

## OTIMIZAÇÃO DE MALHAS MULTIBLOCOS BIDIMENSIONAIS

**Márcio Ricardo Pivello**

Universidade Federal de Uberlândia – Faculdade de Engenharia Mecânica  
[pivello@mecanica.ufu.br](mailto:pivello@mecanica.ufu.br)

**Carlos Roberto Ribeiro**

Universidade Federal de Uberlândia – Faculdade de Engenharia Mecânica  
[cribeiro@mecanica.ufu.br](mailto:cribeiro@mecanica.ufu.br)

**Resumo.** Este trabalho apresenta uma metodologia para otimização de malhas multiblocos sobre domínios bidimensionais. Malhas deste tipo são particularmente úteis na resolução de problemas de escoamentos sobre domínios complexos. Como via de regra estes problemas são resolvidos via volumes finitos, a transição suave da malha entre blocos adjacentes auxiliará bastante a convergência da solução. A geração de uma malha multiblocos é feita em duas etapas: primeiro o domínio é dividido em subdomínios quadriláteros e então os blocos são discretizados por malhas estruturadas. Neste trabalho os blocos serão gerados manualmente. Para que se possa obter bons resultados com uma malha, algumas condições devem ser observadas: para blocos adjacentes, as coordenadas dos pontos da malha localizados na interface devem ser as mesmas para os dois blocos. Além disso, como etapas posteriores deste trabalho envolvem utilização desta malha em problemas de CFD resolvidos por volumes finitos, é importante também que haja continuidade nas derivadas de 1<sup>ª</sup> e 2<sup>ª</sup> ordens nesta interface. Para isso, utilizamos como suavizador um operador laplaciano formulado para coordenadas curvilíneas. O processo de suavização consiste em realocar os nós de fronteira de cada bloco de modo a se obter elementos geometricamente regulares. As coordenadas destes nós servem como condições de contorno de Dirichlet para o laplaciano, que reposicionará os nós internos ao domínio. Para garantir a continuidade  $C^2$  na transição da malha entre blocos vizinhos, os pontos localizados na interface são tratados como pontos internos ao bloco, ou seja: o operador laplaciano é estendido até os pontos de fronteira, que passam a ser os pontos da última linha / coluna da matriz, e os pontos da primeira linha / coluna do bloco vizinho na interface passam a ser as condições de contorno do problema. A representação das fronteiras do domínio foi feita usando-se spline cúbica com formulação paramétrica. Os resultados apresentados mostram que a fusão dos processos de suavização da malha e dos blocos fornece resultados satisfatórios e facilita a implementação do algoritmo.

**Palavras chave:** malhas multibloco, suavização de malhas, laplaciano, malhas bidimensionais

### 1. Introdução

Na simulação de qualquer problema físico sobre domínios complexos, a geração da malha é uma das etapas mais importantes. Do ponto de vista de discretização do domínio, a melhor opção é a malha não estruturada, devido a sua fácil adaptação a geometrias complexas e facilidade de implementação do algoritmo. Entretanto, a discretização das equações é tão importante quanto a discretização do domínio, e muitas vezes determina que tipo de malha será utilizada. Para problemas de CFD, por exemplo, a maior parte dos algoritmos de solução das equações de Navier – Stokes está formulada e devidamente testada para malhas estruturadas. Entretanto, este tipo de malha dificilmente pode ser empregado em geometrias complexas. Uma alternativa é o uso de malhas multiblocos (ou estruturadas por blocos).

Malhas multiblocos podem ser classificadas como uma alternativa intermediária entre malhas estruturadas e não estruturadas. O método consiste em dividir o domínio em subdomínios quadriláteros e discretizar cada um segundo uma malha estruturada. O problema físico é então resolvido localmente em cada bloco, e algum método para garantir a continuidade da solução entre os blocos é aplicado. Este tipo de discretização talvez seja hoje o mais empregado na resolução de problemas de CFD, mas ainda há alguns pontos que são alvos de pesquisa: a automatização da geração dos blocos, [BADCOCK et al., 2000, PARK et. al., 1997, DANNENHOFFER, 1995], e a continuidade na solução entre blocos vizinhos [ENANDER, STERNER, 1996], [LIU, SHYY, 1996]. Estes dois problemas estão intimamente relacionados, pois a precisão nos resultados depende da regularidade da malha e do método utilizado na transferência de informação entre blocos vizinhos. A precisão na transferência de informação entre vizinhos depende, por sua vez, entre outros fatores, da suavidade da transição da malha nas fronteiras dos blocos e da ortogonalidade da malha nestas fronteiras [de NICOLA, 1997; OOSTERLEE, 1995]. Neste trabalho nos concentraremos na segunda classe de problemas, buscando otimizar a forma de uma dada topologia de multiblocos. Uma vez que em etapas posteriores este método será aplicado a problemas de escoamento externo a um corpo imerso em um fluido, serão analisados casos de domínios multiplamente conexos, por ser esta a configuração típica do domínio que representa este tipo de problema. A Figura (1) mostra um exemplo típico de um domínio que representa o escoamento em torno de dois cilindros. Temos, à esquerda, a representação do domínio de cálculo, e à direita a sua representação por uma topologia multiblocos.

