

GRID3D – UM CÓDIGO PARA A GERAÇÃO NUMÉRICA DE MALHAS GENERALIZADAS ESTRUTURADAS

António M. Gameiro Lopes
Universidade de Coimbra
Portugal

António C. Mendes de Sousa,
University of New Brunswick
Canada

Domingos Xavier Viegas
Universidade de Coimbra
Portugal

I - INTRODUÇÃO

A geração numérica de malha é um campo de investigação que tem vindo a ser aperfeiçoado a par com os métodos numéricos para a resolução de problemas de Mecânica Computacional. Na verdade, um bom algoritmo de resolução das equações de Navier-Stokes só pode ser explorado em toda a sua potencialidade, quer em termos de precisão, quer em termos de estabilidade e robustez numérica, se for aplicado numa malha convenientemente adaptada ao problema físico a estudar. Cada situação particular requer uma malha cujas características, em princípio, são dependentes do domínio e do tipo de escoamento esperado.

A geração de uma malha para o cálculo de escoamentos em geometrias que se podem descrever com recurso a uma malha Cartesiana, não requer, de uma forma geral, meios sofisticados de geração de malha. Nestes casos, especifica-se a distribuição de nodos nas fronteiras e procede-se a uma simples interpolação linear para o interior do domínio. Porém, no caso de se tratar de um escoamento em torno de geometrias irregulares, para as quais seja necessário recorrer a um sistema de coordenadas generalizadas, o processo de geração de malha torna-se mais complexo. Ainda que, em alguns casos, uma interpolação da fronteira para o interior do domínio seja suficiente para obter uma malha que preencha os desejados requisitos de estabilidade numérica e precisão da discretização das equações de transporte, na grande maioria das situações torna-se necessário recorrer a métodos de geração mais elaborados.

A derivação das equações de transporte em malhas deformadas requer a transformação das equações originais, escritas no espaço Cartesiano (x,y,z) , para o espaço computacional (ξ,η,ζ) , o qual é definido por um sistema de coordenadas que acompanha as linhas da malha. De forma a que a discretização das equações no espaço computacional seja devidamente executada, a malha deverá preencher os seguintes requisitos:

- Deverá existir uma relação única entre as coordenadas escritas no espaço Cartesiano e as respectivas coordenadas escritas no espaço computacional.
- A malha deverá ser ‘suave’, ou seja, a razão entre as dimensões lineares características de dois volumes de controle adjacentes deve ser próxima da unidade.

- Deve existir uma maior concentração de nodos em regiões em que sejam previstos elevados gradientes das variáveis do escoamento.
- A malha deve ser, em cada nodo, tão ortogonal quanto possível.

O primeiro requisito justifica-se pelo facto de o processo de discretização e integração das equações de transporte se basear no pressuposto de que existe continuidade entre as fronteiras dos volumes de controle sem sobreposição dos mesmos. Os três últimos requisitos destinam-se a manter tão baixos quanto possível os erros de truncatura provenientes da discretização das equações.

Os métodos de geração de malha podem dividir-se, fundamentalmente, em dois tipos: métodos algébricos e métodos numéricos. Thompson, Warsi e Mastin (1982) fazem uma extensa revisão de vários trabalhos publicados sobre estes métodos.

Os métodos algébricos utilizam funções de interpolação para o cálculo das coordenadas dos nodos no interior do domínio. São métodos computacionalmente poucos dispendiosos e daí advém uma grande vantagem no caso da resolução de problemas por recurso a malhas adaptativas. Um dos problemas associados à geração algébrica é a sua tendência para transmitir a informação da fronteira para o interior do domínio, o que poderá ser bastante inconveniente se a fronteira apresentar descontinuidades ou for pouco suave. Alguns métodos algébricos necessitam de simular as fronteiras por intermédio de funções analíticas, que, na maioria dos casos, têm de ser obtidas por processos de aproximação, como *B-splines* ou outros métodos afins. Em suma, a geração analítica apresenta algumas limitações, podendo conduzir a um processo moroso, matematicamente complicado e pouco geral, sobretudo no caso de geometrias tridimensionais.

Os métodos numéricos de geração de malha baseiam-se na resolução de equações diferenciais elípticas, as quais exprimem uma relação matemática entre as coordenadas Cartesianas e as coordenadas computacionais. Apesar de, em termos de esforço de cálculo, serem métodos mais dispendiosos, caracterizam-se por uma elevada robustez e generalidade: Os parâmetros da malha são relativamente fáceis de controlar e as malhas resultantes têm uma ‘suavidade’ característica. O presente trabalho apresenta um método numérico de geração de

malha que resultou num código implementado em *FORTRAN*, disponível na Internet no endereço adaei.dem.uc.pt. Juntamente com o código **GRID3D** disponibiliza-se no mesmo endereço um manual de apoio ao utilizador.

