações, pois não se observam mudanças significativas após a quarta iteração.

No trabalho de Persson (2005), o critério de parada com a analogia de mola é o deslocamento nodal máximo menor que 10% do comprimento desejado das arestas dos triângulos. Entretanto, para criação do critério de convergência próprio, foi estabelecido um parâmetro de qualidade constante para todos os elementos triangulares, uma vez todos os elementos dentro desse critério, o processo de convergência será ativado.

4.3 Parâmetro de qualidade

A tendência é a produção de triângulos quase equiláteros na malha. Este é uma propriedade desejável ao resolver equações diferenciais parciais – EDP com MEF. Segundo Persson (2005) os erros dependem do menor ângulo, se todos os ângulos forem próximos de 60°, bons resultados são alcançados.

Em 2000, Field (2000) discutiu parâmetros de qualidade de vários elementos. O parâmetro de qualidade muito utilizado em elementos triangulares é a relação entre o raio do círculo inscrito (vezes dois) e o raio do circunscrito.

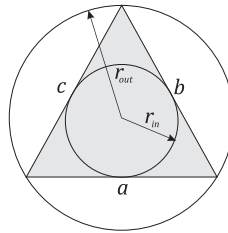
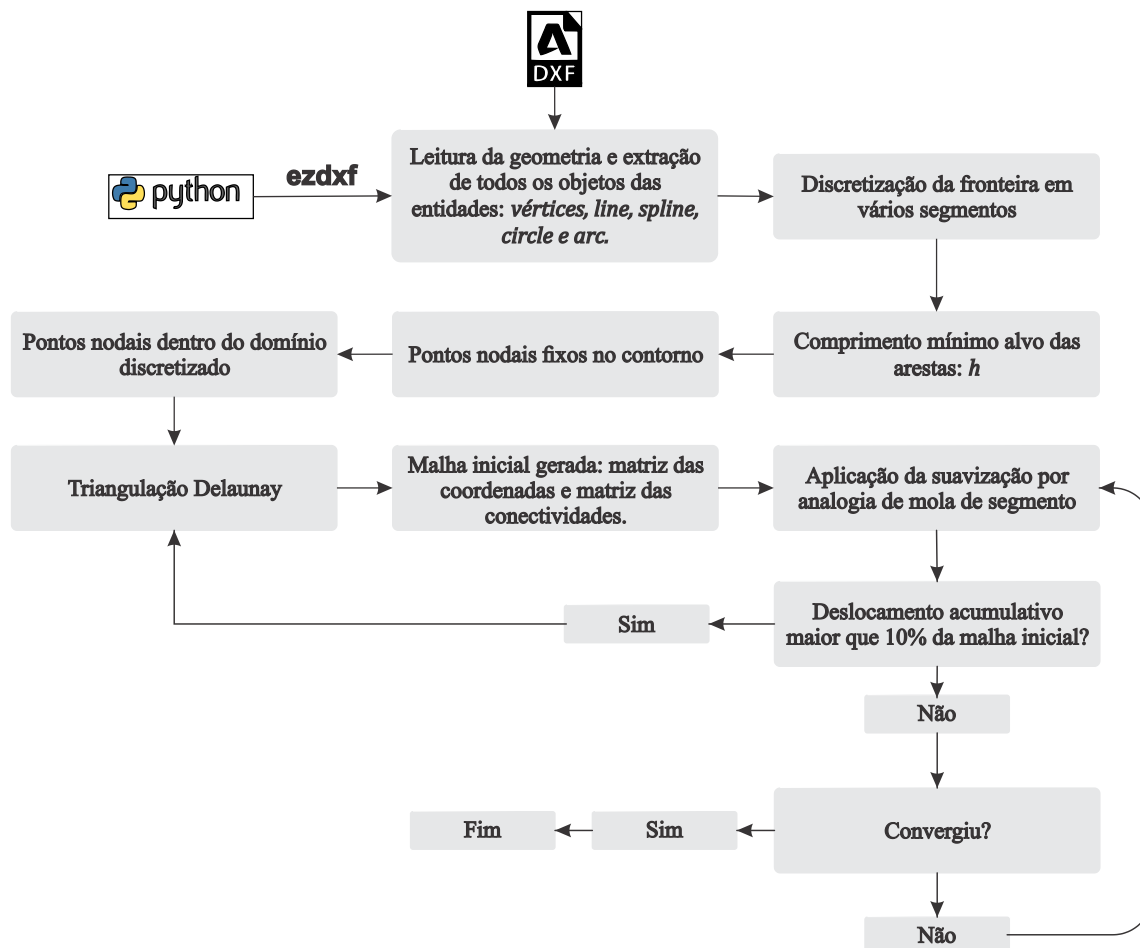