De um modo geral, buscam-se blocos com faces ortogonais entre si, que sejam tão retangulares quanto possível e que se concentrem em torno do domínio estrutural, ou seja, que haja um refinamento dos blocos nas proximidades da

estrutura, devido aos efeitos de viscosidade que surgem na interface fluido – sólido. Isto pode ser feito diretamente sobre os blocos, numa etapa anterior à geração da malha [GRIBBEN, 1999] ou como parte do processo de suavização da malha, utilizando um laplaciano em coordenadas curvilíneas, por exemplo [PIVELLO, 2001, RIBEIRO, 2000]. Quando se faz esta suavização anterior à geração da malha, é necessário que se defina uma função objetivo complexa, de modo a se satisfazer estes quesitos. Já o emprego do Laplaciano como suavizador garante imediatamente a segunda e a terceira condições enumeradas acima. Somente a busca da condição de ortogonalidade da malha é feita através de uma função objetivo, que no caso deste trabalho avalia a diferença entre os elementos localizados na fronteira de cada bloco e um quadrado.

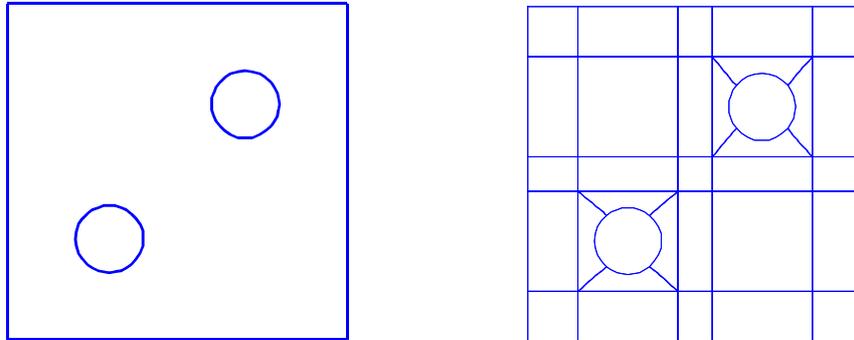


Figura 1: Exemplo do domínio de um escoamento em torno de dois cilindros (esquerda) e respectiva configuração de multiblocos(direita).

O restante deste trabalho está organizado como se segue: na seção 2, apresentamos uma formulação paramétrica da spline cúbica, usada tanto na caracterização do domínio quanto na aplicação das condições de contorno. A seção 3 traz a formulação do operador laplaciano utilizado como suavizador da malha, segundo um sistema de coordenadas curvilíneas genérico. O algoritmo de suavização da malha é apresentado em detalhes na seção 4. Na seção 5 apresentamos os resultados para dois domínios estudados e as conclusões são apresentadas na seção 6. As referências bibliográficas são apresentadas na seção 7.

## 2. Representação do Domínio

Para a representação da fronteira do domínio neste trabalho foi utilizada uma formulação paramétrica da spline cúbica, baseada na analogia da viga de Euler-Bernoulli (Rogers, 1990). Segundo esta formulação, as coordenadas de um ponto sobre uma curva representada por uma spline são dadas por

$$[x(t) \ y(t)] = [T][N][Gx \ Gy] \tag{1}$$

onde

$$[T] = [t^3 \ t^2 \ t \ 1], \quad 0 \leq t \leq 1 \tag{2}$$

$$[N] = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$[Gx \ Gy]^T = \begin{bmatrix} x_k & x_{k+1} & x'_k & x'_{k+1} \\ y_k & y_{k+1} & y'_k & y'_{k+1} \end{bmatrix} \tag{4}$$

e os valores de  $x_k'$ ,  $k=1$  a  $n$ , são obtidos para uma spline natural resolvendo-se o sistema:

$$\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \vdots \\ \vdots \\ \vdots \\ P_n' \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 4 & 1 & 0 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & 4 & 1 \\ \cdot & \cdot & \cdot & \cdot & 0 & 0 & 2 & 4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1.5(P_2 - P_1) \\ 3\{(P_2 - P_1) + (P_1 - P_0)\} \\ 3\{(P_3 - P_2) + (P_2 - P_1)\} \\ \cdot \\ \cdot \\ \cdot \\ 3\{(P_n - P_{n-1}) + (P_{n-1} - P_{n-2})\} \\ 6(P_n - P_{n-1}) \end{bmatrix} \quad (5)$$

onde  $P'$  tem componentes  $x$  e  $y$ .

### 3. Operador Laplaciano em Coordenadas Curvilíneas

O emprego de um operador Laplaciano na geração ou suavização de malhas estruturadas tem, como principal vantagem, o fato de que as soluções destes sistemas são funções harmônicas (ou seja, suas derivadas parciais de segunda ordem são contínuas) cujos valores máximo e mínimo ocorrem nas fronteiras. Este fato, além de garantir que os pontos de máximo e mínimo não anularão o Jacobiano da transformação, também garante a unicidade das soluções, isto é, duas superfícies coordenadas de mesmo valor nunca se interceptarão, o que é requisito básico para a geração de malhas estruturadas. Além disso, todos os problemas de campo são regidos por estas equações e possuem como soluções, portanto, iso-superfícies que podem ser empregadas como superfícies coordenadas (Maliska, 1995).