2 – PRINCÍPIOS DA GERAÇÃO NUMÉRICA DE MALHA

A mais simples das equações diferenciais elípticas é a equação de Laplace. O seu emprego para gerar sistemas de coordenadas (geração de Laplace) foi sugerido pela primeira vez em 1962 por Crowley, o qual propôs que as coordenadas dos nodos da malha fossem obtidos através da resolução do seguinte sistema de equações:

$$\begin{aligned} \frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} + \frac{\partial^2 \xi}{\partial z^2} &= 0 \\ \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} + \frac{\partial^2 \eta}{\partial z^2} &= 0 \\ \frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2} + \frac{\partial^2 \zeta}{\partial z^2} &= 0 \end{aligned} \quad (2.1)$$

Como é referido em (Thompson, Warsi & Mastin, 1985), o 'princípio extremo' inerente à solução de sistemas elípticos, garante a relação biunívoca que deve existir entre os dois sistemas de coordenadas. Adicionalmente, a transformação obtida por meio destas equações beneficia das características 'suavisadoras' do operador Laplaciano. Como consequência, e contrariamente ao que sucede com a geração algébrica, as malhas obtidas por meio do sistema de equações (2.1) tendem a esquecer rapidamente a distribuição de pontos das fronteiras, para se aproximarem da transformação conforme no interior do domínio. A Fig. 1 representa uma malha gerada por meio do sistema de equações de Laplace (2.1), onde se encontram patentes as principais características das malhas assim geradas. No caso de a distribuição de pontos na fronteira não ser constante, a malha tende a intersectar as fronteiras obliquamente, o que, para efeitos de aplicação de condições fronteira, é indesejável. Outro inconveniente resultante da geração de Laplace, é a pobre definição da malha junto à fronteira.

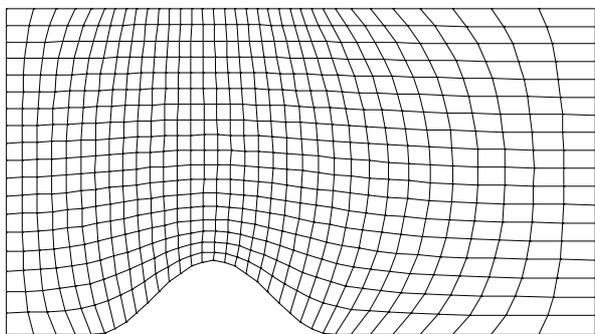


Figura 1 – Malha gerada por meio de um sistema de equações de Laplace.

A ideia de transformar o sistema de equações (2.1) num sistema de equações de Poisson, por meio da adição

de funções de controle (*FC*), foi introduzida em 1972 por Godunov e Prokopov, os quais sugeriram que as *FC* fossem funções do tipo $P(x,y,z)$, *i.e.*, escritas no espaço Cartesiano. Thompson, Thames e Mastin (1974, 1977), propuseram que as *FC* tomassem como variáveis dependentes as coordenadas do espaço computacional, de modo a conseguir-se um mais fácil controle da atracção entre linhas da malha. A forma do sistema de Poisson proposta por estes autores é a seguinte:

$$\begin{aligned} \frac{\partial^2 \xi_i}{\partial x^2} + \frac{\partial^2 \xi_i}{\partial y^2} + \frac{\partial^2 \xi_i}{\partial z^2} &= P_i \\ \xi_1 &= \xi; \xi_2 = \eta; \xi_3 = \zeta \end{aligned} \quad (2.2)$$

Os termos P_i são funções de controle que, dependendo da sua formulação, permitem controlar a malha de diferentes formas: imposição de ortogonalidade e ajuste da concentração das linhas da malha junto às fronteiras, ou transmissão do espaçamento da malha para o interior do domínio. Vários autores têm usado *FC* para controle da malha. Thompson, Thames e Mastin (1977), propuseram funções de controle para conseguir a atracção de linhas da malha para nodos, ou outras linhas da malha, em geometrias bidimensionais. Este processo possui o inconveniente de, por ser definido no espaço computacional, o resultado desejado no espaço físico ser conseguido somente após um processo de erro-tentativa. Um método semelhante foi empregue por Shieh (1982) para geometrias tridimensionais, onde, à semelhança do trabalho anterior, o valor preciso das *FC* só era conseguido por meio de aproximações sucessivas. Thomas (1982), e Thames (1982), geraram malhas tridimensionais em torno de perfis de asa acoplados a carenagens, sendo o controle da concentração das linhas da malha junto às fronteiras conseguido por meio da transmissão do espaçamento da malha para o interior do domínio. O processo seguido por estes autores necessita de uma prévia geração de malhas quase-bidimensionais nas seis faces do domínio, as quais fornecem condições fronteira do tipo Dirichlet para a geração da malha tridimensional. O cálculo das coordenadas nos nodos das fronteiras, para cada uma das seis malhas quase-bidimensionais, é feito algebricamente. Este processo, como é evidente, torna-se bastante moroso, para além de não garantir a ortogonalidade da malha junto às fronteiras, uma condição que deve ser respeitada para o cálculo de escoamentos envolvendo condições fronteira do tipo Neumann. Steger e Sorenson (1979) apresentaram, para duas dimensões, um método bastante mais eficiente para se conseguir uma concentração precisa da malha junto à fronteira, através de um procedimento que somente requer imposição de condições fronteira em duas faces opostas do domínio. O método proposto por estes autores garante também a ortogonalidade das linhas junto à fronteira.