Figura 16 - Parâmetro de qualidade – q

$$q = 2 \frac{r_{in}}{r_{out}} = \frac{(b + c - a)(c + a - b)(a + b - c)}{abc} \quad (16)$$

O triângulo equilátero tem $q = 1$ e um triângulo tendendo a ter ângulos próximos de zero, chega a ter $q \rightarrow 0$. Segundo Persson (2005), como regra geral, se todos os triângulos tiverem $q > 0.5$, os resultados são bons.

4.4 Fluxograma geral do algoritmo final de geração automática de malha bidimensional triangular



5 RESULTADOS

A implementação do algoritmo de suavização por analogia mecânica de mola não é trivial e precisa de um esforço computacional a mais. Este item apresenta a capacidade do código em gerar bons elementos partindo de malhas de má qualidade.

5.1 Exemplo 01

Abaixo segue a malha inicial originada pela triangulação Delaunay:

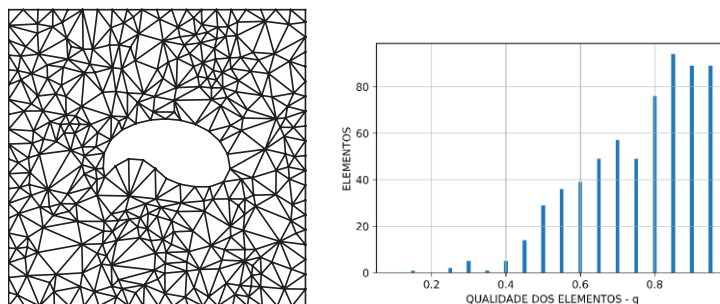


Figura 17 - Malha inicial por triangulação Delaunay

Posteriormente aplica-se quatro vezes a suavização Laplaciana:

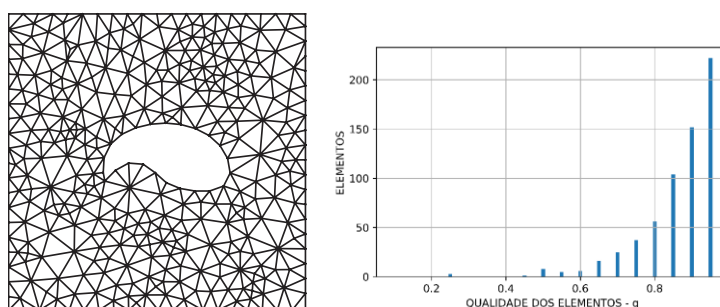


Figura 18 - Malha inicial com a suavização Laplaciana - 4 iterações

Com o intuito de comparar os resultados, partindo da malha inicial da figura 17, aplica-se o algoritmo produto deste trabalho em 1, 2, 3, 4 e 228 iterações:

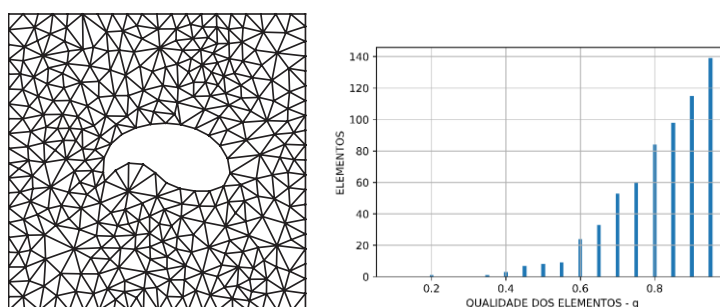


Figura 19 - Malha gerada pelo algoritmo deste trabalho – 1 iteração

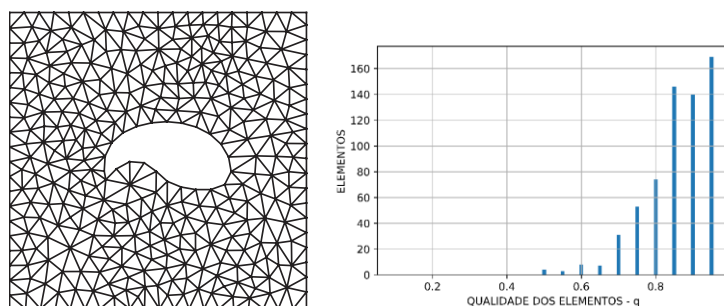


Figura 20 - Malha gerada pelo algoritmo deste trabalho - 4 iterações

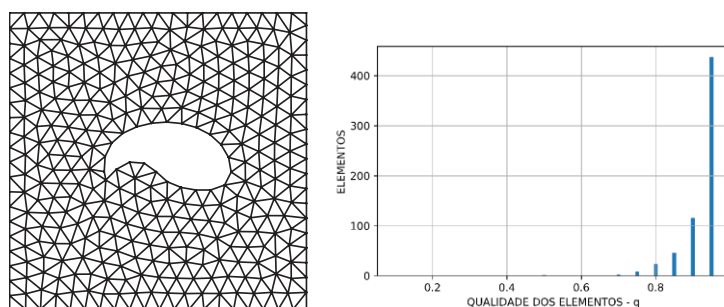


Figura 21 - Malha gerada pelo algoritmo deste trabalho - 228 iterações (em equilíbrio)

Tabela 1 - Comparação dos índices de qualidade das malhas apresentadas

Tipo de malha	Nº de nós	Nº de elem.	Qualidade mín.	Qualidade média
Inicial	369	635	0.1998	0.7859
Lapl. 4 it	369	635	0.2712	0.8890
Alg. 1 it.	369	635	0.2120	0.8425
Alg. 4 it.	369	635	0.5150	0.8831
Alg. 228 it.	369	635	0.5394	0.9540

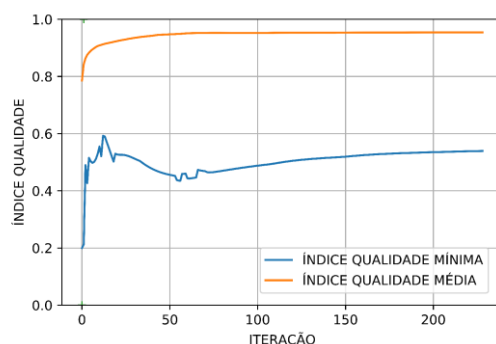


Gráfico 1 - Evolução da qualidade com o número de iterações durante o processo de geração pelo algoritmo