O método empregado neste trabalho consiste em fazer um mapeamento conforme – ou mudança de variáveis – no problema elíptico a ser resolvido, transformando o domínio físico em um retângulo no domínio computacional. Sejam  $\xi$  e  $\eta$  as variáveis dependentes antes do mapeamento e  $x$  e  $y$  as independentes. Após a aplicação da lei de mapeamento, ocorre uma troca de variáveis dependentes, e o par  $(x,y)$  passa a ser dependente de  $(\xi,\eta)$ . Assim, o sistema de equações dado pelas eqs. 6 e 7 é resolvido no domínio computacional, e as iso-soluções são obtidas já no domínio físico.

$$\alpha x_{\xi\xi} + \gamma x_{\eta\eta} - 2\beta x_{\xi\eta} = 0 \quad (6)$$

$$\alpha y_{\xi\xi} + \gamma y_{\eta\eta} - 2\beta y_{\xi\eta} = 0 \quad (7)$$

onde

$$\begin{aligned} \alpha &= x_\eta^2 + y_\eta^2 \\ \gamma &= x_\xi^2 + y_\xi^2 \\ \beta &= x_\xi x_\eta + y_\xi y_\eta \end{aligned} \quad (8)$$

Com relação às condições de contorno, uma possibilidade é especificar os valores de  $x$  e  $y$  na fronteira  $\Gamma$  em função de  $\xi$  e  $\eta$  (condições de Dirichlet). Outra opção seria impor derivada direcional nula nas fronteiras (condições de Neumann), que garantiriam a ortogonalidade do sistema coordenado. Como isto aumenta o custo computacional do problema, foram adotadas condições de Dirichlet, isto é:

$$\text{em } \Gamma_i : \begin{cases} x = f_i(t) \\ y = g_i(t) \end{cases} \quad i = 1,2,3,4 \quad (9)$$

onde

$$\Gamma = \bigcup_{i=1}^4 \Gamma_i \tag{10}$$

e  $f_i(t)$  e  $g_i(t)$  são as funções paramétricas que definem as fronteiras do domínio. Neste caso, foram empregadas splines cúbicas, conforme formulação da seção anterior.

#### 4. Suavização da Malha

Na metodologia proposta neste trabalho, os processos de suavização da malha e dos blocos são executados simultaneamente, de maneira relativamente simples: Aplicamos o Laplaciano como um suavizador da malha em cada bloco, num processo de busca para se obter elementos regulares nas fronteiras. A cada iteração alteramos as coordenadas dos nós das fronteiras do bloco para se obter esta regularidade; estas coordenadas são fornecidas ao laplaciano como condições contorno de Dirichlet. Se uma face do bloco for compartilhada por outro bloco, os pontos que a definem podem ser tratados como pontos internos ao domínio, e desse modo conseguimos uma continuidade  $C^2$  na transição da malha nas faces dos blocos que são internas ao domínio. Entretanto, nem sempre a continuidade  $C^2$  garante melhor distribuição dos elementos sobre os blocos. Ela apenas garante maior suavidade na transição da malha entre blocos adjacentes. Por isso, cabe ao usuário determinar se deseja ou não tal suavidade no domínio. A Figura (2) mostra um exemplo onde em algumas regiões a transição  $C^0$  é mais indicada e outras onde a transição  $C^2$  conduz a melhores resultados. Observe que enquanto na malha da esquerda a adaptação à geometria da estrutura foi melhor, na figura da direita os elementos estão mais regulares geometricamente. Este efeito é tão mais pronunciado quanto mais refinada for a malha.

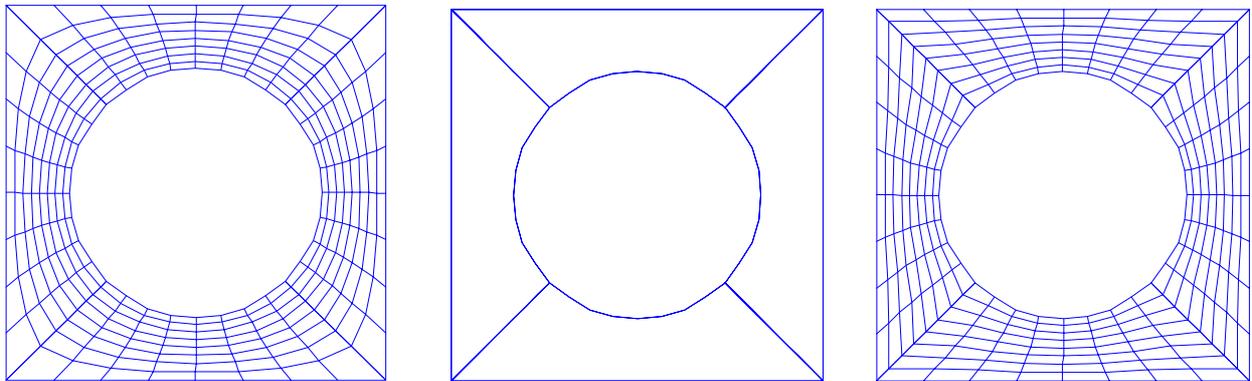


Figura 2: Discretização com continuidade  $C^2$  (esquerda) e  $C^0$  (direita) na transição entre os blocos.