A técnica de geração de malhas tridimensionais que se empregou no presente trabalho, e que será seguidamente exposta, representa uma extensão, para três dimensões, do método proposto por Steger e Sorenson (1979). Incorpora, também, funções de controle propostas por Thomas e Middlecoff (1979). Apesar de o método ter sido

desenvolvido para malhas tridimensionais, por uma questão de facilidade de visualização, os exemplos apresentados envolvem malhas bidimensionais geradas no plano (x, z) .

$$J = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix} \quad (3.7)$$

3 – GERAÇÃO DE MALHA COM O SISTEMA DE POISSON

O sistema de Poisson utilizado no presente trabalho tem a seguinte forma:

$$\begin{aligned} \xi_{xx} + \xi_{yy} + \xi_{zz} &= P_c + P_t \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= Q_c + Q_t \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= S_c + S_t \end{aligned} \quad (3.1)$$

Neste sistema de Poisson, os termos P_c , Q_c e S_c são funções de controle que permitem ajustar a concentração das linhas e ortogonalidade junto das fronteiras. Por sua vez, os termos P_t , Q_t e S_t permitem efectuar a transmissão do espaçamento da malha da fronteira para o interior do domínio.

Uma vez que as incógnitas do sistema de equações (3.1) são as coordenadas Cartesianas, a sua resolução deve ser levada a cabo no espaço computacional. Para tal, efectua-se uma permuta entre as variáveis dependentes e as variáveis independentes, obtendo-se as seguintes equações:

$$\alpha_{11}\vec{r}_{\xi\xi} + \alpha_{22}\vec{r}_{\eta\eta} + \alpha_{33}\vec{r}_{\zeta\zeta} + 2(\alpha_{12}\vec{r}_{\xi\eta} + \alpha_{23}\vec{r}_{\eta\zeta} + \alpha_{31}\vec{r}_{\zeta\xi}) = -J^2[(P_c + P_t)\vec{r}_\xi + (Q_c + Q_t)\vec{r}_\eta + (S_c + S_t)\vec{r}_\zeta] \quad (3.2)$$

onde

$$\vec{r} = (x, y, z)$$

$$\alpha_{ij} = J^2(\nabla_{\xi_i} \cdot \nabla_{\xi_j}) \quad (3.3)$$

$$\xi_1 = \xi ; \xi_2 = \eta ; \xi_3 = \zeta \quad (3.4)$$

As relações métricas α_{ij} na eq. (3.3) são calculadas por intermédio das seguintes equações, escritas no espaço computacional:

$$\alpha_{ij} = \sum_{m=1}^3 A_{mi} A_{mj} \quad (3.5)$$

onde A_{mi} é o co-factor do elemento (m, i) da matriz:

$$\begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \quad (3.6)$$

e o Jacobiano da transformação, J , é dado pelo determinante da matriz (3.6):

3.1 – Funções de controle: Steger e Sorenson

Steger e Sorenson (1979) propuseram um método eficiente para a formulação de funções de controle que permitem impor a distância da primeira linha de nodos à fronteira, para geometrias bidimensionais. O método proposto por estes autores permite, simultaneamente, controlar o grau de ortogonalidade das linhas na fronteira, independentemente da geometria. O código GRID3D emprega esta mesma formulação, numa extensão para 3 dimensões.

Consideremos uma geração de malha tridimensional entre duas fronteiras $\zeta = \zeta_1$ e $\zeta = \zeta_{max}$ nas quais se encontra pré-definida a distribuição de nodos (c.f. Fig. 2).

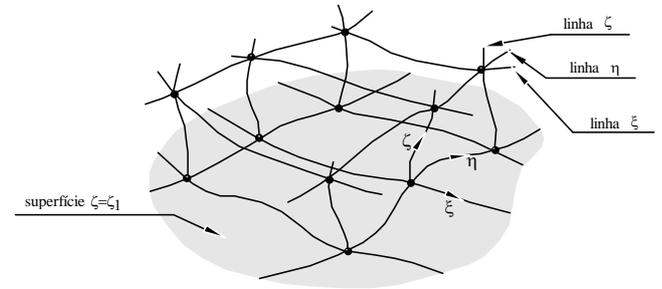


Figura 2 - Representação esquemática do espaço físico junto à fronteira $\zeta = \zeta_1$.

Os termos P_c , Q_c e S_c , que seguidamente serão apresentados, são funções de controle que permitem controlar o grau de atracção das linhas ξ e η (função S_c), bem como a ortogonalidade das linhas ζ (funções P_c e Q_c) próximo das fronteiras $\zeta = c^{te}$. Se a fronteira na qual se pretende fazer o controle é $\zeta = \zeta_1$ (Fig. 2), então será imposto a estas funções um decaimento exponencial para o interior do domínio de acordo com as seguintes fórmulas:

$$P_c = P_1 e^{-a(\zeta - \zeta_1)} \quad (3.8a)$$

$$Q_c = Q_1 e^{-b(\zeta - \zeta_1)} \quad (3.8b)$$

$$S_c = S_1 e^{-c(\zeta - \zeta_1)} \quad (3.8c)$$

onde a , b e c são os chamados factores de decaimento. Os termos P_1 , Q_1 e S_1 representam os valores das funções de controle avaliados na fronteira.

