

### 3 Método Fast Multipole

Nesse capítulo, apresenta-se o Método Fast Multipole (FMM), como proposto por Greengard e Rokhlin (1987). O algoritmo foi eleito um dos 10 melhores do século XX (DONGARRA e SULLIVAN, 2000).

O FMM foi inicialmente concebido para utilização em problemas de simulação de partículas para campos de Coulomb e gravitacionais, mas em meados da década de 90, com os trabalhos de Peirce e Napier (1995), Nishimura, Yoshida e Kobayashi (1999), entre outros, o FMM começou a ser estudado para utilização como ferramenta para acelerar o BEM. A complexidade computacional da solução de um programa de BEM, para um número  $N$  de variáveis, é, normalmente, de  $O(N^3)$ . Tal complexidade pode ser reduzida, com o auxílio de um resolvidor iterativo, para  $O(N \log N)$ , ou até mesmo  $O(N)$ .

Em princípio, para que o sistema de equações apresentado na eq. (2.19) seja solucionado, devem-se construir as matrizes  $\mathbf{H}$  e  $\mathbf{G}$ , e só então operar a solução do sistema. Tal construção é custosa, e consiste no cálculo da influência de um certo ponto *fonte* em todos os outros  $N$  pontos (nesse caso, denominados *campo*) em que o contorno foi discretizado. Tal operação constrói uma das  $N$  linhas das matrizes citadas. Com o advento do FMM, pode-se calcular a influência do ponto fonte sobre certo grupo de pontos campo, o que corresponde à multiplicação da linha das matrizes à medida que são construídas, economizando assim memória e processamento.

O grupo de pontos campo é aglomerado em um único ponto, doravante denominado *polo*, que contém todas as informações relevantes para o cálculo da influência do ponto fonte sobre todos os pontos campo a que diz respeito. Podem existir tantas camadas de polos quantas o usuário necessitar.

O método é tão eficaz quanto citado por Board e Schulten (2000), pois permite que simulações de larga escala utilizando o BEM sejam executadas em computadores pessoais, como pode ser visto em diversos exemplos numéricos,

para diversos tipos de problemas, no livro bastante completo e didático de Liu (2009).

Para acelerar a execução de um programa de BEM, o FMM realiza expansões das funções fundamentais, o que leva à introdução de erros numéricos. Tais erros são considerados no desenvolvimento do método, e podem facilmente ser controlados com o aumento, ou diminuição, do número de termos na expansão em questão e também com o devido controle das distâncias entre os pontos campo e fonte, e também os polos de expansão. Tal controle de distâncias é discutido na seção 3.1.2.

Outra questão relacionada ao FMM é sua complexidade de desenvolvimento, pois como dito acima, este método consiste na expansão em série das soluções fundamentais. Esse trabalho de expansão e manipulação pode se tornar bastante complicado, o que obriga que para cada tipo de problema um novo desenvolvimento seja realizado.

A seguir é apresentada a formulação do FMM para aplicação a problemas de potencial.

### **3.1. FMM para problemas de potencial 2D**

O objetivo do algoritmo apresentado nesta seção é acelerar a solução do sistema de equações apresentado na eq. (2.19), mas que pode ser prontamente estendido a uma de suas variações, como o HBEM ou o EBEM (eqs. (2.39), e (2.44), respectivamente).

Tais sistemas de equações, em termos de implementação, são reescritos na forma

$$\mathbf{Ax} = \mathbf{b}, \quad (3.1)$$

onde a matriz  $\mathbf{A}$  é construída a partir da combinação de linhas e colunas das matrizes  $\mathbf{H}$  e  $\mathbf{G}$ , e o vetor  $\mathbf{b}$  é construído de fórmula análoga à matriz  $\mathbf{A}$ , mas a partir de produtos com os vetores  $\mathbf{t}$  e  $\mathbf{d}$ . As condições de contorno do problema definem quais linhas e colunas serão utilizadas para a construção de  $\mathbf{A}$ .

Trabalha-se no plano complexo, de forma que um dado ponto de coordenadas cartesianas  $(x, y)$  seja representado por  $z = x + yi$ . As soluções fundamentais para um problema de potencial 2D são

$$u^*(z, z_0) = -\frac{\ln|z - z_0|}{2\pi}, \text{ e} \quad (3.2)$$

$$q^*(z, z_0) = \frac{\partial u^*(z, z_0)}{\partial n} = n(z) \frac{\partial u^*(z, z_0)}{\partial z}, \quad (3.3)$$

onde  $u^*$  é a solução fundamental de um problema de potencial,  $q^*$  representa o fluxo,  $n$  é o vetor normal ao contorno do corpo,  $z$  é um ponto campo e  $z_0$  é um ponto fonte.