O processo de suavização da malha foi tratado como um problema de otimização que busca a minimização da deformação dos elementos da fronteira do bloco. Os nós localizados sobre esta fronteira constituem as variáveis de projeto, e o critério de deformação adotado consiste em medir o desvio da forma dos elementos em relação a um quadrado. As equações para quantificação desta deformação foram determinadas empiricamente. O processo consiste em medir a variação do erro para cada elemento, e com base neste valor alterar localmente a posição dos nós do elemento, impondo uma perturbação na sua coordenada paramétrica.

Cada bloco é modelado como um objeto computacional, composto pelas seguintes propriedades:

- Fronteira  $\Gamma$ , composta por quatro curvas splines cúbicas ( $\Gamma_s, \Gamma_E, \Gamma_N, \Gamma_W$ , em referência a sul, leste, norte e oeste), que serão chamadas de faces;
- Conjunto de nós internos  $(x,y)$  no interior do bloco;
- Quantidade de elementos presentes em cada face do bloco( $m$ );
- Conjunto de condições de contorno definidas parametricamente sobre  $\Gamma$  pelo vetor  $\{t\} = \{t_1, t_2, t_3, \dots, t_{4m}\}$ . Estas condições também serão os nós da fronteira do bloco;
- Um valor de mérito  $E$ , que quantifica a distorção dos elementos localizados na fronteira  $\Gamma$ .

Para um nó genérico de fronteira  $k$  ( $1 \leq k \leq 4m$ ), sua coordenada paramétrica  $t_k$  é dada por

$$t_k = k + g_k \tag{11}$$

onde  $k$  representa a parte inteira da coordenada e  $g_k$  sua parte decimal. Por exemplo, para  $t = 3.8$ , temos  $k = 3$  e  $g_k = 0.8$ . Para que a malha inicial seja gerada com nós igualmente espaçados na fronteira, na primeira iteração adotamos  $g_k = 0$ .

A partir daí, inicia-se um processo de busca pelo conjunto ótimo de condições de contorno que minimize a distorção geométrica dos elementos de fronteira, alterando o valor de  $g$ . Formula-se então o seguinte problema de otimização:

**Dados:**

- Um domínio bidimensional  $\Omega$ , limitado pela fronteira  $\Gamma = \Gamma_S \cup \Gamma_E \cup \Gamma_N \cup \Gamma_W$ ;
- Uma Malha  $\mathbf{M}$ , caracterizada pelos seguintes parâmetros:
  - Conjunto de nós internos  $(\mathbf{x}, \mathbf{y})$  no interior do domínio;
  - Quantidade de elementos presentes em cada face do domínio ( $m$ );
  - Conjunto de parâmetros  $\{\mathbf{t}\} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \dots, \mathbf{t}_{4m}\}$  que definem as condições de contorno sobre a fronteira;
  - Valor de mérito  $\mathbf{E}$ , que quantifica a distorção dos elementos localizados na fronteira  $\Gamma$ ;

**Obter:**

Conjunto de condições de contorno  $\{\mathbf{t}^*\}$  que minimize  $\mathbf{E}$ .

Na seqüência de descrição do algoritmo, apresenta-se primeiramente a formulação do critério de otimização, e depois o método de busca.

Uma vez que todos os elementos são quadriláterais, o critério adotado neste trabalho consiste em medir a distorção de cada elemento da fronteira em relação a um quadrado, e o desempenho da malha será medido pela soma destas distorções. Para o elemento genérico da figura 3, a tabela 1 mostra as condições adotadas neste trabalho que determinam o desvio em relação à forma considerada ideal, bem como as equações que quantificam este desvio.

Tabela 1: Condições a serem satisfeitas por um elemento quadrilateral genérico

Condição	Equação
1. $A = F$	$a = (x_0 - x_1) - (x_2 - x_3) = 0$ (12a)
2. $D = I$	$b = (y_0 - y_1) - (y_2 - y_3) = 0$ (12b)
3. $D = F$	$c = (y_1 - y_3) - (x_2 - x_3) = 0$ (12c)
4. $C = G$	$d = (x_1 - x_3) - (y_3 - y_2) = 0$ (12d)

O erro associado ao elemento será dado por:

$$\varepsilon = a^2 + b^2 + c^2 + d^2 \tag{13}$$

Note que este critério é necessário mas não suficiente para se obter um elemento quadrado. Para isso seriam necessários testes de ortogonalidade envolvendo o cálculo de produtos internos, o que aumentaria o custo computacional em até 20%, segundo alguns cálculos preliminares. Como a qualidade da malha obtida foi praticamente a mesma para todos os casos, optou-se por um critério menos rigoroso, a favor do desempenho computacional.