A determinação de P_1 , Q_1 e S_1 efectua-se através do cálculo das eqs. (3.2) na localização $\zeta = \zeta_1$, obtendo-se o seguinte sistema de equações:

$$\begin{aligned}
P_1 &= \frac{(y_\zeta z_\eta - z_\zeta y_\eta)R_1 + (z_\zeta x_\eta - x_\zeta z_\eta)R_2 + (x_\zeta y_\eta - y_\zeta x_\eta)R_3}{J^3} \\
Q_1 &= \frac{(y_\xi z_\zeta - z_\xi y_\zeta)R_1 + (z_\xi x_\zeta - x_\xi z_\zeta)R_2 + (x_\xi y_\zeta - y_\xi x_\zeta)R_3}{J^3} \\
S_1 &= \frac{(z_\xi y_\eta - y_\xi z_\eta)R_1 + (x_\xi z_\eta - z_\xi x_\eta)R_2 + (y_\xi x_\eta - x_\xi y_\eta)R_3}{J^3}
\end{aligned} \quad (3.9)$$

onde:

$$\begin{aligned}
R_i &= \alpha_{11}(x_i)_{\xi\xi} + \alpha_{22}(x_i)_{\eta\eta} + \alpha_{33}(x_i)_{\zeta\zeta} + \\
&+ 2[\alpha_{12}(x_i)_{\xi\eta} + \alpha_{23}(x_i)_{\eta\zeta} + \alpha_{31}(x_i)_{\zeta\xi}]
\end{aligned} \quad (3.10)$$

$$x_1 = x ; x_2 = y ; x_3 = z$$

De notar que, nesta derivação, os termos P_i , Q_i e S_i foram considerados nulos.

Nas equações apresentadas, as derivadas em ξ e η são conhecidas, uma vez que as distribuições dos nodos na superfície ζ_1 é especificada como condição fronteira. As derivadas em ζ são calculadas como função da distância Δs da primeira superfície de nodos $\zeta = \zeta_2$ à fronteira $\zeta = \zeta_1$. Esta distância é imposta *a priori* como condição fronteira e pode ser escrita recorrendo aos métricos covariantes, da seguinte forma:

$$\Delta s = \sqrt{(x_\zeta)^2 + (y_\zeta)^2 + (z_\zeta)^2} \quad (3.11)$$

As duas outras condições necessárias para a determinação das derivadas em ζ são obtidas através da imposição de ortogonalidade das linhas ζ com a fronteira. Esta condição é satisfeita pela ortogonalidade entre os vectores base covariantes \vec{r}_ξ e \vec{r}_ζ , assim como entre \vec{r}_η e \vec{r}_ζ , ou seja:

$$\begin{aligned}
\vec{r}_\xi \cdot \vec{r}_\zeta &= (x_\xi \hat{i} + y_\xi \hat{j} + z_\xi \hat{k})(x_\zeta \hat{i} + y_\zeta \hat{j} + z_\zeta \hat{k}) = \\
&= x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta = 0
\end{aligned} \quad (3.12)$$

$$\begin{aligned}
\vec{r}_\eta \cdot \vec{r}_\zeta &= (x_\eta \hat{i} + y_\eta \hat{j} + z_\eta \hat{k})(x_\zeta \hat{i} + y_\zeta \hat{j} + z_\zeta \hat{k}) = \\
&= x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta = 0
\end{aligned}$$

Combinando as eqs. (3.11) e (3.12) obtêm-se as seguintes expressões para as derivadas segundo ζ :

$$\begin{aligned}
x_\zeta &= \frac{(z_\eta y_\xi - y_\eta z_\xi) \Delta s}{A} \\
y_\zeta &= \frac{(x_\eta z_\xi - z_\eta x_\xi) \Delta s}{A} \\
z_\zeta &= \frac{(y_\eta x_\xi - x_\eta y_\xi) \Delta s}{A}
\end{aligned} \quad (3.13)$$

onde:

$$A = \sqrt{(z_\eta y_\xi - y_\eta z_\xi)^2 + (x_\eta z_\xi - z_\eta x_\xi)^2 + (y_\eta x_\xi - x_\eta y_\xi)^2} \quad (3.14)$$

A avaliação das segundas derivadas em ζ na fronteira $\zeta = \zeta_1$ (eqs. 3.10), é feita por recurso a uma discretização de segunda ordem:

$$\vec{r}_{\zeta\zeta} = \left(\frac{-7\vec{r}_1 + 8\vec{r}_2 - \vec{r}_3}{2\Delta\zeta^2} \right) - \frac{3\vec{r}_{\zeta 1}}{\Delta\zeta} \quad (3.15)$$

onde o índice 1, 2, 3 referencia a posição do nodo relativamente à fronteira, ao longo da linha ζ .