5.2 Exemplo 02

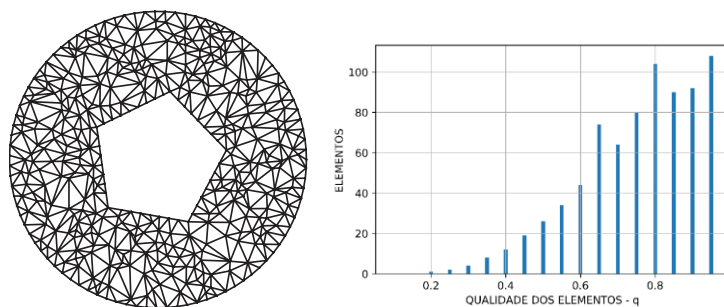


Figura 22 - Malha inicial por triangulação Delaunay

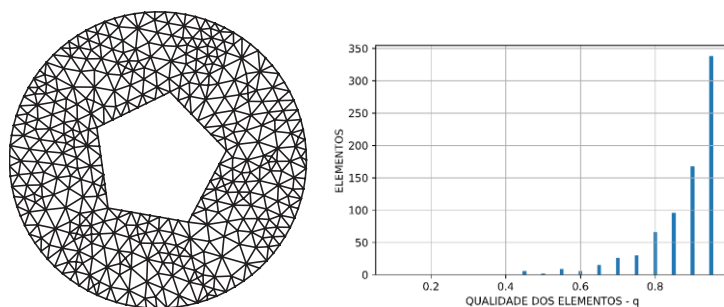


Figura 23 - Malha inicial com a suavização Laplaciana - 6 iterações

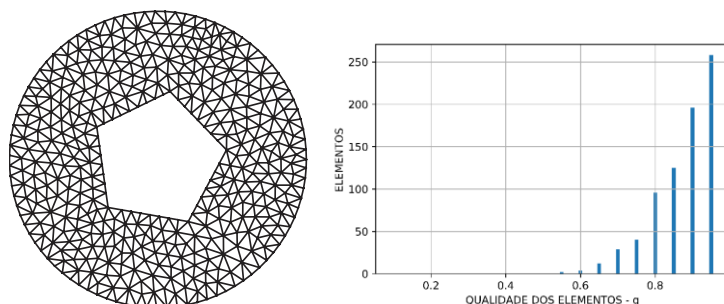


Figura 24 - Malha gerada pelo algoritmo deste trabalho - 6 iterações

Tabela 2 - Comparação dos índices de qualidade das malhas apresentadas

Tipo de malha	Nº de nós	Nº de elem.	Qualidade mín.	Qualidade média
Inicial	447	762	0.2302	0.7788
Lapl. 6 it	447	762	0.4615	0.9059
Alg. 6 it.	447	763	0.5607	0.9001

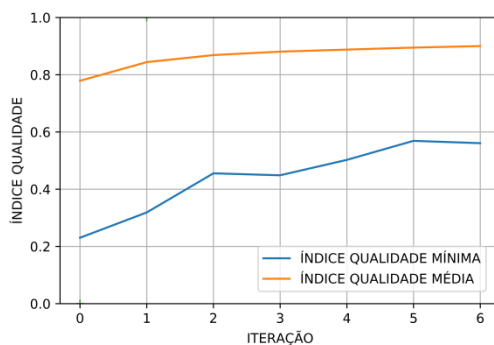


Gráfico 2 - Evolução da qualidade com o número de iterações durante o processo de geração pelo algoritmo

5.3 GiD x Algoritmo

GiD é um software comercial de pré e pós-processamento de simulações numéricas em análise de engenharia. Uma de suas funções é a geração de malhas. Foram geradas duas malhas para o mesmo domínio, uma pelo GiD e outra pelo algoritmo através do grid estruturado.

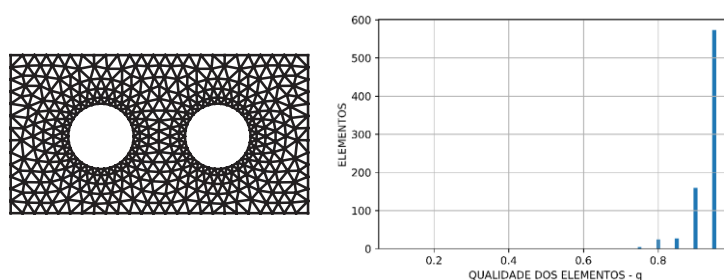


Figura 25 - Malha gerada pelo GiD

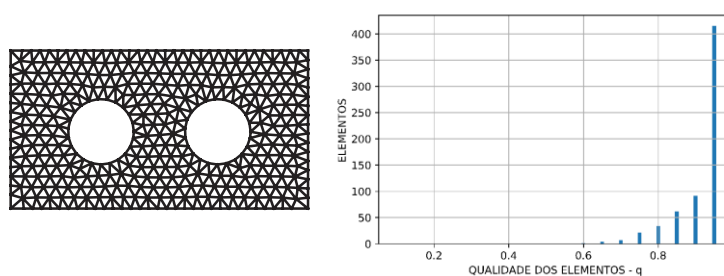


Figura 26 - Malha gerada pelo algoritmo – 175 iterações

Tabela 3 - Comparação dos índices de qualidade das malhas apresentadas

Tipo de malha	Nº de nós	Nº de elem.	Qualidade mín.	Qualidade média
GiD	465	788	0.7613	0.9646
Alg. 175 it.	389	636	0.6280	0.9458

Observa-se que o algoritmo gerou uma malha de alta qualidade apresentando um índice de qualidade mínima de 0.6280. Mas a estratégia eficiente do GiD de refinar a malha em regiões curvilíneas resultou em elementos de maiores qualidades.