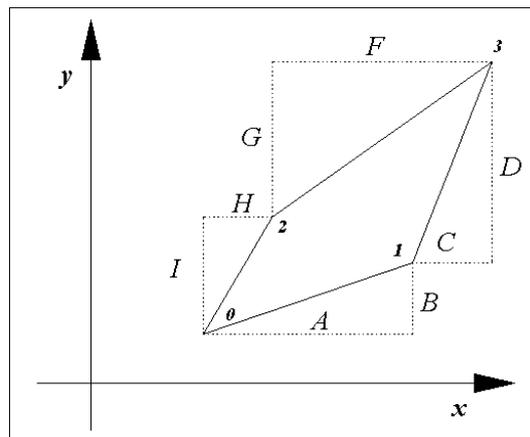


Figura 3: Desvio de forma para um elemento quadrilátero genérico.

O método de busca por gradiente usado é baseado no método da máxima descida. As variáveis de projeto são as coordenadas paramétricas dos nós de fronteira do domínio, e as variáveis dependentes são as distorções dos elementos que definem a fronteira. Portanto, interessa agora obter uma relação entre a variação do posicionamento destes nós e a distorção destes elementos, quantificada pelas eqs. (12).

A atualização de  $t_k$  na iteração  $i$  é dada por

$$t_k^{(i)} = \left( t_k - \Delta t_k \cdot \left( \frac{\partial \mathbf{E}}{\partial t_k} \right) \right)^{(i-1)} \quad (14)$$

Substituindo-se a eq. (14) na eq. (11), obtém-se a expressão para a atualização de  $g_k^{(i)}$ , que é a perturbação imposta à coordenada paramétrica do nó  $k$  na iteração  $i$ .  $\mathbf{E}$  é a distorção ou erro de todos os elementos de fronteira da malha analisada.

$$g_k^{(i)} = \left( g_k - \Delta g_k \cdot \left( \frac{\partial \mathbf{E}}{\partial t_k} \right) \right)^{(i-1)} \quad (15)$$

Sendo o desvio em cada elemento dado pela equação 13, o erro total da malha será

$$\mathbf{E} = \sum_{e=1}^{4m} (a_e^2 + b_e^2 + c_e^2 + d_e^2) = Sa + Sb + Sc + Sd \quad (16)$$

onde

$$Sa = \sum_{e=1}^{4m} (a_e^2), \quad Sb = \sum_{e=1}^{4m} (b_e^2), \quad Sc = \sum_{e=1}^{4m} (c_e^2) \quad e \quad Sd = \sum_{e=1}^{4m} (d_e^2)$$

A variação do erro é calculada considerando-se que apenas os nós da fronteira estejam livres, uma vez que os nós interiores serão modificados pelo gerador elíptico (nesta etapa, os nós que definem vértices do bloco não serão alterados). Então, as variáveis de projeto serão as coordenadas dos nós, que podem ser representadas parametricamente pelo parâmetro  $t$ . As componentes do gradiente do erro serão calculadas pela eq.(17).

$$\frac{\partial \mathbf{E}}{\partial t_k} = \frac{\partial Sa}{\partial t_k} + \frac{\partial Sb}{\partial t_k} + \frac{\partial Sc}{\partial t_k} + \frac{\partial Sd}{\partial t_k} \quad (17)$$

onde

$$\frac{\partial Sa}{\partial t_k} = 2 \cdot \frac{\partial x_k}{\partial t_k} \{a_k - a_{k-1}\} \quad (18a)$$

$$\frac{\partial Sb}{\partial t_k} = 2 \cdot \frac{\partial y_k}{\partial t_k} \{b_k - b_{k-1}\} \quad (18b)$$

$$\frac{\partial Sc}{\partial t_k} = 2 \cdot \frac{\partial y_k}{\partial t_k} \cdot c_k \quad (18c)$$

$$\frac{\partial Sd}{\partial t_k} = 2 \cdot \frac{\partial x_k}{\partial t_k} \cdot d_k \quad (18d)$$

e  $a_k$ ,  $b_k$ ,  $c_k$  e  $d_k$  são dados pelas equações 12.

O gradiente do erro para cada elemento será dado então por

$$\nabla \mathbf{E}_k = 2 \cdot \left\{ \frac{\partial x_k}{\partial t_k} \cdot [a_k - a_{k-1} + d_k] + \frac{\partial y_k}{\partial t_k} \cdot [b_k - b_{k-1} + c_k] \right\} \quad (19)$$

O cálculo de  $\frac{\partial x_k}{\partial t_k}$  e  $\frac{\partial y_k}{\partial t_k}$  é feito derivando-se as eqs. 1 e 2 em relação a t.

#### 4.1 Realocação de vértices internos ao domínio

Em alguns casos, no modelo pode haver não somente faces internas mas também vértices internos ao domínio, que também devem ser realocados durante a alteração das faces dos blocos na otimização da malha. Por facilidade de implementação, o reposicionamento destes vértices não é feito pelo laplaciano. Eles são reposicionados fazendo-se uma média ponderada da posição de seus vizinhos imediatos, através de uma relação desenvolvida empiricamente. A condição para que um vértice seja interno ao domínio é que ele seja compartilhado por pelo menos 3 blocos, e cada um destes blocos tenha dois vizinhos neste conjunto. Com base neste critério, foi formulada uma regra empírica para o reposicionamento de vértices internos ao domínio, ponderando a posição deste ponto pela posição dos vizinhos. Atribui-se peso 3 para os nós localizados *sobre* as faces que o compartilham ( $x_1^{(i)}$  e  $x_2^{(i)}$ ) e peso 1 para os nós vizinhos imediatos *interiores ao domínio* ( $x_3^{(i)}$ ). A Figura (4) mostra um vértice interno e seus pontos vizinhos. A equação (20) mostra como são reposicionados os vértices internos.