Estas equações são expressas em coordenadas cartesianas como

$$u^*(x, y) = \text{Re} \left[ u^*(z, z_0) \right], \text{ e} \quad (3.4)$$

$$q^*(x, y) = \text{Re} \left[ q^*(z, z_0) \right] = n_1 \text{Re} \left( \frac{\partial u^*(z, z_0)}{\partial z} \right) - n_2 \text{Im} \left( \frac{\partial u^*(z, z_0)}{\partial z} \right). \quad (3.5)$$

### 3.1.1. Expansões das soluções fundamentais

Sejam o ponto fonte  $z_0$ , e seus polos  $z_{L'}$  e  $z_L$ , e o ponto campo  $z$ , e seus polos  $z_c$  e  $z_{c'}$ , representados em um plano complexo, na Figura 2 abaixo.

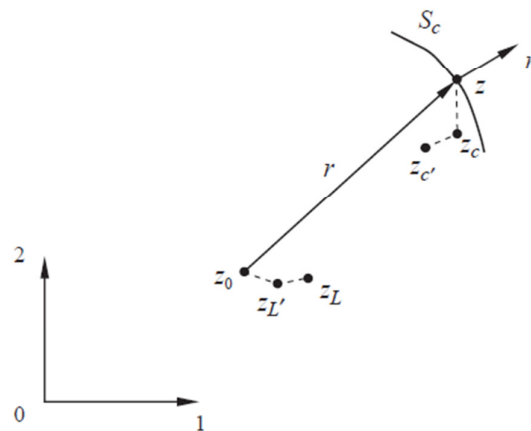


Figura 2 – Pontos descritos em notação complexa e seus polos de expansão. (LIU, 2009)

Realiza-se uma expansão em série das funções  $u^*$  e  $q^*$  inicialmente em torno do ponto  $z_c$ , próximo ao ponto campo, e distante do ponto fonte.

### 3.1.1.1.

#### Expansão da solução fundamental $u^*$

Esta expansão fornece

$$u^*(z, z_0) = -\frac{1}{2\pi} \left( \ln |z_0 - z_c| - \sum_{i=0}^{\infty} i^{-1} \left( \frac{z - z_c}{z_0 - z_c} \right)^i \right). \quad (3.6)$$

Nota-se que  $u^*$  não é mais função da diferença entre os pontos fonte  $z_0$  e campo  $z$  – que representa a distância entre eles, nesta notação complexa –, sendo agora função de  $z - z_c$  e  $z_0 - z_c$ . Esta é a característica mais importante do FMM, pois permite que todos os cálculos relativos aos pontos campo sejam realizados sem que seja necessário fixar um ponto fonte para isto. Isso permite que todas as informações relacionadas aos pontos campo distantes de certo grupo de pontos fonte sejam pré-calculadas.

Em termos de implementação, isso significa que uma certa linha das matrizes  $\mathbf{H}$  e  $\mathbf{G}$ , que convencionalmente é calculada fixando-se um ponto fonte e varrendo todos os outros  $N$  pontos campo, pode ser calculada fixando-se um ponto fonte e calculando sua influência no conjunto de pontos campo. Essa última informação é obtida no polo de expansão  $z_c$ .

Seguindo a notação introduzida por Liu e Nishimura (2006), a eq. (3.6) pode ser reescrita como

$$u^*(z, z_0) = \frac{1}{2\pi} \sum_{i=0}^{\infty} O_i(z_0 - z_c) I_i(z - z_c), \quad (3.7)$$

onde as funções  $O(z)$  e  $I(z)$  são definidas como

$$\begin{aligned} I_i(z) &= \frac{z^i}{i!}, & \text{para } i \geq 0 \\ O_i(z) &= \frac{(i-1)!}{z^i}, & \text{para } i \geq 1, \text{ e } O_0 = -\ln(z) \end{aligned} \quad (3.8)$$

A construção da matriz  $\mathbf{G}$  é realizada através da integração apresentada na eq. (2.21), deduzida para problemas de elasticidade, mas que pode ser prontamente particularizada para problemas de potencial pela supressão do índice  $i$ , que levando em conta a expansão apresentada na eq. (3.7) é escrita como

$$G_{ml} = \int_{\Gamma} u_m^* t_l d\Gamma = \frac{1}{2\pi} \sum_{i=0}^{\infty} O_k(z_0 - z_c) M_i(z_c), \quad (3.9)$$

onde

$$M_i = \int_{\Gamma} I_i(z - z_c) t_l d\Gamma, \quad i = 0, 1, 2, \dots, n. \quad (3.10)$$

Os termos  $M_i$  são comumente denominados na literatura por *momentos* em torno de  $z_c$ . Nota-se que os termos  $M_i$  são independentes dos pontos fonte  $z_0$  e podem, então, ser pré-calculados no algoritmo.