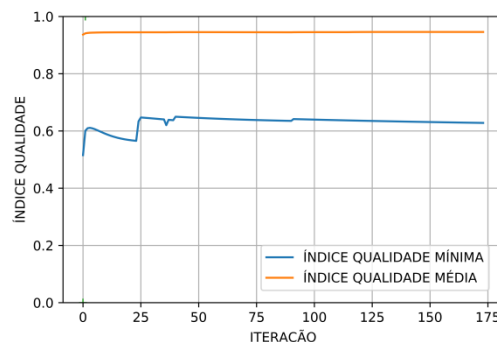


Gráfico 3 - Evolução da qualidade com o número de iterações durante o processo de geração pelo algoritmo

6 CONCLUSÕES

Este trabalho descreve as etapas necessárias para o desenvolvimento de um algoritmo de geração automática de malha triangular bidimensional por analogia mecânica.

O código mostrou que possui a capacidade de uniformizar uma malha de qualidade baixa. Já em relação ao método de suavização Laplaciana, com um número de iterações iguais, o algoritmo obteve resultados superiores.

Quando comparado com o software GiD, nota-se que a proposta atual apresentou desempenho inferior.

O código desenvolvido em PYTHON possui uma integração com qualquer modelador CAD que salve em extensão DXF. Isso facilita no processo de análise para qualquer domínio. Os arquivos de saída são padronizados em vetores: coordenadas dos pontos nodais e suas conectividades. O código não é restrito ao MEF e pode ser aplicado a qualquer finalidade onde a discretização do domínio e geração de malhas suavizadas é necessária.

REFERENCES

- Babuska I. and Aziz A. K. On the angle condition in the Finite Element Method [Journal] // SIAM J.. - 1976. - pp. 214 - 226.
- Batina J. T. Using Unstructured Dynamic Meshes,” Vol. 28, No. 8, [Journal] // AIAA J., Vol 28, No. 8. - 1990. - pp. 1381 - 1388.

Blom Frederic J. Considerations on the spring analogy [Journal] // International Journal for Numerical Methods in Fluids. - 2000. - pp. 647 - 668.

Canann Scott A., Tristano Joseph R. and Staten Matthew L. An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes [Conference] // 7th International Meshing Roundtable. - Dearborn, Michigan, U.S.A. : [s.n.], 1988.

CIMNE GiD versão 12.0.7 - Espanha, 2015.

Delaunay Boris Sur la sphere vide, Bulletin de l'Académie des Sciences de l'URSS [Journal] // Classe des sciences mathématiques et na. - 1934. - pp. 793 - 800.

Dirichlet G. L. Über die reduction der positiven quadratischen formen mit drei unterstimmten [Journal] // Zeitschrift für Angewandte Mathematik und Mechanik. - 1850. - pp. 209–227.

Field David A. Qualitative measures for initial meshes [Journal] // International Journal for Numerical Methods in Engineering. - 2000. - Vol. 47. - pp. 887 - 906.

Herrmann Leonard R. Laplacian-isoparametric grid generation scheme [Journal] // Journal of the Engineering Mechanics Division. - 1976. - pp. 749 - 756.

Lo S. H. A new mesh generation scheme for arbitrary planar domains [Journal] // International Journal for Numerical Methods in Engineering. - 1985. - pp. 1403 - 1426.

Lo S. H. Generation of high-quality gradation finite element mesh [Journal] // Engineering Fracture Mechanics. - 1992. - pp. 191 - 202.

Persson Per-Olof Mesh Generation for Implicit Geometries. - 2005.

Shimrat M. Algorithm 112: Position of point relative to polygon [Journal] // Communications of the ACM . - 1962. - 8 : Vol. 5.

Sibson R. Locally equiangular triangulations [Journal] // The Computer Journal. - 1978.

Voronoi Georgy Feodosevich Nouvelles applications des paramètres continus à la théorie de formes quadratiques [Journal] // Journal für die reine und angewandte Mathematik. - 1908. - pp. 198 - 287.

Zeng D. and Ethier C. R. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains [Journal] // Finite Elem. Anal. Des., Vol. 41, No. 11–12,. - 2005. - pp. 1118 – 1139.