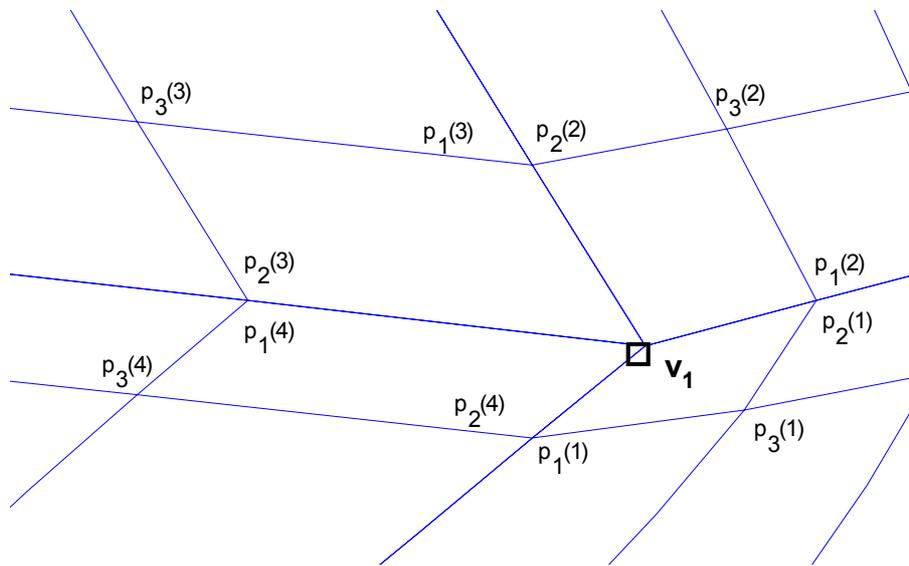


Figura 4. Identificação dos pontos vizinhos a um vértice interno, que influenciam no seu reposicionamento.

$$x_p = \frac{\sum_{i=1}^{n\_blocos} [3 \cdot (x_1^{(i)} + x_2^{(i)}) + x_3^{(i)}]}{7 \cdot n\_blocos} \tag{20}$$

$$y_p = \frac{\sum_{i=1}^{n\_blocos} [3 \cdot (y_1^{(i)} + y_2^{(i)}) + y_3^{(i)}]}{7 \cdot n\_blocos}$$

onde  $n\_blocos$  é o número de blocos que compartilham o vértice interno.

Portanto, o algoritmo para suavização de uma malha sobre um domínio bidimensional arbitrário pode ser assim resumido:

1. Definir os blocos parametricamente a partir dos pontos de suporte fornecidos pelo usuário.
2. Para cada bloco, identificar seus vizinhos, faces internas e vértices internos ao domínio.
3. Discretizar cada bloco em uma malha estruturada.
4. Aplicar o método de suavização da malha.
5. Realocar os vértices internos, se houverem.
6. Atribuir mérito à malha.
7. Voltar ao passo 3 até a convergência.

Como critério de convergência foi adotada a estabilização do erro médio após duas iterações consecutivas. Se este valor se mantiver abaixo de um determinado valor máximo, consideramos que o processo convergiu.

## 5. Resultados

O caso apresentado a seguir permite analisar os prós e contras desta metodologia de suavização de malhas. Na figura 5 são apresentados, respectivamente, o domínio a ser discretizado, sua discretização em blocos, a malha gerada com continuidade  $C^0$ , a forma final dos blocos após a discretização com malha de continuidade  $C^2$  e a respectiva malha. A figura 6 mostra em detalhes as extremidades superior e inferior do domínio discretizado com continuidade  $C^0$  e  $C^2$ . A tabela 2 mostra o desempenho do algoritmo em cada caso. O número entre parênteses na coluna referente ao número de elementos da malha refere-se à quantidade de elementos em cada face de cada bloco.

Analisando as figuras 5 e 6, observamos que a suavidade na transição da malha nem fornece a melhor discretização. Isto se deve a uma característica intrínseca das equações de Laplace: as isoterms tendem a migrar de regiões convexas (cantos) para regiões côncavas (quinas), o que pode ou não contribuir para o processo de discretização. Por um lado, ocorre a concentração de elementos em torno do que seria o sólido imerso no fluido, sujeito portanto a um escoamento externo ao corpo. Por outro lado, os elementos próximos aos cantos, se não ficam geometricamente deformados, têm dimensões bem maiores do que o restante do domínio, numa região que não necessariamente necessita de uma malha grosseira. Já observando a extremidade inferior do domínio, notamos que a continuidade  $C^2$  leva a elementos melhor adaptados ao domínio. Quanto ao custo computacional, consideramos o desempenho do algoritmo satisfatório, embora o tempo de execução cresça em escala não linear com o refinamento da malha.