Sucessivas expansões  $z_c$  podem ser realizadas em torno de um novo polo  $z_{c'}$ , que levam a

$$M_i(z_{c'}) = \sum_{j=0}^i I_{i-j}(z_c - z_{c'}) M_j(z_c), \quad (3.11)$$

onde  $M_i(z_{c'})$  é definido como a translação de *momento-para-momento* (M2M). Tal translação M2M tem um número finito de termos, ou seja, não introduz erros de aproximação no algoritmo.

De maneira análoga, pode-se expandir a eq. (3.9) em torno de um ponto  $z_L$  próximo ao ponto fonte  $z_0$  e distante do ponto campo, ou de seus polos, chegando-se a

$$G_{ml} = \frac{1}{2\pi} \sum_{i=0}^{\infty} O_i(z_0 - z_c) M_i(z_c) = \frac{1}{2\pi} \sum_{j=0}^{\infty} L_j(z_L) I_j(z_0 - z_L), \quad (3.12)$$

onde

$$L_j(z_L) = (-1)^j \sum_{i=0}^{\infty} O_{i+j}(z_L - z_c) M_i(z_c) \quad (3.13)$$

são denominados como translações *momento-para-local* (M2L).

Deve-se ainda considerar a possibilidade da expansão de  $z_L$  em torno de um novo polo  $z_{L'}$  que fornece

$$L_j(z_{L'}) = \sum_{k=0}^{n-j} I_k(z_{L'} - z_L) L_{j+k}(z_L), \quad (3.14)$$

onde  $n$  representa o número de termos para a expansão, e  $L_j(z_{L'})$  são as translações *local-para-local* (L2L).

### 3.1.1.2.

#### Expansão de $q^*$

Realiza-se então o estudo das integrações pertinentes à matriz  $\mathbf{H}$ , segundo a eq. (2.20). De forma análoga à expansão realizada para a função  $u^*$  na eq. (3.6), realiza-se a expansão da função  $q^*$  em torno de um certo ponto  $z_c$ , de forma que, segundo a definição de  $q^*$  apresentada na eq. (3.5) e da definição de  $u^*$  apresentada na eq. (3.9), obtém-se

$$\int_{\Gamma} q_{jm}^* n_j u_n d\Gamma = \frac{1}{2\pi} \sum_{i=1}^{\infty} O_i(z_0 - z_c) \tilde{M}_i(z_c), \quad (3.15)$$

onde

$$\tilde{M}_i(z_c) = \int_{\Gamma} n_j I_{i-1}(z - z_c) u_{jn} d\Gamma, \quad i=1,2,3,\dots, \quad (3.16)$$

são os *momentos* para a função  $q^*$ . Todas as outras translações são válidas para a função  $q^*$ , tomando-se em conta apenas que  $\tilde{M}_0 = 0$ .

### 3.1.2.

#### Pseudo-algoritmo do FMM aplicado a problemas de potencial no BEM

Nessa seção apresenta-se um breve resumo do procedimento computacional de execução do FMM. Tal procedimento é baseado no algoritmo apresentado por Liu (2009).

Inicia-se o algoritmo pela discretização do contorno. Tal discretização pode ser realizada da maneira apresentada nas eqs. (2.16) e (2.17).

Deve-se então determinar os polos de expansão e as dependências entre nós e polos. A abordagem comum, proposta por Liu (2009), é a criação de uma estrutura de árvore, em que o contorno a ser simulado é inserido em um domínio quadrado e que é então subdividido sucessivas vezes até que haja apenas um elemento do contorno por quadrado. Yasuda e Sakuma (2005) apresentam um trabalho bastante minucioso a respeito da criação da estrutura de árvore utilizada no FMM.

Tal estrutura de árvore se torna necessária, pois o erro do FMM está diretamente associado à distância entre polos e pontos fonte/campo, de forma que algum esforço computacional deve ser utilizado para determinar quais pontos

estão distantes o suficiente para que uma expansão possa ser aplicada de forma que haja uma perda de precisão controlada.