As funções de controle correspondentes para outras fronteiras são obtidas por um processo análogo ao apresentado.

As equações anteriormente apresentadas permitem a determinação das funções de controle P_c , Q_c e S_c , como função do parâmetro Δs . Os factores de decaimento \underline{a} e \underline{b} nas eqs. (3.8), actuando sobre as linhas ζ , permitem ajustar a propagação da condição de ortogonalidade para dentro do domínio. O coeficiente \underline{c} , nas mesmas equações, controla o efeito de atracção das linhas ξ e η para a fronteira. Valores típicos para estes coeficientes situam-se na gama de 0.2 a 1.0. Como exemplo, a Fig. 3 representa uma malha gerada em torno da colina sinusoidal apresentada na Fig.1, mas desta vez com recurso às funções de controle. Como se pode observar, conseguiu-se impor um espaçamento constante das primeiras linhas de nodos até à fronteira inferior e a intersecção das linhas com a colina faz-se de uma forma ortogonal.

A influência do coeficiente de decaimento \underline{c} pode ser apreciada na Fig. 4, onde a geração de malha foi efectuada recorrendo a um maior valor para este parâmetro. Neste caso, o efeito de atracção é menos pronunciado dentro do domínio, se bem que o espaçamento da primeira linha de nodos até à fronteira permaneça inalterado.

O efeito do coeficiente de decaimento \underline{a} (o coeficiente \underline{b} reporta-se à terceira dimensão) pode ser observado na Fig. 5. Esta malha foi gerada utilizando os mesmos parâmetros da malha da Fig. 3, à excepção do valor de \underline{a} , o qual foi diminuído de 0.4 para 0.2. Neste caso, a condição de ortogonalidade transmite-se a uma maior distância dentro do domínio.

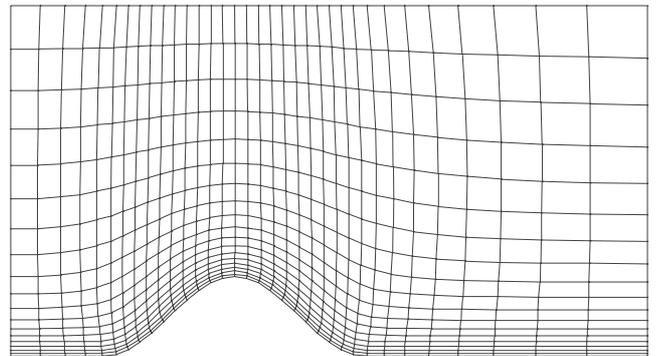


Figura 3 - Malha gerada em torno de uma colina sinusoidal. ($a=0.5$, $c=0.4$).

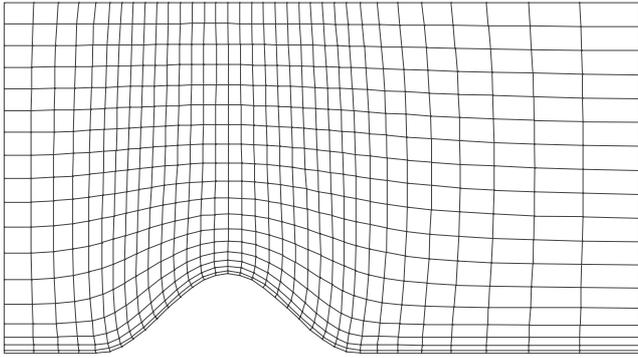


Figura 4 - Malha gerada em torno de uma colina sinusoidal. ($a=0.5$, $c=1.0$).

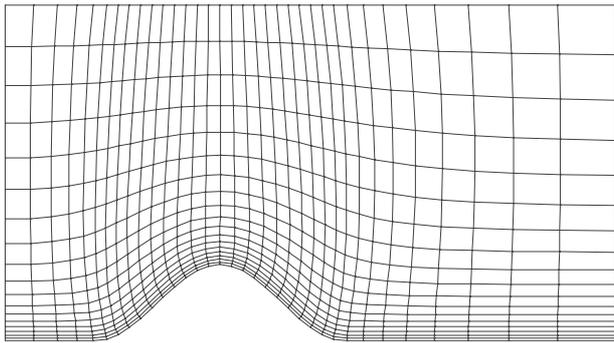


Figura 5 - Malha gerada em torno de uma colina sinusoidal. ($a=0.2$, $c=0.4$).

3.2 – Funções de controle: Thomas e Middlecoff

As malhas anteriormente apresentadas foram geradas incorporando também termos que permitem a transmissão do espaçamento da fronteira para o interior do domínio. A necessidade destas funções de controle deriva do facto de as equações de Laplace sem formulação especial tenderem a uniformizar a distribuição de nodos, devido ao seu efeito 'suavizador'. As consequências que daí advêm podem ser apreciadas na Fig. 6, onde é representada uma malha gerada sem recurso a estas funções, empregando somente os termos de Steger e Sorenson.