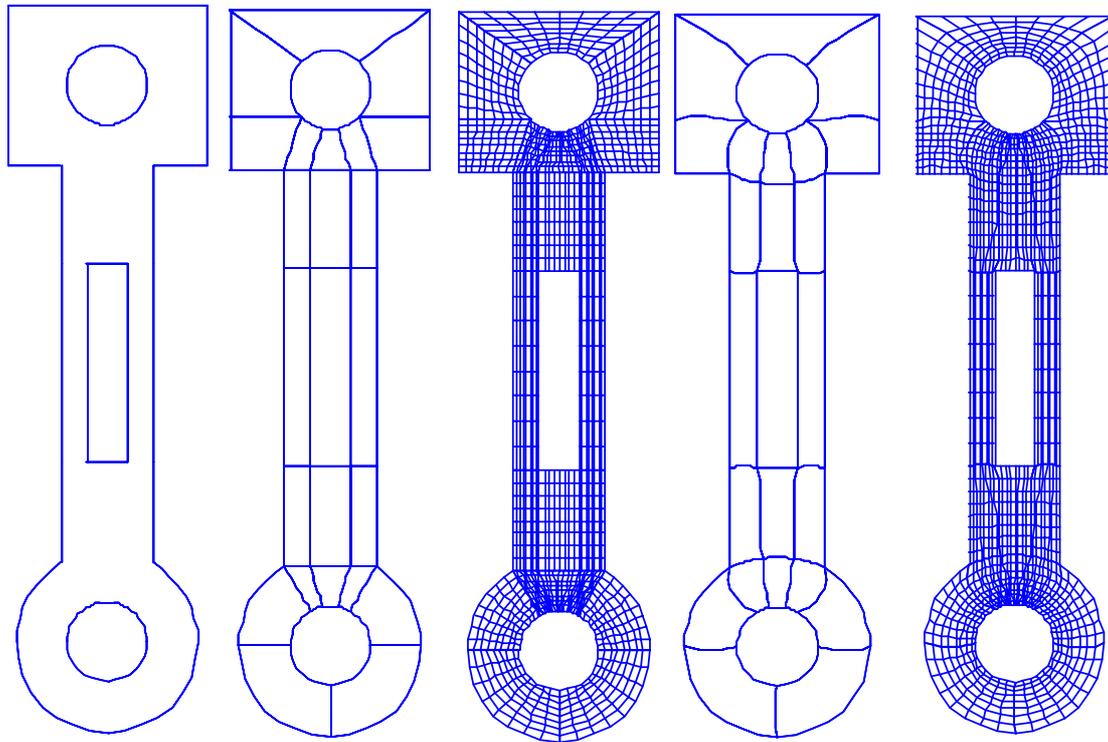


Figura 5: Domínio a ser discretizado e discretização multiblocos com continuidade  $C^0$  e  $C^2$ , da esquerda para a direita.

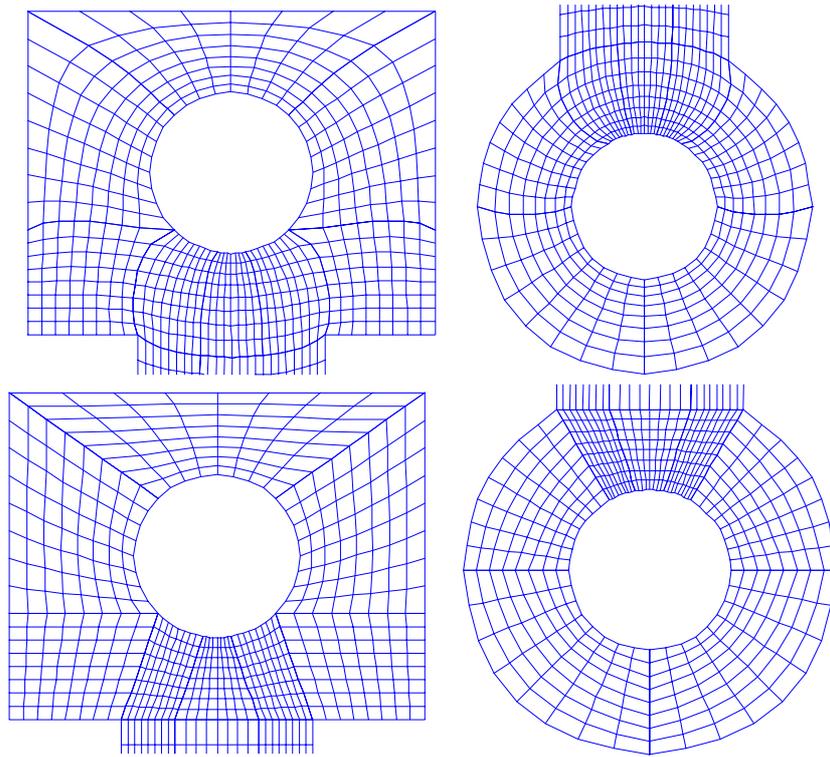


Figura 6: Detalhes das extremidades do domínio discretizado, apresentado na figura anterior. Acima, discretização com continuidade  $C^2$ . Abaixo, discretização com continuidade  $C^0$ .