Um exemplo de tal estrutura hierárquica é apresentado na Figura 3. As células hachuradas representam células que contém apenas um nó, ou, neste caso, em que os elementos são constantes, apenas um elemento. Tais células são comumente denominadas de *folha*, fazendo alusão a uma estrutura hierárquica de árvores, em que o elemento mais externo seria uma folha. As células em branco representam células vazias, que não são de interesse para o algoritmo.

Note que esta estrutura cria uma relação hierárquica entre células, que pode ser – e é – utilizada para determinar a quantidade de polos, quais nós estarão ligados a eles, e quais pontos estão distantes o suficiente para a expansão. Os polos são definidos como os centros das células hachuradas.

Após a criação da estrutura de células e determinação de polos e dependências, inicia-se o FMM propriamente dito, como proposto por Greengard e Rokhlin (1987).

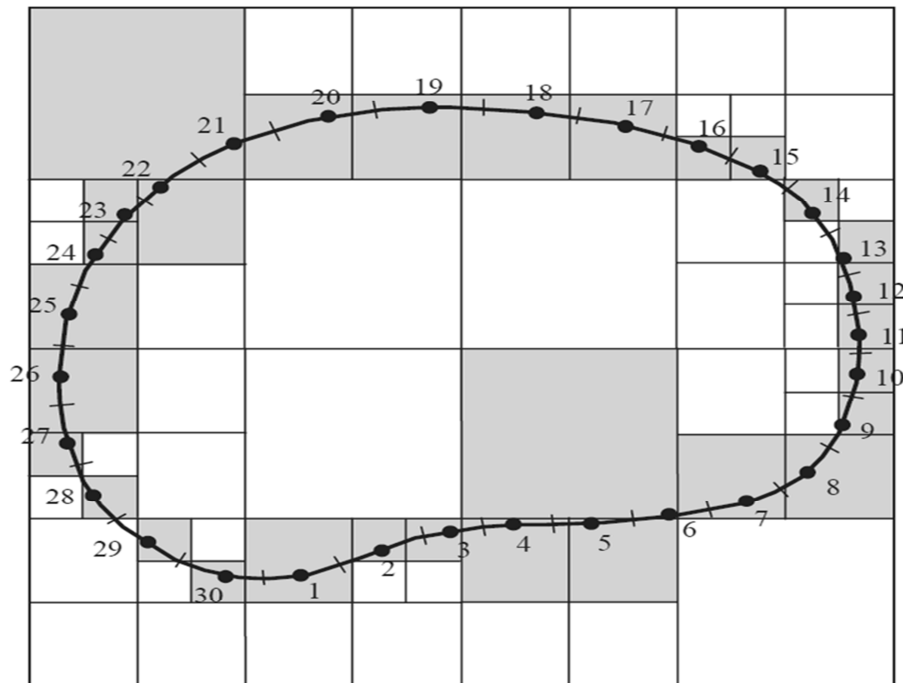


Figura 3 – Estrutura hierárquica de células para um contorno discretizado com elementos constantes, (LIU, 2009).

O primeiro passo é o *upward pass*, em que são computados os momentos dos pontos campo, tal como definido nas eqs. (3.10) e (3.16). O termo *upward* (para cima, em inglês), refere-se ao fato de que cada subdivisão das células cria

um nível mais baixo na estrutura, levando a entender que nesse passo do algoritmo, sobe-se a estrutura das folhas até determinado nível. Este nível é definido de forma que exista distância suficiente para a utilização das expansões. Após o cálculo dos momentos das folhas, faz-se a translação desses momentos, através da translação M2M, para os polos de nível mais alto até o nível estabelecido, em geral, o segundo.

Após a realização de todas as translações M2M, é realizado o *downward pass*, em que os momentos dos pontos campo são traduzidos, a partir das translações M2L, para os polos de nível 2 que estão distantes o suficiente para tal. Daí operam-se translações L2L, agora em direção às folhas, para entregar os resultados dos momentos dos pontos campo aos pontos fonte.

São realizadas, então, as integrações dos pontos campo próximos ao campo fonte em questão – situação em que não se podem desenvolver as expansões do FMM, sob pena de erros numéricos de ordem elevada.

O procedimento acima está integrado num resolvidor iterativo de sistemas de equações. Ele itera sobre o vetor  $\{x\}$  da eq. (3.1), que é atualizado ao fim das translações M2L e então todo o procedimento a partir do *upward pass* é repetido até que seja atingida a precisão desejada na solução do sistema de equações.