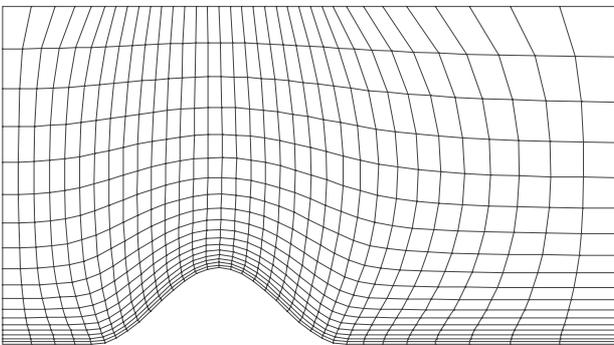


Figura 6 - Malha gerada em torno de uma colina sinusoidal, sem transmissão do espaçamento da malha para o interior do domínio ($a=0.5$, $c=0.4$).

Pode-se notar o maior enviesamento da malha, consequente da distribuição não uniforme de nodos na fronteira. A formulação das funções de controle que permite corrigir este problema foi proposta por Thomas e Middlecoff (1979), da seguinte forma (c.f. eq. 3.1):

$$P_i = \frac{\alpha_{11} \phi}{J^2} \quad (3.16)$$

$$Q_i = \frac{\alpha_{22} \varphi}{J^2} \quad (3.17)$$

$$S_i = \frac{\alpha_{33} \omega}{J^2} \quad (3.18)$$

onde

$$\phi = \frac{\vec{r}_\xi \cdot \vec{r}_{\xi\xi}}{\vec{r}_\xi \cdot \vec{r}_\xi} \quad (3.19)$$

$$\varphi = \frac{\vec{r}_\eta \cdot \vec{r}_{\eta\eta}}{\vec{r}_\eta \cdot \vec{r}_\eta} \quad (3.20)$$

$$\omega = \frac{\vec{r}_\zeta \cdot \vec{r}_{\zeta\zeta}}{\vec{r}_\zeta \cdot \vec{r}_\zeta} \quad (3.21)$$

As funções de controle P_i , Q_i e S_i transmitem o espaçamento segundo as direcções computacionais ξ , η e ζ , respectivamente. Se, por exemplo, se pretender transmitir o espaçamento das linhas ξ , segundo a direcção ζ , o termo ϕ deve ser calculado nas localizações ζ_1 e ζ_{max} . Os correspondentes valores nos nodos do interior do domínio são obtidos por meio de interpolação. É importante notar que, ao contrário do termo ϕ , os termos α_{11} e J^2 não são calculados nas fronteiras, seguindo-se a interpolação. Estes termos são calculados em cada nodo, seguindo-se a aplicação da eq. (3.16) para cálculo da função de controle P_i . Se, adicionalmente, se desejar transmitir o espaçamento em ξ ao longo das linhas η , os valores de ϕ em cada nodo no interior do domínio são obtidos somando simplesmente as contribuições vindas da interpolação entre os dois pares de fronteiras.

4 – RESOLUÇÃO DO SISTEMA DE EQUAÇÕES

Substituindo as eqs. (3.16-18) na eq. (3.2) e rearranjando os termos, obtém-se:

$$\alpha_{11} \vec{r}_{\xi\xi} + (J^2 P_c + \alpha_{11} \phi) \vec{r}_\xi + \alpha_{22} \vec{r}_{\eta\eta} + (J^2 Q_c + \alpha_{11} \psi) \vec{r}_\eta + \alpha_{33} \vec{r}_{\zeta\zeta} + (J^2 R_c + \alpha_{11} \omega) \vec{r}_\zeta + 2(\alpha_{12} \vec{r}_{\xi\eta} + \alpha_{23} \vec{r}_{\eta\zeta} + \alpha_{31} \vec{r}_{\zeta\xi}) = 0 \quad (4.1)$$

Esta é o sistema cuja resolução permite obter as coordenadas (x,y,z) de cada nodo da malha. Por discretização deste sistema, obtém-se:

$$a_p \vec{r}_p = a_E \vec{r}_E + a_O \vec{r}_O + a_N \vec{r}_N + a_S \vec{r}_S + a_T \vec{r}_T + a_B \vec{r}_B + b \quad (4.2)$$

onde os índices E, O, N, S, T, B se referem à posição relativa dos nodos no espaço computacional: Este, Oeste, Norte, Sul, Topo e Baixo, ou seja:

$$\vec{r}_p = (x, y, z)_{i,j,k} \quad (4.3a)$$

$$\vec{r}_E = (x, y, z)_{i+1,j,k} \quad \vec{r}_O = (x, y, z)_{i-1,j,k} \quad (4.3b)$$

$$\vec{r}_N = (x, y, z)_{i,j+1,k} \quad \vec{r}_S = (x, y, z)_{i,j-1,k} \quad (4.3c)$$

$$\vec{r}_T = (x, y, z)_{i,j,k+1} \quad \vec{r}_B = (x, y, z)_{i,j,k-1} \quad (4.3d)$$