Tabela 1. Custo computacional da discretização do Domínio

Elementos	Continuidade $C^2$			Continuidade $C^0$		
	Tempo em segundos	Número de Iterações	Erro Máximo	Tempo em segundos	Número de Iterações	Erro Máximo
1472(8)	0.281	6	33.5	0.09	4	36.8
5888(16)	1.382	10	17.6	0.4	4	18.7
13248(24)	6.9	13	12.4	1.5	4	12.6
23552(32)	21.35	16	9.8	4.4	4	9.5

## 6 Conclusões e Perspectivas

Os resultados mostram que o método de suavização empregado é robusto quanto ao tratamento de geometrias complexas. Quanto à continuidade da malha, os dois métodos fornecem bons resultados, dependendo da forma do domínio. Por isso, deve-se estudar a aplicação mista destas condições, de modo a aplicar cada uma nas regiões onde forem necessárias. O tempo de execução mostrou-se competitivo em problemas que não exigem remalhagem. Para a aplicação deste método em problemas de geometria variável, a geração automática dos blocos deverá ser estudada nas próximas etapas.

## 7. Referências Bibliográficas

- Dannenhoffer J.F. *Automatic blocking for complex three-dimensional configurations*. NASA-CP-3291, workshop Proceedings: Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics, May 1995.
- De Nicola, C., Pinto, G., Tognaccini, R. *Stability of two-dimensional model problems for multiblock structured fluid-dynamics calculations*. Computers & Fluids Vol. 26, No. 1, pp 43-58, 1997.
- Enander, R., Sterner, E. *Analysis of internal boundary conditions and communication strategies for multigrid multiblock methods*. Report No. 7, 5:th December 1996
- Gribben, B.J., Badcock, K.J., Richards, B.E.. *Towards Automatic Multiblock Topology Generation*. University of Glasgow, Department of Aerospace Engineering, Report Number 9826, pp.41, 1998.
- Maliska, C. R., 1995, *Transferência de Calor e Mecânica dos Fluidos Computacional*, LTC – Livros Técnicos e Científicos Editora S. A., Rio de Janeiro.

- Oosterlee, C.W. *The convergence of parallel multiblock multigrid methods*. Applied Numerical Mathematics 19 (1995) 115 – 128
- Park, S., Lee, K. *Automatic Multiblock Decomposition Using Hypercube++ for grid Generation*. Computers & Fluids Vol. 27, No. 4, pp. 509-528, 1998
- Pivello, M.R. *Geração e Otimização de Malhas Estruturadas sobre Domínios Bidimensionais Arbitrários*. Dissertação de Mestrado, Universidade Federal de Uberlândia, 2001.
- Roger, D. F., Adams J. A., 1990, *Mathematical Elements for Computer Graphics*, McGraw-Hill, Inc, New York.
- Liu, J., Shyy, W. *Assessment of grid interface treatments for multi-block incompressible viscous flow computation*. Computers & Fluids Vol. 25, No. 8, pp 719-740, 1996.

## TWO DIMENSIONAL MULTIBLOCK GRIDS OPTIMIZATION

### Márcio Ricardo Pivello

Universidade Federal de Uberlândia – Faculdade de Engenharia Mecânica  
[pivello@mecanica.ufu.br](mailto:pivello@mecanica.ufu.br)

### Carlos Roberto Ribeiro

Universidade Federal de Uberlândia – Faculdade de Engenharia Mecânica  
[cribeiro@mecanica.ufu.br](mailto:cribeiro@mecanica.ufu.br)

**Abstract.** *In this work we present a scheme for smoothing 2D block-structured grids, which are widely used to solve CFD problems. Since these problems are often solved via finite volume method, the grid transition between adjacent blocks is of great importance in the convergence process. A multiblock grid is generated in two steps, namely the domain partition into quadrilateral subdomains, which are called blocks, and the structured grid generation within each block. The block generation in this work is manual and an initial structured grid is assumed to be given within each block. In order to obtain good results after solving the physical problem, some conditions concerning the grid must be observed: grid points located on a boundary shared by two blocks must have the same coordinates for each of them. Also, since this grid will be used later to solve CFD problems via finite volume methods,  $C^2$  continuity on the interface is important. To achieve such a smoothness we will use a Laplacian operator formulated in curvilinear coordinates. In order to ensure  $C^2$  continuity between adjacent blocks, the nodes located on the boundaries shared by two blocks are treated as if they were inside the block domain, that is: the Laplacian operator is applied to the boundary nodes, which become the last row / column of the block, and the new boundary nodes will be the ones located at the first row / column of the adjacent block, which will be the boundary conditions to the Laplacian operator. The smoothing process replaces the boundary nodes in order to obtain geometrically regular elements, and the design variables are the boundary nodes coordinates. These work as Dirichlet boundary conditions to the Laplacian operator, which will replace the internal nodes. The domain representation was done with parametric cubic splines, which were used to apply the boundary conditions too. The results show that the merging of the grid and block topology smoothing processes leads to good results and makes the algorithm implementation easier.*

**Keywords.** *Multiblock grids, grid smoothing, Laplacian operator, 2D grids*