Neste conjunto de equações, as segundas derivadas são calculadas por meio de diferenças centradas. Os termos envolvendo primeiras derivadas são avaliados utilizando diferenças à direita ou à esquerda, dependendo do sinal do coeficiente correspondente. Trata-se de um procedimento destinado a manter os coeficientes das equações positivos e assim melhorar a estabilidade numérica do processo iterativo. Exemplificando, para os termos correspondentes à derivadas segundo a direcção computacional ξ , vem:

$$\text{Se } (J^2 P_c + \alpha_{11} \phi) > 0$$

$$a_E = \alpha_{11} + (J^2 P_c + \alpha_{11} \phi) \quad \text{e} \quad a_O = \alpha_{11} \quad (4.4a)$$

$$\text{Se } (J^2 P_c + \alpha_{11} \phi) < 0$$

$$a_E = \alpha_{11} \quad \text{e} \quad a_O = \alpha_{11} - (J^2 P_c + \alpha_{11} \phi) \quad (4.4b)$$

O coeficiente para a localização central é:

$$a_p = a_E + a_O + a_N + a_S + a_T + a_B \quad (4.5)$$

e o termo fonte tem a seguinte formulação:

$$b = 2(\beta_1 \vec{r}_{\xi\eta} + \beta_2 \vec{r}_{\eta\xi} + \beta_3 \vec{r}_{\zeta\xi}) \quad (4.6)$$

Sub-relaxação das funções de controle

Os termos de Thomas e Middlecoff não necessitam de sub-relaxação. Os termos de Steger e Sorenson introduzem grande instabilidade no estágio inicial do processo iterativo, pelo que têm de ser muito sub-relaxados. Esta sub-relaxação pode ser reduzida ao longo do processo iterativo. O método implementado em GRID3D consiste em fazer variar o factor de sub-relaxação γ de uma forma exponencial, de acordo com a seguinte equação:

$$\gamma = \frac{10^{\left(\beta \frac{n}{n_{max}}\right)}}{10^\beta} \gamma_{max} \quad (4.7)$$

onde β representa a taxa de incremento, n é o número da presente iteração e γ_{max} é o valor máximo de γ alcançado na iteração n_{max} . A dependência de γ com β encontra-se representada no gráfico da Fig. 7, para $n_{max} = 100$ e

$\gamma_{max} = 0.25$. Os valores mais altos de β (normalmente não excedendo 7) são recomendados para problemas nos quais Δs (distância à fronteira) é muito baixo.

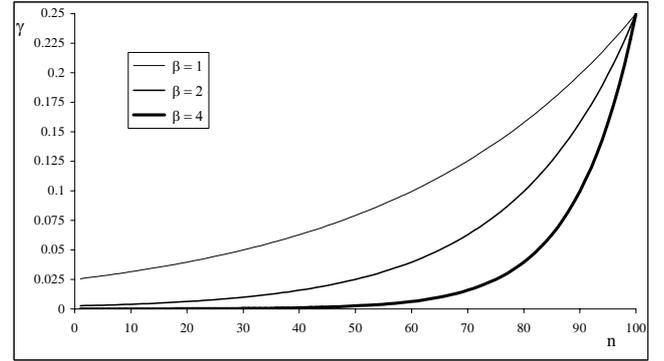


Figura 7 – Dependência do factor de sub-relaxação γ com o parâmetro β .

Relaxação do processo iterativo

A resolução das eqs. (4.2) é feita utilizando o método PSOR (*Point Successive Over Relaxation*). O factor de relaxação óptimo é calculado em cada ponto da malha de acordo com o procedimento proposto por Ehrlich (1981).

Considerem-se as eqs. (4.2) escritas na seguinte forma :

$$-a_p \vec{r}_p + a_E \vec{r}_E + a_O \vec{r}_O + a_N \vec{r}_N + a_S \vec{r}_S + a_T \vec{r}_T + a_B \vec{r}_B = -b \quad (4.8)$$

Os valores próprios complexos desta equações são:

$$\begin{aligned} \mu &= \mu_r + \mu_i = \\ &= \frac{2}{-a_p} \left(\sqrt{a_E a_O} \cos \frac{\pi}{i_{max}} + \sqrt{a_N a_S} \cos \frac{\pi}{j_{max}} + \sqrt{a_T a_B} \cos \frac{\pi}{k_{max}} \right) \end{aligned} \quad (4.9)$$

onde μ_r e μ_i representam a parte real e a parte imaginária de μ , respectivamente. No presente caso, a parte imaginária é nula, uma vez que foi assegurada a inexistência de coeficientes negativos. Prosseguindo, definem-se as seguintes quantidades:

$$A = \mu_r^2 + \mu_i^2 \quad (4.10a)$$

$$B = \mu_r^2 - \mu_i^2 \quad (4.10b)$$

$$C = A^2 - B^2 \quad (4.10c)$$

$$D = A^2 - B \quad (4.10d)$$

$$E = \sqrt{C + D^2} \quad (4.10e)$$

$$F = \sqrt[3]{C} \quad (4.10f)$$

e

$$\bar{\kappa} = \frac{(3D+E)F\sqrt[3]{E-D} - (3D-E)F\sqrt[3]{E+D+A^2+3B^2-4A^2B}}{A^2D} \quad (4.11)$$

O factor de relaxação κ é dado por:

$$\kappa = \begin{cases} \frac{\left(\bar{\kappa} - \sqrt{\bar{\kappa}^2 + 4\bar{\kappa}}\right)}{2} & \text{if } D > 0 \\ \frac{\left(\bar{\kappa} + \sqrt{\bar{\kappa}^2 + 4\bar{\kappa}}\right)}{2} & \text{if } D < 0 \end{cases} \quad (4.12)$$

Esta formulação conduz a um factor de sub- ou sobre- relaxação que varia de ponto para ponto, permitindo aumentar consideravelmente a velocidade de convergência do processo. Os valores de κ assim calculados podem, contudo, ser por vezes demasiado altos, pelo que é recomendável aplicar um factor de redução, para o qual se toma, normalmente, o valor de 0.7.

Processo de resolução das equações

O campo inicial de variáveis (coordenadas x, y, z), necessário para iniciar o processo de cálculo, é obtido por interpolação linear dos valores nas fronteiras. Após algumas iterações feitas sem uso das funções de controle, estas são introduzidas gradualmente de acordo com a eq. (4.7), no caso das funções de Steger e Sorenson. Não se encontra implementado nenhum critério de paragem das equações. Especifica-se *a priori* o número máximo de iterações, sendo monitorizadas as coordenadas de um ponto da malha seleccionado. Os tempos de cálculo são modestos. Como exemplo, mostra-se na Fig. 8, alguns planos seleccionados de uma malha de 60x30x30 nodos, gerada em torno de uma colina sinusoidal tridimensional, cujo cálculo demorou cerca de 70 segundos num PC Pentium II 600MHz.

5 – CONCLUSÕES

Apresentou-se um método numérico para geração de malhas tridimensionais estruturadas, o qual permite o controle das características da malha junto às fronteiras do domínio. O presente método, baseado no mapeamento do espaço físico e computacional por meio de equações de Poisson, resulta numa distribuição de nodos progressiva e suave, tornando-o particularmente apto para aplicações de Mecânica de Fluidos e Transmissão de Calor. O software que implementa as equações expostas no presente trabalho encontra-se disponível na Internet, tendo já sido exaustivamente testado para a geração de malha aplicada a vários problemas na área de Mecânica dos Fluidos.

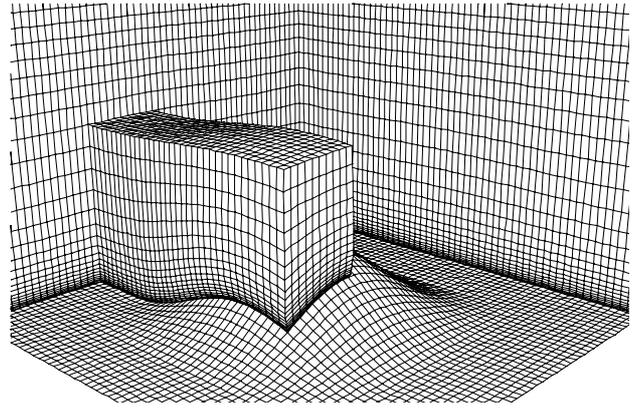


Figura 8 – Malha tridimensional.

6 – REFERÊNCIAS

- Crowley, W.P. (1962) - "An Equipotential Zoner on a Quadrilateral Mesh", memorandum, Lawrence Livermore National Laboratory, July 5.
- Ehrlich, L. W. (1981) – "An Ad Hoc SOR Method", Journal of Computational Physics, Vol. 44, pp. 31-45.
- Shieh, C.F. (1982) - "Three-Dimensional Grid Generation Using Poisson Equations", Applied Mathematics and Computation, Vol. 11, pp. 687-694.
- Steger, J.L. and Sorenson, R.L. (1979) - "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations", Journal of Computational Physics, Vol. 33, pp. 405-410.
- Thames, F.C. (1982) - "Generation of Three-Dimensional Boundary-Fitted Curvilinear Coordinate Systems For Wing/Wing-Tip Geometries Using The Elliptic Solver Method", Applied Mathematics and Computation, Vol. 11, pp. 695-716
- Thomas, P.D. (1982) - "Numerical Generation of Composite Three Dimensional Grids By Quasilinear Elliptic Systems", Applied Mathematics and Computation, Vol. 11, pp. 667-686.
- Thomas, P.D. and Middlecoff, J.F. (1979) - "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", AIAA Journal, Vol. 18, N. 6, pp. 652-656.
- Thompson, J.F, Warsi, Z.U.A. and Mastin, C.W. (1982) "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review", Journal of Computational Physics, Vol. 47, N. 1.
- Thompson, J.F., Thames, F.C. and Mastin, C.W. (1974) - "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies", J. Computational Physics, Vol. 15, pp. 299-319.

- Thompson, J.F., Thames, F.C. and Mastin, C.W. (1977)
 - "TOM-CAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies", J. Computational Physics, Vol. 24, pp. 274-302.
- Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W. - (1985) - *Numerical Grid Generation: Foundations and Applications*, North-Holland, Amsterdam.